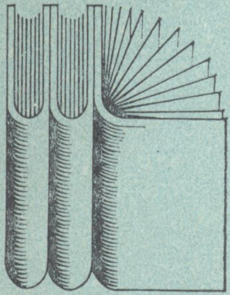
МІНІСТЕРСТВО
ВИЩОЇ
ТА СЕРЕДНЬОЇ
СПЕЦІАЛЬНОЇ
ОСВІТИ УРСР

НАВЧАЛЬНО-
МЕТОДИЧНИЙ
КАБІНЕТ
З ВИЩОЇ
ОСВІТИ

НАВЧАЛЬНИЙ
ПОСІБНИК



**М. М. Шкодин, А. І. Моргун,**
**А. С. Моргун**

# ПРОГРАМУВАННЯ ЗАДАЧ З БУДІВНИЦТВА ДЛЯ ЕОМ

КИЇВ

1990

МІНІСТЕРСТВО ВИЩОЇ І СЕРЕДНЬОЇ СПЕЦІАЛЬНОЇ ОСВІТИ УРСР
НАВЧАЛЬНО-МЕТОДИЧНИЙ КАБІНЕТ З ВИЩОЇ ОСВІТИ
ВІННИЦЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

М.М.Шкодин, А.І.Моргун, А.С.Моргун

# ПРОГРАМУВАННЯ ЗАДАЧ З БУДІВНИЦТВА ДЛЯ ЕОМ

Затверджено Радою Навчально-методичного кабінету
з вищої освіти при Мінвузі УРСР як навчальний посібник
для студентів спеціальності "Промислове і цивільне будівництво"

Київ НМК ВО 1990

Программирование задач по строительству для ЭВМ: Учеб. пособие / М.М.Шкодин, А.И.Моргун, А.С.Моргун. – К.: УМК ВО, 1990. – 144 с. – На укр. яз.

Навчальний посібник складено на основі курсу лекцій з дисципліни "Чисельні методи і програмування для ЕОМ". Він містить теоретичні відомості про ЕОМ, програмування алгоритмічною мовою Фортран-IУ та методику побудови програм розв"язування інженерних задач. Розглядаються найпоширеніші чисельні методи математики, застосовувані в практиці інженера-будівельника.

Призначається для студентів будівельних спеціальностей. Може бути використаний в курсовому проектуванні, а також при виконанні розрахунково-проектувальних завдань з теоретичної і будівельної механіки.

Іл. 53. Табл. 24.

Учебное пособие составлено на основе курса лекций по дисциплине "Численные методы и программирование для ЭВМ". Содержит теоретические сведения об ЭВМ и программировании на алгоритмическом языке Фортран-IУ, а также методику построения программ решения инженерных задач. Рассматриваются наиболее распространенные численные методы математики, применяемые в практике инженера-строителя.

Предназначается для студентов строительных специальностей. Может быть использовано в курсовом проектировании, а также при выполнении расчетно-проектировочных заданий по теоретической и строительной механике.

Ил. 53. Табл. 24.

Рецензенти: Н.А.Малахова, канд.техн.наук, доц.
В.І.Сумський, канд.фіз.-мат.наук

ISBN 5-7763-0264-1

# І. ПРОГРАМУВАННЯ

## І.І. Коротка історія розвитку обчислюваних засобів. Дані про сучасну елементну базу обчислювальної техніки

Сучасні електронно-обчислювальні машини /ЕОМ/ застосовуються не тільки для обчислювальної роботи /така робота становить близько 20% загального обсягу використання ЕОМ/, вони виконують і такі ролі:

інформаційних систем та засобів збереження інформації /скарбниця людських знань/;

систем автоматизованого управління різними процесами і об"єктами;

систем для математичного моделювання об"єктів та процесів різної природи.

Виникнення та нагромадження обчислювальних засобів, формування основних ідей обчислювальної техніки тривало протягом багатьох століть. До найдавніших лічильних пристроїв належать китайська рахівниця суан-пан, лічильна дошка абак, яка застосовувалася в Стародавній Греції, Римі, у країнах Стародавнього Сходу.

У 16 ст. як різновид абака виникла руська рахівниця. У 20-ті роки 17 ст. була створена логарифмічна лінійка, яка служить людству понад 350 років.

Серед перших механічних лічильних машин можна назвати машину французького вченого Б.Паскаля /1642 р./, арифмометр німецького математика Г.В.Лейбніца /1694 р./, арифмометри петербурзького інженера В.Т.Однера /1874 р./ та видатного російського математика П.Л.Чебишева /1878 р./, які набули широкого практичного застосування.

В розробку теорії математичних машин великий вклад вніс відомий російський вчений О.М.Крилов, який у 1904 р. створив машину для інтегрування звичайних диференціальних рівнянь.

Перші обчислювальні машини з програмним управлінням з"явилися в першій третині 19 ст. Англійський вчений Ч.Беббідж, якому належить багато провідних ідей, покладених в основу перших ЕОМ, у 1833 р. розробив проект першої машини з програмним управлінням.

У 1941-1944 рр. з"являються машини німецького інженера К.Цузе, американського фізика Г.Эйкена.

Ідея створення електронних /лампових/ обчислювальних машин належить американському вченому Д.В.Атанасову. У 1946 році американські фізики створюють першу ЕОМ ЕНІАК.

Перші радянські ЕОМ з програмним управлінням /БЕСМ/ були розроблені в 1947-1951 рр. у Києві під керівництвом академіка С.О.Лебедєва, основоположника радянської електронної обчислювальної техніки. Це так звані ЕОМ першого покоління /1950-1960 рр. "Стрела", "Минск-I" та ін./ Вони виконували 20 тисяч операцій за секунду. Машини були громіздкі, на електронно-вакумних лампах, потребували багато енергії.

ЕОМ другого покоління /1960-1965 рр./ виконували до мільйона операцій за секунду, базувалися на напівпровідникових елементах із застосуванням діодів та транзисторів /"Мир, "Промінь", "Минск-22", "Наири-2", "М-220"/.

Незабаром з"являється багато різних типів ЕОМ, які відрізняються один від одного системою команд та способом їх кодування. Для кожного з них необхідно було створити своє математичне забезпечення: транслятори з алгоритмічними мовами, бібліотеку стандартних програм тощо. Вартість математичного забезпечення іноді перевершувала вартість ЕОМ. Тому ЕОМ третього покоління /1969 р./ почали розробляти "родинами", щоб вони були програмно, конструктивно та інформаційно подібними, мали однакове математичне забезпечення. ЕОМ цього покоління належать до єдиної системи /ЄС/, вони створюються та випускаються в рамках міжнародного співробітництва соціалістичних країн - учасниць РЕВ. Основа елементної бази ЄС ЕОМ - мікросхеми інтегрального типу. Швидкодія сягає до 3 мільйонів операцій за секунду, обсяг оперативної пам"яті /ОП/ - від 512 до 8192 кілобайт /К/. Нині сконструйовані та діють ЄС ЕОМ 1010, 1020, 1022, 1045, ..., 1055, ..., 1060, 1065.

На зміну ЕОМ третього покоління приходять машини четвертого покоління. Швидкодія їх становить $10^7 \dots 10^8$ операцій за секунду. Елементна база таких машин - великі та надвеликі інтегральні схеми /НІС/.

Це мультипроцесорні системи паралельноЇ обробки інформаціЇ, побудовані на функціональній модульній основі, а також сучасні персональні комп"ютери.

В ЕОМ п"ятого покоління плануються системи мовного введення/виведення, що дають змогу вести з машиною діалог в прямому розумінні цього слова.

## I.2. Подання інформаціЇ в ЕОМ

### Структурна схема ЕОМ. Зображення інформаціЇ в ЕОМ. Принцип обробки даних за допомогою ЕОМ

Логічну структуру будь-якоЇ моделі єдиноЇ системи /ЄС/ ЕОМ можна зобразити у вигляді узагальненоЇ схеми /рис. I.I/.

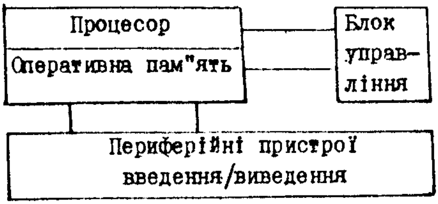
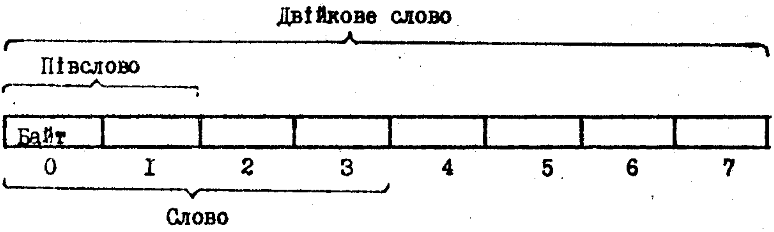


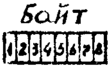
Рис. I.I. Загальна логічна схема ЕОМ

До складу ЕОМ входять: I/ центральний процесор; 2/ оперативна пам"ять /ОП/; 3/ периферійні пристроЇ введення та виведення; 4/ блок керування; 5/ системи програмного забезпечення.

Оперативна пам"ять - центральний блок машини, куди записується інформація, де вона зберігається і звідки за потребою видається різним блокам машини. В пам"яті ЕОМ зберігається різноманітна інформація, закодована у вигляді послідовних цифр 0 та I, які прийнято називати бітами /англ. *bit* - окорочено від *binary digit* - двійкова цифра/. Мінімальна одиниця інформаціЇ в пам"яті - байт, яка складається із восьми двійкових розрядів /бітів/, пронумерованих I,2,...,8 /рис. I.2/. Таким чином, в кожному біті можна записати 0 чи I, а в кожному байті подати $2^8 = 256$ різних комбінацій, що складаються з 0 та I. Кожний байт має адресу, яка визначає місце його знаходження в пам"яті. Адреса починається з 0.



2 байт = півслово;
4 байт = слово;
8 байт = двійкове слово.



Адреса слова /півслова, двійкового слова/ =
= адресі його першого байта
/I байт = 8 біт/

Рис. I.2. Структура внутрішньої пам"яті ЕОМ

Основна характеристика пам"яті - місткість. Одиниці вимірювання місткості: I Кбайт = I024 байт, I Мбайт = I024 Кбайт.

Процесор - головна частина ЕОМ - безпосередньо виконує обробку інформації та керує нею. З фізичної точки зору інформація, яка циркулює в ЕОМ, кодується за допомогою електричних сигналів. Обробку цих сигналів виконують електронні пристрої, які складаються із сотень тисяч окремих елементів - мікросхем, що спільно працюють. Процесор забезпечує автоматичне виконання програми, яка зберігається в пам"яті.

Текстова програма, або вихідний модуль, подається в машинних командах - транслюється. Якщо в програмі є помилки, то вона повертається програмісту на доопрацювання; якщо помилок немає, то процесор формує об"єктний модуль, залучаючи друкування та редактор зв"язків. З одного чи n об"єктних модулів формується завантажувальний модуль. Далі програму, переведену в машинні команди, процесор передає в ОП. Отже, програма, підготовлена для реалізації на ЕОМ проходить наступні етапи: запис - трансляція - редагування - виконання в ОП /рис. I.3/ [I].

Пристрої введення/виведення забезпечують обмін даними між оперативною пам"яттю та перфокартками, дисками, дисплеями тощо.

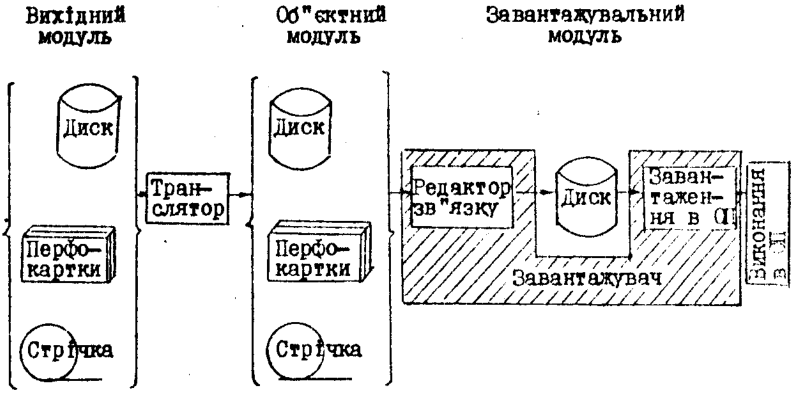


Рис. 1.3. Спрощена схема роботи процесора

Блок керування ЕОМ забезпечує всі режими її роботи, відбиває стан роботи машини в поточний момент часу.

Системи програмного забезпечення /СПЗ/ – це сукупність програм, процедур та правил для спілкування користувача з ЕОМ. СПЗ бере на себе головну роботу щодо виконання обчислень, керує роботою процесора, пам"яті, пристроїв введення/виведення тощо /рис. 1.4/ [1].

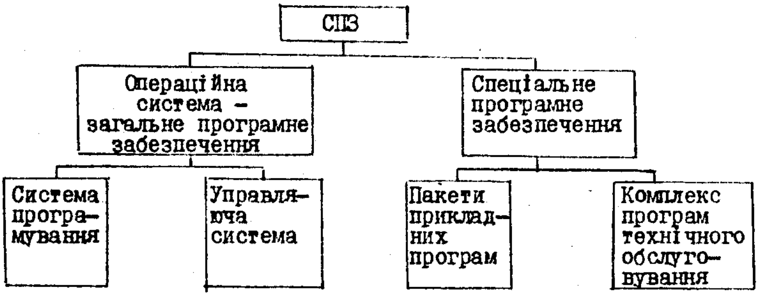


Рис. 1.4. Склад системи програмного забезпечення

# Алфавітний засіб задання інформації. Системи числення

Алфавітом називають будь-яку скінченну сукупність попарно різних символів. Символи, з яких складається алфавіт, називаються буквами цього алфавіту. Наприклад, сукупність $\{ \Delta ; a;x \}$ – це алфавіт, а її елементи – букви. Усяка скінченна послідовність букв, узятих з будь-якого алфавіту, називається словом. Так, у розглянутому прикладі словом буде кожна послідовність виду $\Delta a \Delta\Delta x \Delta$, $|x|$, $ax$ тощо.

Якщо взяти український алфавіт, то словами в ньому будуть всі слова розмовної мови, а також довільні сполучення букв МФТТК тощо. Слово, яке не містить у собі символів, є порожнім. Воно відповідає прогалині між словами.

Алфавітний спосіб задання інформації є універсальним. Будь-яку інформацію можна записати таким способом.

Системою числення називають спосіб подання чисел за допомогою спеціальних знаків /цифр/. У позиційних системах числення значення кожної цифри числа зумовлюється не тільки самою цифрою, але й її позицією – місцем, яке ця цифра займає в числі. Номер позиції, якщо лічити справа наліво, називається розрядом числа. Наприклад, розглянемо число 7I2 в десятковій системі числення. Перша справа цифра 2 /розряд одиниць/ означає два, друга цифра I /розряд десятків/ означає один десяток, третя цифра 7 /розряд сотень/ означає сім сотень.

Основа системи числення – це відношення наступного розряду до попереднього. Так, у десятковій системі числення кожний наступний старший розряд в I0 раз більший від попереднього.

В повсякденному житті для лічби людина використовує десяткову систему числення, алфавіт якої складається з I0 арабських цифр

$$\{0,\ I,\ 2,\ \ldots,\ 9\}_{I0}. \qquad /I.I/$$

З"ясувалося, що ЕОМ зручніше лічити не в десятковій, а в двійковій системі числення, алфавіт якої складається з двох цифр:

$$\{0,\ I\}_2. \qquad /I.2/$$

У позиційних системах числення для запису числа може бути використана тільки строго визначена кількість введених цифр. У десятковій системі таких цифр I0. Якщо основа позиційної системи числення дорівнює $q$, то для запису будь-яких цілих чисел в цій системі досить $q$ цифр, включаючи 0.

Якщо $q \leq$ 10, використовують перші $q$ цифр з десяткової системи числення, а саме: 0,1,2,...$q$-1. Наприклад, алфавіт п"ятіркової системи числення $\{0,1,2,3,4\}_5$. Якщо $q >$ 10, використовуються всі цифри десяткової системи і необхідна кількість нових заздалегідь обумовлених символів. Наприклад, для позначення 10,11,12,... можна взяти букви латинського алфавіту: A,B,C,... Зокрема, алфавіт шістнадцяткової системи числення, яка поряд з двійковою широко застосовується в ЕОМ, має такий вигляд:

$$\{0,1,\ldots,8,9,A,B,C,D,E,F\}_{16}. \qquad /1.3/$$

У десятковій системі числення будь-яке натуральне число $N > 9$ можна записати у вигляді многочлена

$$N_{10} = a_{n-1}10^{n-1} + a_{n-2}10^{n-2} + \ldots + a_1 10^1 + a_0 10^0, \qquad /1.4/$$

де

$$a_i = 0,1,2\ldots9;\ i = 0,1,2\ldots n-1; \qquad /1.5/$$

$$N_{10} = \overline{a_{n-1}a_{n-2}a_{n-3}\ldots a_1 a_0}. \qquad /1.6/$$

Записати будь-яке число в позиційній системі числення з основою $q$ означає подати це число у вигляді многочлена, записаного за спадними степенями символа $q$ з різними коефіцієнтами, меншими за $q$ :

$$N_q = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \ldots + a_1 q^1 + a_0 q^0, \qquad /1.7/$$

або

$$N_q = \overline{a_{n-1}a_{n-2}\ldots a_1 a_0}. \qquad /1.8/$$

Отже, коефіцієнти при степенях $q$ і будуть цифрами в запису числа виду /1.8/. Наприклад,

$$234_5 = 2\cdot5^2 + 3\cdot5^1 + 4\cdot5^0;$$
$$78_{10} = 7\cdot10^1 + 8\cdot10^0.$$

## Двійкова система числення. Переведення цілих чисел з однієї системи числення в іншу

Двійкова система числення широко застосовується в кібернетиці на таких підставах:

1/ двійкові числа зручніше, ніж числа інших систем, демонструвати, розрізняти та запам"ятовувати;

2/ двійкова арифметика значно простіша за інші.

Демонструвати двійкові числа /0 та I/ легко за допомогою гірлянди електричних лампочок: засвічена лампочка відповідає I, погашена – 0. Зображене таким чином число легко прочитати.

Вводити в ЕОМ двійкове число можна по провідниках електричного струму. Наявність напруги на кінці провідника означає, що демонструється I, якщо напруга відсутня – демонструється 0.

Таблиці арифметичних дій в двійковій системі числення найпростіші і мають такий вигляд:

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 10 |

| × | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| - | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Додавати, віднімати та множити навіть багатоцифрові двійкові числа дуже легко. Розглянемо приклади:

$$+\begin{array}{r} 1101011_2 \\ 110111_2 \\ \hline 10100010_2 \end{array} \qquad -\begin{array}{r} 1011_2 \\ 101_2 \\ \hline 110_2 \end{array} \qquad \times\begin{array}{r} 1101_2 \\ 101_2 \\ \hline 1101 \\ +\;1101\phantom{00} \\ \hline 1000001_2 \end{array}$$

При діленні чисел в двійковій системі використовуються таблиці множення та віднімання. Наприклад:

$$\begin{array}{r|l} -10100_2 & 101_2 \\ \cline{2-2} 101\phantom{00} & 100_2 \\ \hline -000 & \\ 000 & \\ \hline - & \end{array}$$

Цілі числа з будь-якої системи в десяткову переводяться методом підстановки. Для цього число подається в розгорнутій формі /I.7/ за спадними степенями основи і виконати арифметичні дії.

Наприклад:

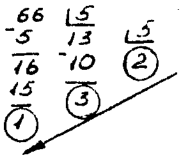
$$(1010)_2 = 1\cdot2^3+0\cdot2^2+1\cdot2^1+0\cdot2^0 = 1\cdot8+0\cdot4+1\cdot2+0\cdot1 = (10)_{10};$$

$$(A1E)_{16} = A\cdot16^2+1\cdot16^1+E\cdot16^0 = 10\cdot16^2+1\cdot16^1+14\cdot16^0 = 2560+16+14 = (2590)_{10}.$$

Переведення цілих чисел з десяткової системи числення в двійкову виконується послідовним діленням націло десяткового числа на основу $q$ системи числення.

Наприклад:

$$(66)_{10} \Rightarrow (231)_5$$



## Форми подання чисел в ЕОМ

Для того щоб ЕОМ могла оперувати з числами /цілими, натуральними, раціональними, ірраціональними/, потрібно було розробити стандартну форму запису чисел і створити загальну арифметику дій з ними.

Створено дві таких стандартних форми.

I. Форма з плаваючою комою:

$$A = \pm M \cdot 10^{p}, \qquad /I.9/$$

де $M$ – мантиса, $p$ – порядок числа /характеристика/.

Наприклад, число – 0,000I92 можна записати так:

$$-0{,}0I92 \cdot I0^{-2}; \quad -0{,}I92 \cdot I0^{-3}; \quad -0{,}00I92 \cdot I0^{-I}.$$

Цієї багатозначності можна уникнути, якщо мантиса числа задовольнятиме умову

$$0{,}I \leq |M| < I. \qquad /I.I0/$$

Запис числа у формі $A = \pm M \cdot I0^{p}$, де $0{,}I < |M| < I$, називають поданням його в нормалізованому вигляді.

2. Форма з фіксованою комою: положення коми, яка відділяє цілу частину числа від дробової, залишається незмінним для всіх чисел. Таке подання чисел дає змогу досить просто сконструювати ЕОМ з невеликою кількістю елементів.

У сучасних ЕОМ застосовують обидві списані форми подання чисел, використовуючи переваги обох режимів обчислень.

## I.3. Програмне забезпечення ЕОМ

Коли програму розміщено в пам"яті ЕОМ, остання може виконувати корисну роботу. Отже, потрібні різні за призначенням програми для ЕОМ. Сукупність таких програм називається системою програмного забезпечення /СПЗ/.

До складу СПЗ входять такі компоненти: операційна система /ОС/ і спеціальне програмне забезпечення /див. рис. І.4/. Операційна система призначена для планування і організації безперервного процесу обробки програм, керування даними, налагодження програм. ЕОМ виконує програми, які знаходяться в її пам"яті та записані в командах машини. Проте записати в текст від руки мовою машини дуже важко. Простіше написати програму однією з мов програмування і навчити ЕОМ розуміти і виконувати ці програми.

В ЄС ЕОМ для записування програм використовуються такі мови програмування: Асемблер, Алгол-60, Фортран-ІУ, РПГ, Кобол і ПЛ/І. Отже, ЕОМ має вміти перекладати програму, записану кожною з цих мов, на мову машини.

З цією метою створено два типи програм-перекладачів: а/ програма-інтерпретатор; б/ програма-компілятор.

Інтерпретатор читає вихідну програму частинами і, якщо помилок нема, відразу виконує відповідні дії.

Компілятор читає текст програми мовою програмування від початку до кінця і, якщо в ньому немає помилок, створює еквівалентну програму мовою машини. При наявності помилок ЕОМ "знімає" програму для їх виправлення.

Щоб полегшити введення програми в пам"ять ЕОМ, існує редактор програм - програма, що враховує правила запису програм мовою програмування. Якщо редактор програм поєднано з компілятором, ЕОМ може відразу перевіряти правильність програми.

Подальші етапи проходження програми зображено на рис. І.3: вихідна програма /вихідний модуль/ транслюється /перекладається компілятором чи інтерпретатором в коди машини/, після чого формується так званий об"єктний модуль. Програми об"єктного модуля зберігаються на зовнішній пам"яті ЕОМ, частіше на диску.

Наступна програма /редактор зв"язків/ охоплює об"єктні модулі програми і додає допоміжні алгоритми /підпрограми/. Формується завантажувальний модуль, який за допомогою програми-завантажувача вводиться /копіюється/ в оперативну пам"ять ЕОМ для реалізації.

У зовнішній пам"яті ЕОМ /диску/ зберігається не тільки машинна програма, а й інша інформація. Певна порція інформації, яка зберігається у зовнішній пам"яті, називається файлом. При виконанні задач на ЕОМ часто виникає потреба щодо організації, записування, зберігання, пошуку та читання файлів. Для реалізації відповідних дій застосовуються програми, які називають файловими системами.

Оскільки швидкодія ЕОМ досить велика, вона може одночасно виконувати кілька процесів. Щоб організувати таку багатопрограмну роботу, потрібні керуючі програми.

До програмного забезпечення ЕОМ входять також пакети прикладних програм /ППП/ і комплекс програм технічного обслуговування /КПТО/ /див. рис. I.4/. Пакети прикладних програм - це комплекс програмних засобів разом з документацією, необхідною для установлення й експлуатації ЕОМ. Вони застосовуються для розв"язування певних класів задач. Комплекс програм технічного обслуговування - це типові програми для налагодження, планування й діагностики функціонування апаратури ЕОМ.

## Основи алгоритмізації задач

Поняття алгоритму й програми належать до фундаментальних, первинних понять сучасної математики, які не визначаються через простіші, а подаються описово..

Слово "алгоритм" походить від запису латинською мовою імені видатного узбецького математика аль-Хорезмі /9 ст./, якому належить відкриття правил виконання арифметичних дій.

Що ж таке алгоритм? Правила додавання натуральних чисел можна вважати алгоритмом додавання цих чисел. Графічний спосіб ділення кута навпіл - також алгоритм. Інструкція одержання якоїсь нової речовини - також алгоритм. Отже, йдеться про те, як складну роботу подати через послідовність простіших дій /операцій/. Поняття алгоритму є результатом нашого прагнення до розбиття складних процесів на якомога простіші, доступні для виконання без будь-яких додаткових пояснень.

Зокрема, набули поширення так звані обчислювальні алгоритми, оскільки спеціально розроблені прикладною математикою чисельні методи дозволили звести дуже багато математичних операцій до арифметичних дій. Проте чітко описувані обчислювальні алгоритми - це аж ніяк не єдиний вид алгоритмів.

Графічний спосіб поділу кута навпіл - це алгоритм побудови. Метод одержання нової речовини - алгоритм-інструкція. Ці алгоритми набули широкого поширення в таких областях, як економіка та біологія. Алгоритм можна подати й у вигляді системи правил /наприклад, алгоритм добування кореня будь-якого степеня з числа/.

Отже, алгоритм можна подати в одному з таких виглядів: 1/ формула; 2/ система правил; 3/ інструкція щодо виконання побудов з лінійкою й циркулем.

Таким чином, під алгоритмом розуміємо систему формальних правил та умов, які чітко й однозначно визначають певні елементарні операції і порядок їх виконання, строго додержуючи якого завжди дістанемо шуканий розв"язок будь-якої задачі відповідного типу.

У наш час розширюється обсяг задач, які можна алгоритмізувати. Алгоритмізована задача в принципі вважається розв"язаною.

Зауважимо, проте, що існують класи задач, які не піддаються алгоритмізації. Наприклад, ЕОМ не може розв"язати таку, здавалося б, просту задачу: як відрізнити на фотографії собаку від кішки, дорослого собаку від цуценяти. Намагаючись скласти відповідний алгоритм, людина на сьогоднішньому рівні розвитку науки стикається з непереборними труднощами. Не вдалося й досі запрограмувати інтуїцію шахіста.

## Властивості алгоритмів

Кожний алгоритм має такі властивості:

I. Дискретність. Алгоритм записується у вигляді послідовності, окремих пунктів, що чітко відрізняються один від одного.

2. Доступність. Кожна вказівка алгоритму зрозуміла, однозначна, доступна для виконання.

3. Масовість. Алгоритм застосовний до розв"язування не якоїсь окремої, а всіх задач певного класу.

4. Результативність. Через скінченне число кроків алгоритм приводить до результату.

5. Формальність. Виконуючи будь-яку вказівку алгоритму, можна не вникати в її зміст, тобто перейти від усвідомленого розв"язування до чисто формального, механічного. Саме завдяки цьому при достатній деталізації алгоритму його виконання можна доручити й машині.

## Етапи розв"язування задачі на ЕОМ

Процес розв"язування задачі на ЕОМ включає в себе такі головні етапи:

I. Постановка задачі. На цьому етапі усвідомлюється суть задачі та кінцева мета її розв"язування.

2. Побудова математичної моделі. Шукана модель має бути адекватною досліджуваному процесу і якомога повніше відображувати його характерні риси.

3. Розробка чисельного методу розв"язування. Щоб реалізувати будь-яку задачу на ЕОМ, необхідно звести її до обчислювального алгоритму, який складається з арифметичних дій, виконуваних на ЕОМ. Розробкою чисельних методів, які зводять всі математичні операції до основних арифметичних дій, займаються спеціалісти з обчислювальної математики. А інженеру-практику, щоб розв"язати на ЕОМ свою задачу, потрібно вміти серед відомих чисельних методів вибрати необхідний.

4.Розробка алгоритму й побудова його схеми. Процес розв"язування задачі подається у вигляді послідовності найпростіших арифметичних і логічних операцій. Графічно його записують у вигляді спеціальної схеми.

5. Складання програми алгоритмічною мовою та її налагодження. Для розв"язування задач за розробленими чисельними методами складаються програми, які утворюють бібліотеки стандартних і пакети прикладних програм, а також програмні комплекси. Так, для розрахунку будівельних конструкцій розроблено програмний комплекс "Ліра". З його допомогою можна розрахувати як стержневі системи, так і пластини, плити, оболонки, масиви. Ефективне використання таких остаточно завершених і перевірених на практиці програм й розв"язування на ЕОМ різних технічних задач має велике народногосподарське значення.

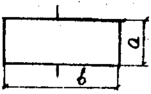
6. Аналіз отриманих результатів.

## Графічний спосіб запису алгоритмів

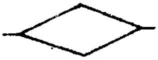
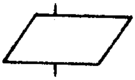
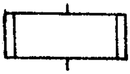
Найбільш наочний спосіб подання алгоритмів - їх графічне зображення у вигляді схем. Схема - це послідовність спеціальних символів /блоків/, кожний з яких вказує на виконання певних функцій, а також зв"язків між ними. У кожному символі міститься інформація про відповідні дії /табл. I.I/.

Таблиця I.I

Найчастіше вживані блоки алгоритмів

| Символ | Назва | Зміст |
|---|---|---|
| I | 2 | 3 |
|  | Арифметичний блок | Послідовність обчислювальних дій |

Закінчення табл. І.І

| І | 2 | 3 |
|---|---|---|
|  | Логічний блок | Перевірка дій |
|  | Блок введення/виведення даних | Введення вихідних даних; при виведенні - друкування результатів |
|  | Блок початку циклу | Початок обчислень |
|  | Блок обчислення за підпрограмою чи стандартною програмою | Виконання обчислень |

## І.4. Розробка алгоритмів структур

### Алгоритми лінійної структури

У лінійному алгоритмі блоки виконуються послідовно один за одним. Такий порядок виконання блоків називається природним.

Складаючи алгоритм, необхідно домагатися тощо, щоб задача була розв"язана з потрібною точністю та з найменшими затратами машинного часу.

Розглянемо приклади лінійних алгоритмів, розв"язавши такі задачі.

Задача І.І. Дано трикутник із сторонами $a$ , $b$ та кутом C між ними. Знайти третю сторону c і решту кутів /рис. І.5/.

Розв"язання. Скористаємося відповідно теоремами косинусів і синусів:

$$c^2 = a^2 + b^2 - 2\ ab \cos C;$$

$$\sin A/a = \sin C/c,$$

а також тотожністю

$$A + B + C = 180^\circ.$$

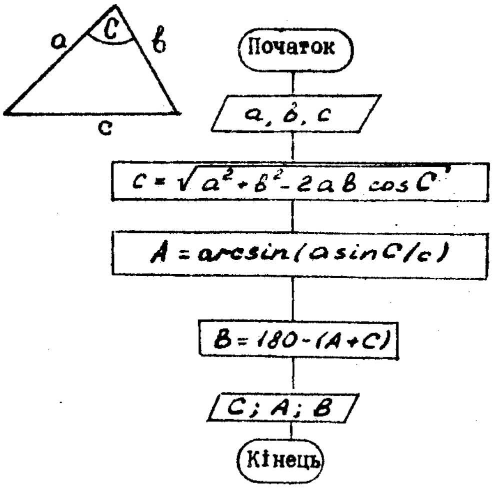


Рис. I.5. Схема лінійного алгоритму до задачі I.I

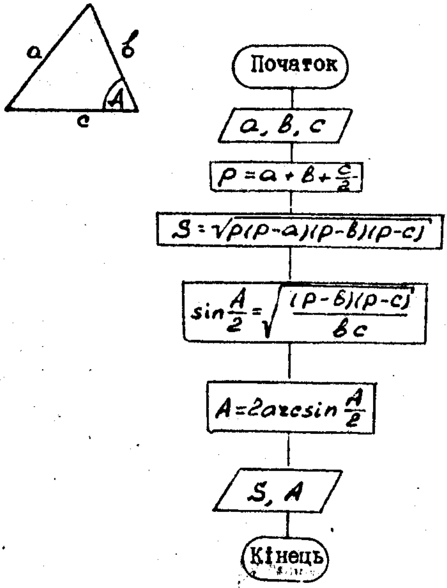


Рис. I.6. Схема лінійного алгоритму до задачі I.2

Задача I.2. Дано три сторони трикутника $a, b, c$. Знайти його площу та кут А /рис. I.6/.

Розв"язання. Скористаємося формулами:

$$p = (a + b + c)/2;$$

$$S = \sqrt{p(p-a)(p-b)(p-c)};$$

$$\sin\frac{A}{2} \sqrt{\frac{(p-b)(p-c)}{bc}}.$$

## Алгоритми з розгалуженою структурою

Якщо залежно від того, виконується якась логічна умова чи ні, обчислювальний процес відбувається за різними напрямами /обчислення виконуються за різними формулами/, то відповідний алгоритм називають розгалуженим.

На практиці порушувати природний порядок виконання блоків доводиться досить часто. Розглянемо приклади.

Задача I.3. Скласти алгоритм знаходження коренів квадратного рівняння $ax^2 + bx + c = 0$.

Розв"язання. Залежно від знака дискримінанта $D = b^2 - 4ac$ обчислювальний процес іде в одному з двох напрямів, тобто він розгалужується. Якщо $D \geqslant 0$, корені рівняння дійсні й обчислюються за формулами $x_{1,2} = O \pm p$ /рис. I.7, ліва вітка/; якщо $D < 0$, корені рівняння уявні й обчислюються за формулами $x_{1,2} = O \pm ip$ /рис. I.7, права вітка/.

Задача I.4. Розв"язати систему рівнянь другого порядку /рис. I.8/

$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 \end{cases}$$

методом Крамера.

Розв"язання.

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1}, \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1}.$$

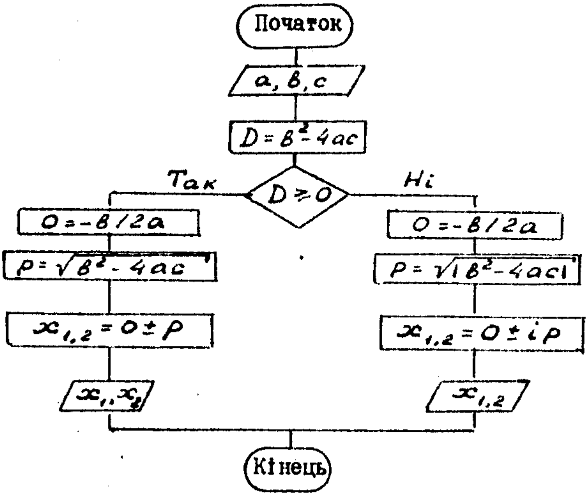


Рис. I.7. Схема розгалуженого алгоритму до задачі I.3

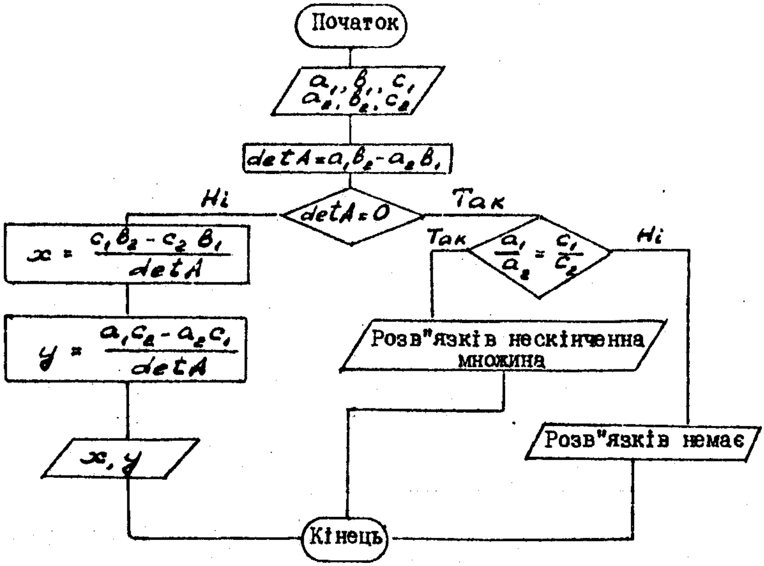


Рис. I.8. Схема розгалуженого алгоритму до задачі I.4

Можливий один з трьох випадків:

1/ система рівнянь має один розв"язок, якщо визначник матриці А, складений з коефіцієнтів при невідомих, не дорівнює нулю:

$$det A = det \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \neq 0;$$

2/ система має нескінченну множину,розв"язків, якщо $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$; $det A = 0$;

3/ не має розв"язків, якщо $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$; $det A = 0$.

Схему відповідного алгоритму показано на рис. I.8.

## Алгоритми циклічних структур

При розв"язуванні багатьох задач часто буває потрібно обчислювати за одними й тими самими математичними формулами, але при різних значеннях аргументів. Ці багато разів повторювані частини обчислювального процесу називають циклами. Кількість повторень кожного циклу може бути заданою до початку обчислень, а може бути й невідомою. Цикли, в яких кількість повторень заздалегідь невідома, називаються ітераційними.

Застосовуючи цикли при складанні алгоритмів, можна скоротити як схему, так і програму обчислювального процесу.

Величини, які змінюються в циклі від одного проходження до другого, називаються параметрами циклу.

Щоб організувати цикл, необхідно виконати такі дії:

1/ до початку циклу задати початкове значення його параметра;

2/ після кожного виконання циклу збільшити на одиницю його параметр;

3/ перевірити умову закінчення циклу і перейти до наступного його виконання, якщо ця умова не виконується, і вийти з циклу, завершивши його, якщо ця умова виконується.

Розглянемо приклади.

Задача I.5. Обчислити з точністю до 0,1 значення кореня рівняння $y = x - arctg\, x - \pi$.

Розв"язання. Шуканий корінь міститься на інтервалі (2, 5), де функція змінює знак. Якщо $x = 2$, $y < 0$. Схему відповідного алгоритму зображено на рис. I.9. Цикл, очевидно, є ітераційним.

Задача I.6. Записати в масив, який складається із 70 елементів, значення цих елементів, що дорівнюють одиниці.

Розв"язання. Щоб зменшити довжину схеми, доцільно застосувати блок початку циклу /рис. I.10/. Кількість повторень циклу відома заздалегідь.

Задача I.7. Для функції $y=(x-1)^n/n^3$ знайти $n$, при якому $y$ буде менше за $\mathcal{E}$ /$\mathcal{E}$ - дане число/.

Розв"язання. У даному разі $n$ набуває цілочислових значень. Схему відповідного алгоритму показано на рис. I.II. Цикл є ітераційним.

### Алгоритми із структурою вкладених циклів

Цикл, який містить у собі один або більше інших циклів, називається вкладеним. При цьому розрізняють зовнішній і внутрішній цикли. Зовнішній цикл охоплює решту циклів, які є внутрішніми. Параметри зовнішнього і внутрішнього циклів змінюються не одночасно. При якомусь фіксованому значенні параметра зовнішнього циклу параметри внутрішнього циклу набувають усіх своїх значень.

Задача I.8. Вивести на друк елементи головної діагоналі матриці $A(N,M)$.

Розв"язання. Зовнішній цикл змінюється за $i$ /індексом рядка/. Внутрішній цикл змінюється за $j$ /індексом стовпця/. Якщо $I=J$, виводиться на друк елемент головної діагоналі. Схему відповідного алгоритму зображено на рис. I.I2.

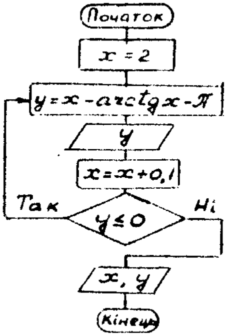


Рис. I.9. Схема циклічного алгоритму до задачі I.5

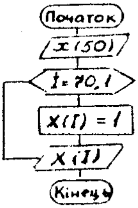


Рис. I.10. Схема циклічного алгоритму до задачі I.6

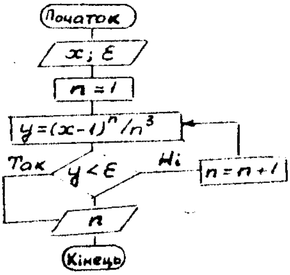


Рис. I.II. Схема циклічного алгоритму до задачі I.7

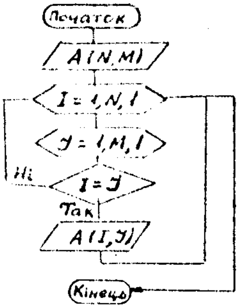


Рис. I.12. Схема алгоритму з вкладеними циклами до задачі I.8

## I.5. Мови програмування. Фортран-ІУ

З проникненням ЕОМ у практичну діяльність людини інтенсивно почали розвиватися мови програмування.

Мова програмування, або алгоритмічна мова, це спеціально розроблена штучна мова для запису алгоритмів. Завдяки таким мовам підготовка задачі та її розв"язання на ЕОМ значно спростилися. Тепер, оволодівши однією з мов програмування, до ЕОМ може звернутися конструктор, будівельник, біолог, економіст, астроном чи інший спеціаліст, і йому не буде потрібна допомога професіонального програміста.

Сьогодні налічується понад 1000 мов програмування, та лише близько десяти з них можна вважати загальновизнаними. Насамперед Алгол, Фортран, PL/I, Бейсік, Кобол та деякі інші. Переважне використання тієї чи іншої алгоритмічної мови зумовлюється її простотою, зручністю, наочністю, забезпеченістю пакетами прикладних програм тощо.

Зауважимо, що всі відомі мови програмування є предметно-орієнтованими.

Наприклад, мова Кобол зручна для записування алгоритмів розв"язування економічних і бухгалтерських задач. Існує група мов, за допомогою яких зручно записувати алгоритми розв"язування розрахункових задач, до яких саме й належать задачі, пов"язані з обчисленням будівельних конструкцій. Це Алгол, Фортран, PL/I, Бейсік.

Використовуватимемо далі мову Фортран /від англ. "перекладач формул/, створену раніше від усіх із вживаних тепер мов. Вона добре пристосована для наукових і математичних обчислень, проста у використанні, зручна при введенні/виведенні даних, забезпечена пакетами прикладних програм.

Можливості мови Фортран розширені завдяки її версії Фортран ІУ.

### Найпростіші конструкції мови Фортран

Згідно з прийнятою послідовністю реалізації задачі на ЕОМ програмуванню алгоритмічною мовою передує складання схеми алгоритму розв"язування даної задачі. Коли таку схему складено, програмувати за нею можна будь-якою алгоритмічною мовою, тобто схема того чи іншого конкретного алгоритму є основою для програмування.

Первинними елементами мови Фортран, з яких утворюються всі її конструкції, є строго фіксовані основні символи, подані на рис. І.13. Як додаткові символи можуть використовуватися букви російського алфавіту, причому набір додаткових символів в алгоритмічній мові строго не фіксується. Він залежить від складу обладнання ЕОМ. Додаткові символи застосовуються для побудови символьних констант.

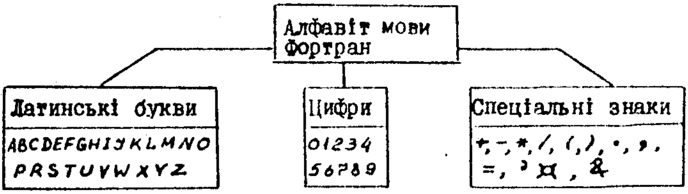


Рис. І.13. Основні символи мови Фортран

Основними конструкціями мови є оператори, які використовуються для описування програми за складеним алгоритмом, причому на етапі програмування оператори в програмі потрібно записувати в такій самій послідовності, у якій ідуть блоки в схемі алгоритму, не порушуючи структури їх зв"язків.

Найпростіші конструкції мови такі: 1/ константи; 2/ змінні; 3/ функції; 4/ вирази. Вони записуються за допомогою спеціальних символів /див. рис. І.13/.

## Константи

У мові Фортран константи використовуються для зображення сталих значень величин. Розрізняють шість типів констант: 1/ цілі; 2/ реальні; 3/ комплексні; 4/ логічні; 5/ символьні; 6/ шістнадцяткові. Усі константи, крім символьних і комплексних, можуть бути стандартної /4 байт/ і нестандартної довжини.

І. Цілі константи /INTEGER / – послідовність цифр з відповідним знаком /"+" або "-"/. Зображуються в ЕОМ точно. Цілі константи стандартної довжини /І4/ займають в пам"яті слово, їх значення за модулем не більше 2; такі константи нестандартної /12/ довжини займають в пам"яті півслова; інтервал їх зміни /- /$2^{12}$-1/,+/$2^{12}$-1/.

Модуль цілої константи для кожної ЕОМ має свою верхню межу. Для ЕОМ ЄС ця межа становить 2147483647.

2. Реальні константи /*REAL*/ – числа, які мають цілу й дробову частини. Зображуються в ЕОМ наближено. Можуть записуватися в трьох формах: *F*, *E* і *D* . Форма *F* основна – ціла частина числа відокремлюється від дробової крапкою. Наприклад, -217.3; 4.88. Форми *E* і *D* включають експоненту, тобто число подається у вигляді його порядку і мантиси. Наприклад, форма E: 0.87E-4; 12E3; форма *D* : 2.5*D*-14; 18.923147817*D*+21.

Реальні константи стандартної довжини /*R* 4/ займають в пам"яті слово, значення їх за модулем не перевищують $10^{75}$. Наприклад, нехай реальні константи, записано у формі *F* : 3.141529; -0.8034. Вони належать до типу *R* 4, точність їх гарантована до сьомого знака /основна константа складається не більш як із семи цифр/. А реальні константи, записані у формі E: 18.4321E+1; 0.1843321E+03, мають довжину *R*4, мантиси їх мають включати не більш як сім цифр.

Реальні константи нестандартної довжини /*R* 8/ займають в пам"яті два слова; основна константа складається з шістнадцяти цифр; константа з експонентою має мантису, що включає не більш як шістнадцять цифр.

Форми E і *D* зручні для зображення великих і малих чисел.

3. Комплексні константи /*COMPLEX*/ записуються у вигляді пари реальних, поданих у формах *F* , E, *D* констант, розділених комою і взятих в круглі дужки: $(a_1, a_2)$ де $a_1, a_2$ – відповідно дійсна і комплексна частина розглядуваного числа.

Комплексні константи стандартної довжини /C8/ займають в пам"яті 8 байт: 4 на дійсну частину і 4 на комплексну частину.

Комплексні константи нестандартної довжини /C16/ займають 4 слова: 2 дійсна і 2 комплексна частина.

Наприклад, запишемо комплексне число 4 + 2 $i$.

У формі *F* : /4.,2./ – довжина /C8/;

у формі E: /4.E0, 2.E0/ – довжина C8/;

у формі *D* : /4.*D*0, 2.*D*0/ – довжина /C16/.

4. Логічні константи /*LOGICAL*/ набувають одного з двох значень – "істина" і "неправда". Вони записуються у вигляді послідовності символів ".*TRUE*." та ".*FALSE*.". Стандартна довжина становить 4, а нестандартна 2 байт.

5. Символьні константи являють собою послідовності будь-яких алфавітно-цифрових та спеціальних символів, довжиною, що не перевершує 255 символів. Записуються в одному з двох форматів:

I/ з покажчиком довжини /$nHS$/ /конструкція $nH$ називається покажчиком довжини символьної константи; $S$ – символьна константа; $n$ – число символів в послідовності $S$ ; H – специфікація символьної константи/; наприклад,

26H⌣ВЛАСНІ ЧИСЛА МАТРИЦІ

2/ без покажчика довжини / '$S$' /; наприклад,

'ВЛАСНІ ЧИСЛА МАТРИЦІ'

6/ Шістнадцяткові константи /використовуються тільки для зображення початкових значень величин/ записуються у вигляді

$$Z a_1 a_2 \ldots a_n ,$$

де $Z$ – основний символ, $a_i$ / $i$ = I,2,...,$n$/ – шістнадцяткові цифри.

Шістнадцятковий алфавіт складається із символів $\{0, I, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, \}_{16}$ . Тому шістнадцяткові константи можна, наприклад, записати так:

$ZF$ /означає послідовність бітів IIII/; .

$Z3B$/означає послідовність бітів 00II I0II/.

## Змінні

Для позначення змінних, масивів, функцій, процедур тощо використовуються ідентифікатори. Ідентифікатор являє собою послідовність символів з основного алфавіту, який починається обов"язково з букви. Кількість символів в ідентифікаторі має не перевищувати шести, наприклад: $X$, $XY$, $ALFA$ 6I. Ідентифікатори є словами користувача і не повинні збігатися зі службовими словами.

Змінною називають величину, яка набуває різних значень. Змінні бувають прості та з індексами.

Прості змінні позначаються ідентифікаторами. Вони в кожний момент часу обчислювального процесу мають одне й те саме значення.

Змінні з індексами є елементами масивів, тобто вони являють собою короткий запис впорядкованої сукупності змінних з індексами. Наприклад, змінна $X(I, J)$ описує масив $X$ , який має $I$ рядків та $J$ стовпців; змінна $K(3*N+$I,4$)$ позначає елемент, що міститься на перетині $(3N+$I$)$-го рядка і 4-го стовпця. Змінні з індексами позначаються ідентифікатором масиву, за яким в круглих дужках через кому може бути записано підряд від одного до семи індексів. Індексом може бути константа, змінна або арифметичний вираз, що набуває цілочислового значення. Індекс в алгоритмічній мові Фортран не може бути від"ємним числом.

## Описування типу змінних

У програмах, записаних мовою Фортран, можуть використовуватися величини наступних типів: цілі /INTEGER/, реальні /REAL/, комплексні /COMPLEX/, логічні /LOGICAL/, подвійної точності /DOUBLE PRECISION/.

Тип величини визначається за допомогою спеціальних невиконуваних операторів. Існують три способи описування типу величин.

### I. Описування типу змінних за замовчуванням /погодженням/.

Якщо ідентифікатор змінної починається з букв I,J,K,L,M,N, то ця змінна цілого типу, а якщо з інших букв алфавіту Фортрану, то ця змінна реального типу. В обох випадках ці змінні стандартної довжини /4 байт/.

Тип змінних в масивах за замовчуванням описується за допомогою оператора DIMENSION. При цьому вказується ідентифікатор масиву і в круглих дужках найбільше значення індексів. Наприклад, DIMENSION A (4,3), Z (4) описує масив A, який включає 4 рядки та 3 стовпці, і вектор Z, який має 4 рядки.

### 2. Описування типу змінних операторами явного описування.

Оператор явного описування має вигляд:

$$\sqcup\, t * s\,a_1 * s_1(k_1), a_2 * s_2(k_2), \ldots, a_n * s_n(k_n), \qquad /I.10/$$

де $t$ – покажчик типу даних /як покажчик типу даних використовуються ключові слова: INTEGER, REAL, COMPLEX, LOGICAL, DOUBLE PRECISION/; $s$ – загальний покажчик довжини змінної в байтах; $s_i$ – покажчик довжини величини $a_i$; $a_i$ – ідентифікатор змінної або масиву; $k_i$ – розмірність масиву.

Якщо $s_i$ відсутнє, довжина змінної знаходиться за $s$; якщо $s$ відсутнє, змінна або масив мають стандартну довжину. Наприклад, оператор

INTEGER Y(10),A*2(5)

описує цілі величини: вектор Y, який складається з 10 елементів стандартної довжини, і вектор $A$ із 5 елементів нестандартної довжини /2 байт/. Оператор

$$REAL*8Z, A(10,10), K*4(3,2)$$

описує реальні змінні: $Z$ довжиною 8 байт, матриця $A$ розмірністю 10x10, кожний елемент якої має довжину 8 байт, матрицю $K$ розмірністю 3x2, елементи якої мають стандартну довжину 4 байт.

3. Описування типу змінних оператором.

Відповідний оператор має вигляд:

$$\sqcup IMPLICIT \sqcup t_1*s_1(a_1), t_2*s_2(a_2), \ldots, t_n*s_n(a_n), \qquad /I.11/$$

де $t_i$, $s_i$ – покажчик відповідно типу та довжини змінних; $a_i$ – ідентифікатори змінних, яким присвоюються тип $t_i$ і довжина $s_i$.

Наприклад, оператор

$$IMPLICIT \sqcup REAL*4(K,L), INTEGER(A\text{-}D,O)$$

вказує, що змінні $K$, $L$ – реальні стандартної довжини 4 байт, змінні від $A$ до $D$ і $O$ – цілі стандартної довжини 4 байт.

До програми одночасно можуть входити всі три види операторів описування типу змінних: явний, неявний та за замовчуванням. Вони можуть і суперечити один одному. Перевага завжди надається оператору явного описування.

## Функції

Звернення до функцій відбувається через покажчик функції, який має вигляд

$$f(x_1, x_2, \ldots, x_n), \qquad /I.12/$$

де $f$ – ім"я функції; $x_i$ – її аргументи, які називаються фактичними параметрами.

Іменем функції може бути будь-який ідентифікатор, а аргументами можуть бути константи, імена змінних, вирази, імена масивів, функцій та процедур. Цілий ряд функцій, що зустрічаються найчастіше, подаються як складова частини мови Фортран. Такі функції називаються вмонтованими. За ними закріплені певні імена. Деякі з них наведені в табл. 1.2.

Таблиця 1.2

Вмонтовані функції мови Фортран

| Функція | Математичне позначення | Ім"я | Тип аргументу |
|---|---|---|---|
| Модуль | $y = \|x\|$ | IABS<br>DABS<br>ABS | I4<br>R8<br>R4 |
| Квадратний корінь | $y = \sqrt{x}$ | SQRT<br>DSQRT | R4<br>R8 |
| | $y = \sqrt{z}$ | CSQRT<br>CDSQRT | C8<br>C16 |
| Експонента | $y = e^{x}$ | EXP<br>DEXP | R4<br>R8 |
| Натуральний логарифм | $y = \ln x$ | ALOG<br>DLOG | R4<br>R8 |
| Десятковий логарифм | $y = \log_{10} x$ | ALOG10<br>DLOG10 | R4<br>R8 |
| Синус | $y = \sin(x)$ | SIN | R4 |
| Косинус | $y = \cos(x)$ | COS | R4 |
| Тангенс | $y = tg(x)$ | TAN | R4 |
| Котангенс | $y = ctg(x)$ | COTAN | R4 |
| Арксинус | $y = arcsin(x)$ | ARSIN | R4 |
| Арккосинус | $y = arccos(x)$ | ARCOS | R4 |
| Арктангенс | $y = arctg(x)$ | ATAN | R4 |

Примітка. Вираз $z = x_1 + x_2 i$ – комплексне число в алгебраїчній формі.

## Вирази

Вирази бувають арифметичні та логічні.

Арифметичні вирази компонуються з арифметичних операндів /константи, змінні /їх імена/, покажчики функцій/, знаків арифметичних операцій і круглих дужок. Результатом обчислення арифметичного виразу буде числова константа /ціла, реальна або комплексна/. Порядок виконання операцій у виразі задається дужками. Якщо їх немає, операції виконуються в такому порядку: 1/ обчислення функцій; 2/ піднесення до степеня; 3/ множення або ділення; 4/ додавання або віднімання.

Дії одного й того самого порядку виконуються зліва направо.

Знаходження кореня замінюється піднесенням до дробового степеня.

Приклади алгебраїчних виразів:

A*(B+SQRT(x)**2)
X(4+2*Y+DLOG10(x)

Найпростіші логічні вирази - це відношення виду $a\ \sigma\ b$, де $a$ і $b$ - арифметичні вирази цілого чи реального типу; $\sigma$ - позначення одного з основних відношень, що являє собою послідовність чотирьох символів:

.GT. – >;    .LE. – ≤;
.GE. – ≥;    .EQ. – =;    /I.13/
.LT. – <;    .NQ. – ≠.

Приклад відношення: X.GT.5.6; Y.EQ.0.

Логічний вираз може набувати одного з двох значень: .TRUE. /істина/ або .FALSE. /неправда/.

Логічні вирази складаються з логічних операндів /константи, змінні, покажчики функцій логічного типу, відношення/ позначень логічних операцій та круглих дужок.

Позначення кожної з логічних операцій - це послідовність символів:

.NOT. /заперечення, ні/;
.AND. /логічне множення, і/;    /I.14/
.OR. /логічне додавання, або/.

Приклади логічних виразів:

X.GT.0.AND.Y.GT.0
X.GT.I.OR.Y.GT.0

З'ясовуючи значення логічного виразу, спочатку розглядають всі відношення, а далі виконують логічні операції в такому порядку: .NOT., .AND., .OR.. Правила виконання логічних операцій подано в табл. I.3.

Таблиця I.3

Правила виконання логічних операцій

| *A* | *B* | *.NOT.A* | *A.AND.B* | *A.OR.B* |
|---|---|---|---|---|
| Істина | Інстина | Неправда | Істина | Істина |
| Істина | Неправда | Неправда | Неправда | Істина |
| Непранда | Істина | Істина | Неправда | Істина |
| Неправда | Неправда | Істина | Неправда | Неправда |

## I.6. Оператори. Програмні модулі. Структура програмного модуля. Типи модулів

Оператор – основна конструкція мови. За допомогою операторів описуються дії, які потрібно виконати, або задаються властивості величин. На перфокартці оператори розміщуються починаючи із сьомої позиції. Оператор програми може мати мітку. Це ціла константа без знака, що складається з небільш як з п"яти цифр. Розташовується мітка з першої позиції по п"яту. Область дії мітки – програмний модуль.

Програмний модуль являє собою спеціальним чином оформлену послідовність операторів. Він реалізує логічно самостійну частину обчислювального процесу.

Програми, записані алгоритмічною мовою Фортран, мають модульну структуру, тобто складаються з одного /головного/ або кількох зовнішніх один щодо одного програмних модулів, серед яких є головний.

Виконання програми починається з головного модуля.

Структура програмного модуля:

Оператор – заголовок
Невиконувані оператори } Тіло програмного модуля
Виконувані оператори
*END*

У Фортрані розрізняють чотири типи програмних модулів, кожний з яких, крім головного, позначається відповідним оператором-заголовком:

головний модуль /не має оператора-заголовка/;
підпрограма-функція /оператор-заголовок *FUNCTION*/;
підпрограма-процедура /оператор-заголовок *SUBROUTINE*/;
підпрограма задання початкових значень /оператор-заголовок *BLOCK-DATA*/.

## Класифікація операторів

Оператори мови Фортран поділяються на виконувані та невиконувані.

Виконувані оператори вказують дію і /або/ порядок її виконання.

Виконувані оператори бувають таких видів:

1/ введення/виведення;

2/ присвоювання;

3/ управління: безумовного переходу - GOTO; умовного переходу - IF ; циклу - DO ; виклику підпрограми - CALL ; повернення з підпрограми - RETURN ; допоміжні - CONTINUE, STOP.

Невиконувані оператори інформують компілятор про величини та їх властивості. У процесі трансляції машинних команд не створюють.

Невиконувані оператори бувають таких видів:

1/ описування:
змінних - IMPLISIT, INTEGER, REAL, COMPLEX, LOGICAL, DOUBLE PRECISION
масивів - DIMENSION
початкових значень - DATA
формату - FORMAT
2/ розподілу пам"яті - COMMON
3/ заголовку:
підпрограми-функції - FUNCTION
підпрограми-процедури - SUBROUTINE
підпрограми задання початкових значень - BLOCK-DATA
4/ імені зовнішніх підпрограм - EXTERNAL
5/ додаткової точки входу - ENTRY
6/ закінчення - END

## Загальні області пам"яті

Оскільки обсяг оперативної пам"яті ЕОМ обмежений, інколи виникає потреба в одному і тому самому місці зберігати кілька змінних або масивів. Це можна зробити, використавши оператор COMMON. За його допомогою створюються загальні області і вказуються змінні та масиви.

Загальна область - спеціальна область пам"яті для збереження даних. Ця область доступна кільком програмним модулям однієї й тієї самої програми. Вид оператора:

$$\mathrm{COMMON}\,/z^{1}/a_1^{(1)}(k_1)^{(1)},\dots,a_n^{(1)}(k_n)^{(1)}/\dots/\dots/z^{m}/a_1^{(m}(k_1)^{(m)},\dots,a_n^{(m)}(k_n)^{(m)}, \qquad /I.I5/$$

де $z^{1}\dots z^{m}$ – ідентифікатор /ім"я загальної області/; якщо імені немає, то дві похилі (//) означають неіменовану загальну область; $a_1^{(m)}\dots a_n^{(m)}$ – список змінних, розміщених в загальній області з іменем $z^{m}$ або в неіменованій області, якщо $z$ відсутнє; $k_n^{(m)}$ – список меж масиву, якщо його розмірність в інших операторах не описана. Погоджені масиви повинні мати однакову розмірність. Наприклад, нехай маємо розподіл пам"яті, показаний у табл. I.4.

Таблиця I.4

Розподіл пам"яті

| Модуль | Загальна область | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | неіменована | | | | | | | з іменем IPA | | |
| Головна програма | Z(1) | Z(2) | Z(3) | Z(4) | A | B | C | X | Y | W1 |
| Підпрограма | Y(1,1) | Y(1,2) | Y(2,2) | Y(2,1) | W | L(1) | L(2) | HE | S3 | V1 |

Тоді

```
COMMON Z(4), A /IRA/X,Y,W1 //B,C        Головна програма
COMMON V(2,2),W /IRA/HE,S3,V1 // L(2)   Підпрограма
```

## Введення і виведення даних

### Введення і виведення простих змінних

Для введення/виведення значень простих змінних використовуються оператори:

```
  READ (m,n) S
  WRITE (m,n) S
n FORMAT (c)
```

де $n$ – мітка оператора FORMAT ; $S$ – список змінних для введення/виведення; $m$ – номер пристрою введення/виведення /у ОС ЄС ЕОМ для пристрою введення з перфокарток $m = 5$, а для пристрою друкування – $m = 6$/; $c$ – список специфікацій, що обумовлює формат даних та їх розміщення на картці при введенні та на папері при виведенні. Часто застосовуються специфікації:

$aIW$ – введення/виведення змінних цілого типу;

$aFW.d$ – введення/виведення реальних змінних в основній формі;

aEw.d,aDw.d – введення/виведення реальних змінних з експонентою в формах E і D ; aX – пропуск позицій;

aH<...> – введення/виведення текстових констант;

aLw – введення/виведення логічних констант;

w – загальна кількість відведених позицій;

d – кількість позицій під дробову частину;

a – коефіцієнт повторення специфікації.

Наприклад, введення значень змінних M = 100, J = 20, A = = -178,13; C = -2,78E-4 може бути таким:

```
  READ (5,4) M,J,A,C
4 FORMAT(2I3,F8.2,E9.2)
```

Виведення цих даних можна організувати так:

```
  WRITE(6,5) M,J,A,C
5 FORMAT(7X,3H␣M=,I3,3H␣J=,I3,'␣A=',F8.2,'␣C=',E9.2)
```

На папері буде віддруковано:

```
␣␣␣␣␣␣␣␣M=100␣J=20␣A=␣-178.13␣C=␣-2.78E-4
```

## Введення і виведення масивів

Введення/виведення значень елементів масивів можна організувати трьома способами:

1/ масив вводиться/виводиться цілком:

```
  DIMENSION A(100)
  READ (5,1)A
1 FORMAT (10F8.4)
```

2/ введення/виведення в циклі:

```
  DIMENSION A(100)
  DO 2I = 1,100
  READ (5,1) A(I)
1 FORMAT (F8,4)
```

3/ використання автоматичної індексації:

```
  READ (5,1)(A(I),I=1,100)
1 FORMAT (10F8.4)
```

Запис виду

```
  DIMENSION B(10,8)
  . . . . . . . . . . . .
  WRITE (6,8)((B(I,J),J=1,8),I=1,10)
8 FORMAT(8F10.3)
```

організує виведення матриці B/10,8/ по 8 елементів у рядку.

## Оператор задання початкових значень

За допомогою оператора DATA можна надати початкового значення змінним та масивам. Перед виконанням програми послідовним елементам списку змінних надають послідовних значень із списку констант в порядку їх запису зліва направо.

Вид оператора:

$$DATA\ V_1,\ldots,V_k/\tau_1*d_1,\ldots,\tau_m*d_m/ \qquad /I.16/$$

де $V_k$ - список змінних чи масивів; $d_m$ - список констант /за типом узгоджується зі змінними/; $\tau_m$ - повторювач константи.

Приклад:

```
DIMENSION Z(4),J(2)
LOGICAL O
DATA Z,J/0.7,2*0.0,1.8,32,0.9/,O/.TRUE./
```

## I.7. Організація програм

### Програми лінійної структури

Для записування лінійних програм використовуються такі оператори: присвоєння, введення/виведення, STOP, FORMAT, END.

Оператор присвоєння має вигляд

$$v = a,$$

де $v$ - ідентифікатор обчислювальної величини; $a$ - арифметичний або логічний вираз.

Якщо $a$ - арифметичний вираз, то $v$ набуває числового значення; якщо $a$ - логічний вираз, $v$ набуває логічного значення.

Оператор STOP завершує виконання програм. Оператор END означає кінець програми.

Приклад I.I. Скласти програму розв"язування задачі I.I.

| Величина | Ідентифікатор |
|---|---|
| a | A |
| b | B |
| c | C |
| <A | AUGOL |
| <B | BUGOL |
| <C | CUGOL |

Програма.

```
Введення    {   READ(5,1) A,B,CUGOL
вихідних    { 1 FORMAT(3F10.3)
даних           C=SQRT(A*A+B*B-2*A*B*COS(CUGOL))
                AUGOL=ARSIN(A*SIN(CUGOL)/C)
                BUGOL=180-(AUGOL+CUGOL)
Друкування  {   WRITE(6,2) C,AUGOL,BUGOL
результатів { 2 FORMAT(3F10.3)
                STOP
                END
```

<u>Приклад I.2</u>. Скласти програму розв"язування задачі I.2.
Програма.

```
      READ (5,1) A,B,C
  1   FORMAT (3F10.3)
      P=(A+B+C)/2
      S=SQRT(P*(P-A)*(P-B)*(P-C))
      AUGOL=2*ARSIN(SQRT((P-B)*(P-C)/B*C))
      WRITE (6,2) S,AUGOL
  2   FORMAT(2F10.3)
      STOP
      END
```

## Програми розгалуженої структури

Оператори керування існують для організації переходу обчислювального процесу за різними напрямами.

Оператор безумовного переходу *GO TO* організує безумовний перехід від оператора з міткою $n$. Оператор, який йде за *GO TO*, обов"язково повинен мати мітку, інакше він не виконуватиметься.

Оператор умовного переходу змінює природний порядок виконання операторів після перевірки деякої умови. Розрізняють арифметичний і логічний оператори умовного переходу.

Арифметичний оператор умовного переходу записується у вигляді

$$IF(a)\ n_1, n_2, n_3,$$

де $a$ – арифметичний вираз; $n_1, n_2, n_3$ – мітки.

Якщо $a < 0$, керування передається оператору з міткою $n_1$; якщо $a = 0$, керування переходить до оператора з міткою $n_2$; якщо $a > 0$, здійснюється перехід до оператора з міткою $n_3$. Оператор, який йде за арифметичним оператором умовного переходу, повинен мати мітку.

Логічний оператор умовного переходу має вигляд:

$$IF\ (B)\ S,$$

де $B$ – логічний вираз; $S$ – будь-який діючий оператор, крім іншого логічного оператора умовного переходу та оператора циклу.

Якщо логічний вираз у круглих дужках має значення .TRUE., виконується оператор $S$ і оператор, наступний за $IF(B)S$ /якщо оператор $S$ не є керівним і не передасть керування іншому оператору/.

Якщо логічний вираз у круглих дужках має значення .FALSE., то оператор $S$ пропускається і виконується наступний за $IF(B)S$ оператор.

Приклад І.3. Скласти програму розв"язування задачі І.3.
Програма.

```
     READ (5,7) A,B,C
   7 FORMAT(3F8.3)
     D=B**2-4.*A*C
     Z=2.*A
     O=-B/Z
     P=SQRT(ABC(D))/2
     IF (D) 2,3,3
   2 WRITE(6,4)O,P
     GOTO5
   3 X1=O+P
     X2=O-P
     WRITE (6,4)X1,X2
   4 FORMAT(2F14.5)
   5 STOP
     END
```

Приклад І.4. Скласти програму розв"язування задачі І.4.
Програма.

```
     READ(5,1) A1,B1,C1,A2,B2,C2
   1 FORMAT (6F13.3)
     DET=A1*B2-A2*B1
     IF(DET.EQ.0.) GOTO2
     X=(C1*B2-C2*B1)/DET
     Y=(A1*C2-A2*C1)/DET
     WRITE (6,3) X,Y
   3 FORMAT(4X,'X=',F10.4,2X,'Y=',F10.4)
     GO TO 4
```

```
2   IF((A1/A2).EQ.(C1/C2)) GO TO 5
    WRITE (6,6)
6   FORMAT (3X,'СИСТЕМА  НЕСОВМЕСТИМА')
    GO TO 4
5   WRITE (6,7)
7   FORMAT(3X,'РЕШЕНИЙ  БЕСЧИСЛЕННОЕ МНОЖЕСТВО')
4   STOP
    END
```

## Програми циклічної структури

Цикл можна організувати двома способами.

I. Якщо цикл ітераційний, перед ним задається початкове значення його параметра /за допомогою оператора присвоєння/; потім обчислюється поточне значення цього параметра /за допомогою оператора присвоєння/ і, нарешті, перевіряється умова виконання циклу /за допомогою логічного виразу/; керування циклом здійснюється операторами керування.

Приклад I.5. Скласти програму розв"язування задачі I.5.
Програма.

```
      X = 2
100   Y = X - ATAN(X) - 3,14159
      X = X + 0.1
      IF (Y.LT.0) GOTO 100
      WRITE (6,1) X
  1   FORMAT (F8.3)
      STOP
      END
```

← Задання початкового значення параметра циклу

Обчислення поточного значення параметра циклу

Перевірка умови повторення циклу й управління циклом

Приклад I.6. Скласти програму розв"язування задачі I.7.
Програма.

```
    READ (5,1) X,E
1   FORMAT (2F18.5)
    N = 1
5   IF((X+1)**N/N**3-E) 3,2,2
2   N = N+1
    GOTO 5
3   WRITE (6,4) N
4   FORMAT (I6)
    STOP
    END
```

Цей спосіб організації циклу зручний, якщо цикл ітераційний.

2. Якщо число повторень циклу відоме, зручно скористатись оператором виду

$$DO \sqcup N \sqcup I = m_1, m_2, m_3,$$

де $N$ – мітка останнього оператора циклу; $DO$ – оператор циклу; $m_1$ – початкове значення параметра циклу; $m_2$ – кінцеве значення параметра циклу; $m_3$ – крок зміни параметра циклу. Якщо $m_3 =$ I, це значення можна не вказувати. Останнім оператором циклу може бути будь-який /крім умовного логічного оператора *IF*, який не містить в собі керуючого оператора безумовного переходу *GO TO* / оператор. У такому разі для позначення кінця циклу використовується оператор *CONTINUE*, який має мітку.

Вхід до циклу – тільки через оператор *DO*. Вихід можна здійснити двома способами: I/ натуральним, коли параметр циклу досягнув кінцевого значення; 2/ за допомогою керуючого оператора.

Приклад I.7. Скласти програму розв"язування задачі I.6.

Програма.

```
      DIMENSION X(20)
      READ (5,1) X
    1 FORMAT(10F8.3)
      DO 2 I = 1,20
      X(I) = 1.
    2 WRITE (6,3) X
    3 FORMAT (10F8.3)
      STOP
      END
```

## I.8. Підпрограми

При розв"язуванні багатьох задач часто виникає потреба багато разів виконувати одні й ті самі обчислення при різних аргументах. Доцільно виділити ці обчислення в підпрограму, де вони будуть виконуватися при формальних параметрах.

Звертання до підпрограми здійснюється з основної програми при заданих значеннях фактичних параметрів. Функція, яка обчислюється в підпрограмі, називається зовнішньою функцією.

Розрізняють оператор-функцію, підпрограму-функцію, підпрограму-процедуру, підпрограму задання початкових значень.

## Оператор-функція

Це підпрограма, що складається з одного оператора

$$F(a_1, a_2, \dots, a_n) = E, \qquad /1.17/$$

де $F$ – ім"я функції, яку знаходять /ідентифікатор/; $a_i$ – формальні параметри /прості змінні без індексів/; $E$ – вираз, арифметичний або логічний.

Звертання до оператора-функції виконується за допомогою покажчика функції

$$F(b_1, b_2, \dots, b_n), \qquad /1.18/$$

де $F$ – ім"я функції; $b_i$ – фактичні параметри.

Формальні й фактичні параметри узгоджуються за кількістю, порядком знаходження й типом.

Оператор-функція розміщується в головній програмі перед першим виконуваним оператором, але після всіх інших невиконуваних операторів.

Приклад 1.8. Скласти програму обчислення функції

$$Z = \frac{e^{a^2-1}}{2-e^{b^2}} + \frac{0,8}{4\sqrt{e^{c^2+1}}},$$

використовуючи оператор-функцію.

Програма.

```
      E(X,D)=EXP(X*X+D)
      READ(5,1) A,B,C
    1 FORMAT(3F5.2)
      Z=E(A,-1)/2-E(B,0)+0.8/(4*SQRT(E(C,1.))
      WRITE(6,2) Z
    2 FORMAT(E12.4)
      STOP
      END
```

## Підпрограма-функція

Використовується для обчислення лише одного значення, яке присвоюється імені підпрограми. Може містити в собі будь-яку кількість операторів. Структура підпрограми-функції

```
t FUNCTION F*S(a1,a2,...,an)
 < Оператори підпрограми >                    /1.19/
  F = ...
 < Оператори підпрограми >
 RETURN
 END
```

де $t$ – покажчик типу функції /REAL,INTEGER,LOGICAL,COMPLEX/; якщо цей покажчик відсутній, то тип функції знаходиться за узгодженням; $*S$ – одна з допустимих довжин для вказаного типу функції, якщо вона не зазначена, то довжина стандартна;

RETURN – оператор повернення в основну програму;

$F$ – ім"я підпрограми, ідентифікатор; $a_i$ – формальні параметри /змінні, імена масивів або підпрограм/; $F$ – оператор присвоєння, в лівій частині якого стоїть ім"я підпрограми.

Підпрограма-функція – незалежний програмний модуль, оформляється у вигляді окремої програмної одиниці і розміщується після основної програми /після оператора END/.

Звертання до підпрограми-функції можна здійснити за допомогою покажчика функції $F(b_1, b_2, \ldots, b_n)$, де $F$ – ім"я підпрограми-функції /ідентифікатор/, $b_i$ – фактичні параметри /константи, прості або індексовані змінні, масиви, імена інших підпрограм/. Фактичні параметри узгоджуються за кількістю, порядком розміщення, видом і типом з відповідними формальними.

Приклад 1.9. Скласти програму обчислення функції $N = n!$

Програма

```
      FUNCTION␣FN(N)
      FN = 1.
      IF(N.EQ.0)GO␣TO␣4
      DO␣2␣I=1,N
    2 FN=FN*I
    4 RETURN          Оператор повернення до головної
      END             програми
```

## Підпрограма-процедура

Використовується для обчислення в підпрограмі кількох вихідних результатів. Структура підпрограми-процедури:

```
SUBROUTINE F(a1,a2,...,an)
<Оператори підпрограми >                    /1.20/
RETURN
END
```

де $F$ – ім"я підпрограми-процедури /ідентифікатор/; $a_i$ – формальні параметри /ідентифікатори змінних, масивів, програм/;

RETURN – оператор виходу з підпрограми;

END – кінець тексту підпрограми.

Спосіб звертання до підпрограми:

$$CALL\_F \quad \text{або} \quad CALL\ F\,(x_1, x_2, \ldots, x_n),$$

де $x_i$ – фактичні параметри, узгоджені за кількістю, порядком розміщення, видом і типом з відповідними формальними.

Підпрограма-процедура – це незалежний програмний модуль, який розміщується після основної програми /після оператора *END*/.

Обмін даними між основною програмою і підпрограмою проходить за допомогою списків-параметрів або через так звані загальні області /*COMMON*/.

В алгоритмічній мові Фортран передбачена можливість створення загальної області – спеціальної області оперативної пам"яті для зберігання даних, доступних кільком програмним модулям однієї й тієї самої програми.

Приклад 1.10. Скласти головну програму і підпрограму загального вигляду для обчислення зусиль в статично визначеній балці /рис. 1.14/.

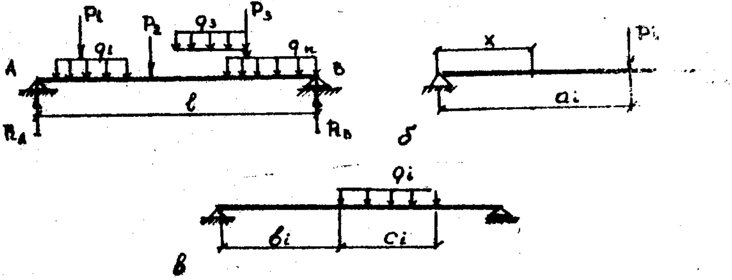


Рис. 1.14. До обчислення зусиль в статично визначеній балці: а – розрахункова схема; б – схема навантаження зосередженими силами; в – схема навантаження розподіленим вантажем

Скористаємося формулами:

$$M_x = R_A \cdot x - \begin{cases} 0, & x \le a_i \\ \Sigma P_i (x - a_i), & x > a_i \end{cases} - \begin{cases} 0, & x \le b_i \\ \Sigma q_i (x - b_i)^2/2, & b_i < x \le (b_i + c_i) \\ \Sigma q_i c_i \left(x - \frac{c_i}{2} - b_i\right), & x > (b_i + c_i) \end{cases} .$$

$$Q_x = R_A - \begin{cases} 0, & x \leq a_i \\ \sum P_i, & x > a_i \end{cases} - \begin{cases} 0, & x \leq b_i \\ \sum q_i(x-b_i), & b_i < x \leq (b_i+c_i) \\ \sum q_i c_i, & x > (b_i+c_i) \end{cases}$$

Головна програма

```
   ␣ REAL MBX,L,QBX,RA,X
   ␣ COMMON /Pi/ KP,P(10),A(10)
   ␣ COMMON/Qi/ KQ,Q(5),B(5),C(5)
   ␣ READ (5,1) L,KP,KQ
 1 ␣ FORMAT(F8.3,2I4)
   ␣ READ(5,8) P,A,Q,B,C
 8 ␣ FORMAT (10F8.2/10F8.3/5F8.3/5F8.3)
   ␣ WRITE (6,11)L,KP,KQ
11 ␣ FORMAT(1X,'L=',F8.3,' KP=',I4,' KQ=',I4/1X,50('-'))
   ␣ WRITE (6,12) P
12 ␣ FORMAT (1X,'МАССИВ P:'/1X,10F8.2 /1X,5F8.3/1X,50('-')/
   *␣␣1X,5F8.3 /1X,50('-')/1X,5F8.3/1X,50('-'))
   ␣ CALL BALKA (MBX,QBX,0.,L,L)
   ␣ RA=-MBX/L
   ␣ RB= -QBX-RA
   ␣ KS = 13
C  ␣ вычисление усилий в балке
   ␣ DO 3 I=1,KS
   ␣ X =L*(I-1.)/(KS-1.)
   ␣ CALL BALKA(MBX,QBX,RA,L,X)
   ␣ WRITE (6,15) X,MBX,QBX
15 ␣ FORMAT(3X,'X=',F8.3,3X,'MBX=',F10.3,3X,'QBX=',F10.3)
 5 ␣ CONTINUE
   ␣ STOP
   ␣ END
```

Підпрограма

```
   ␣ SUBROUTINE BALKA (MBX,QBX,RA,L,X)
   ␣ REAL MBX,L
   ␣ COMMON /Pi/ KP,P(10),A(10)              / KP – к-сть сил P /
   ␣ COMMON /Qi/ KQ,Q(5),B(5),C(5)            / KQ – к-сть сил q /
   ␣ MBX = RA*X
```

```
  ␣ QBX = RA
  ␣ DO 4 I=1,KP
  ␣ IF (X-A(I)) 4,4,2
2 ␣ MBX = MBX-P(I)*(X-A(I))
  ␣ QBX = QBX-P(I)
4 ␣ CONTINUE
  ␣ DO 8 I=1,KQ
  ␣ IF (B(I)-X)  5,6,6
5 ␣ IF  (X-(B(I)+C(I)))  7,7,8
7 ␣ MBX =MBX-Q(I)*(X-B(I))**2/2.
  ␣ QBX =QBX-Q(I)*(X-B(I))
  ␣ GOTO  8
8 ␣ MBX =MBX-Q(I)*C(I)*(X-B(I)-C(I)/2.)
  ␣ QBX =QBX-Q(I)*C(I)
6 ␣ CONTINUE
  ␣ RETURN
```

Цикл за зосередженими силами

Цикл за розподіленим вантажем

Підпрограма задання початкових значень /ініціалізація/

*BLOCK-DATA* – програмний модуль для надання початкових значень змінним і масивам, які входять в іменовані загальні області *COMMON*.

Підпрограма має вигляд:

```
BLOCK-DATA  – оператор-заголовок підпрограми
  COMMON з іменем                                     /I.21/
         <Оператори описування типу змінних >
  DIMENSION
  DATA
  END
```

Наприклад:

```
BLOCK-DATA
REAL A(5,2), B, C(2)
INTEGER K1(3,3), J, M
COMMON /MIR/ K1, B, C
COMMON /VEF/ A, J, M
DATA J/50/, A/10*0.0/, K1/1,4*2/
END
```

## I.9. Етапи підготовки Фортран-програми до виконання на ЕОМ

Для реалізації написаної програми на ЕОМ потрібно перекласти цю програму на мову даної ЕОМ. В ОС ЄС ЕОМ цей переклад складається з двох етапів: трансляція і редагування.

Програма, яку називають транслятором, перекладає кожний модуль Фортран-програми на мову ЕОМ, формує об"єктний модуль /див.мал.I.3/.

Вихідні модулі Фортран-програми звичайно вводяться з перфокарток, а об"єктні модулі розміщуються на магнітних дисках або стрічках як набори даних послідовного доступу.

Програма, яку звуть редактором зв"язку, виконує такі операції: I/ поєднує об"єктні модулі програми в єдину програму, яку називають завантажувальним модулем; 2/ додає до сформованого завантажувального модуля ряд модулів із системної бібліотеки Фортрану для виконання математичних функцій /вмонтованих функцій, таких, як $\sin, \cos, e^x, \ln x$, $\lg x, \mathrm{tg}\, x$ тощо/ і оператори введення/виведення.

Вихідні об"єктні модулі редактор зв"язку бере з наборів, в які їх розмістив транслятор, а завантажувальні модулі розміщуються у власній або тимчасовій бібліотеці програм, організованій на магнітних дисках.

### Бібліотеки програм

В ЕОМ, в яких є ОС ЄС ЕОМ, програми, виконувані на ній, знаходяться в бібліотеці програм у вигляді завантажувальних модулів. Бібліотеки програм розміщуються в наборах даних з бібліотечною організацією.

В ОС ЄС ЕОМ передбачено три типи бібліотек завантажувальних модулів: загальну /системну/, власну та тимчасову.

Загальна бібліотека /SYS1.LINKLIB/ включає такі програми: транслятор, редактор зв"язку та інші обслуговуючі програми. Вона постачається разом з ОС.

Власні бібліотеки створюються користувачем, і в них зберігаються розроблені ним програми.

Тимчасові бібліотеки створюються і використовуються в процесі виконання одного завдання. Після виконання завдання ОС ЕС ЕОМ ліквідує їх без додаткових вказівок.

## Загальні відомості про мови управління завданням

Мова управління завданням /МУЗ/ включає в себе 9 основних управляючих операторів:

- JOB – оператор завдання;
- EXEC – оператор виконання, або оператор пуску завдання;
- DD – оператор описування даних;
- // – порожній оператор.

Оператори МУЗ записуються в такому форматі:

// поле імені ␣ поле операції ␣ поле параметрів ␣ поле коментарів.

Символи // ідентифікують початок управляючого оператора, записуються в перших двох позиціях стрічки бланка, екрана дисплею.

Для описування часто виконуваних наборів дій, необхідних користувачеві, у бібліотеці ОС знаходяться готові послідовності управляючих операторів. Їх називають каталогізованими процедурами. Вони зберігаються в системній бібліотеці процедур SYS1.PROCLIB. Для звертання до цих процедур в завданні записується оператор EXEC, у полі параметрів якого указується ім"я каталогізованої процедури.

Наведемо приклад завдання на компіляцію, редагування і виконання програми мовою Фортран.

```
   //<ИМЯ> JOB␣MSGLEVEL=(1,1)
   //ST1␣EXEC␣FORTGCLG                 Початок кроку завдання
   //FORT.SYSPRIN␣DD␣SYSOUT=P          Трансляція завдання
   //FORT.SYSIN␣DD␣*              ┐
            <ТЕКСТ ПРОГРАММЫ>     }  Введення тексту програми
                                  ┘
   /*
  {//LKED.SYSLIB␣DD␣DSN=SYS1.FORTLIB,DISP=SHR
★ {//␣DD␣DSN=SYS1.SSPLIB,UNIT=SYSDA,DISP=SHR,
  {//␣VOL=SER=VVVVVV
   //LKED.SYSPRINT␣DD␣SYSOUT=P         Редагування завдання
   //GO.FT05F001␣DD␣*             ┐
            <ИСХОДНЫЕ ДАННЫЕ>     }  Введення вихідних даних
   /*                             ┘
   //GO.FT06F001␣DD␣SYSOUT=P           Виконання завдання
   //
```

Примітка. Картки /★/ присутні, коли підключається бібліотека стандартних підпрограм /ПНП/.

# 2. ЧИСЕЛЬНІ МЕТОДИ

## 2.1. Чисельні методи лінійної алгебри

Розвиток електронної обчислювальної техніки, створення алгоритмічних мов програмування та широкого математичного забезпечення ЕОМ дає змогу широко використовувати методи обчислювальної математики для розв"язування різного роду прикладних задач, зокрема задач будівництва.

Інженеру-будівельнику часто доводиться розв"язувати алгебраїчні, трансцендентні, диференціальні рівняння /як лінійні, так і нелінійні/, різні задачі оптимізації, а також обробляти масиви числових даних.

Вміння кваліфіковано використовувати весь арсенал чисельних методів для реалізації на ЕОМ тієї чи іншої задачі стало важливою вимогою підготовки інженера-будівельника.

Чисельними, як уже зазначалося, називаються такі методи розв"язування задач, які зводяться до арифметичних і логічних дій з числами, тобто дій, які може виконати ЕОМ.

У курсі чисельних методів широко використовується основи матричного числення. Зокрема, поняття вектора, матриці, оберненої матриці, власного значення і власного вектора.

### Матриці та дії з ними. Обчислення визначників матриці 2-го та 3-го порядків методом Крамера. Знаходження визначника методом Гаусса

Інженерне програмування можна назвати матричним.

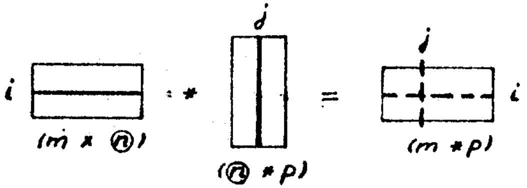
Матриця - це система з $n \times m$ чисел, упорядкованих у формі прямокутної таблиці, що має $n$ рядків та $m$ стовпців. Записують:

$$\underset{(n \times m)}{A} = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{vmatrix}.$$

Додавати або віднімати можна тільки матриці одного й того самого порядку. Результатом є матриця такого ж порядку. Якщо $c_{ik} = a_{ik} \pm b_{ik}$, то A ± B = C. Наприклад:

$$\begin{vmatrix} a & c \\ b & d \end{vmatrix} + \begin{vmatrix} f & q \\ e & h \end{vmatrix} = \begin{vmatrix} (a+f) & (c+q) \\ (b+e) & (d+h) \end{vmatrix}.$$

Дві матриці можуть бути перемножені, якщо число стовпців однієї з них дорівнює числу рядків другої. Схема обчислення добутків двох матриць має такий вигляд:



$$c_{ik} = \sum_{j=1}^{j=p} a_{ij} b_{jk}$$

$$\begin{vmatrix} 3 & 0 & 5 \\ 4 & 2 & 1 \end{vmatrix} * \begin{vmatrix} 5 & 4 \\ 0 & 4 \\ 4 & 5 \end{vmatrix} = \begin{vmatrix} [(3\cdot5)+(0\cdot0)+(5\cdot4)][(3\cdot4)+(0\cdot4)+(5\cdot5)] \\ [(4\cdot5)+(2\cdot0)+(1\cdot4)][(4\cdot4)+(2\cdot4)+(1\cdot5)] \end{vmatrix} = \begin{vmatrix} 35 & 37 \\ 24 & 29 \end{vmatrix}.$$

Поняття визначника виникло в зв"язку із знаходженням формул для значень невідомих в системі лінійних рівнянь.

Нехай маємо систему лінійних рівнянь 2-го порядку:

$$\begin{cases} a_1 x + b_1 y = c_1 \\ a_2 x + b_2 y = c_2 . \end{cases} \qquad /2.1/$$

Щоб розв"язати її, помножимо перше рівняння на $b_2 (a_2)$, друге на $b_1 (a_1)$ і додамо ліву й праву частини. Після перетворень дістанемо:

$$x = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1}; \qquad y = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1}. \qquad /2.2/$$

Припустивши, що $a_1 b_2 - a_2 b_1 \neq 0$, розглянемо таблицю чисел

$$\left( \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \right) \qquad /2.3/$$

Це квадратна матриця $A$ другого порядку. Вираз $a_1 b_2 - b_1 a_2$ називається визначником матриці $A$ :

$$\det A = |A| = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1 . \qquad /2.4/$$

Обчислюється він за схемою:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22}-a_{12}a_{21}.$$

Запишемо чисельники формул /2.2/ так:

$$c_1b_2-c_2b_1=\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}=|A_1|;\quad a_1c_2-a_2c_1=\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}=|A_2|.$$

Тоді /2.2/ можна подати у вигляді

$$x=\frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}=\frac{|A_1|}{|A|};\quad y=\frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}=\frac{|A_2|}{|A|}. \qquad /2.5/$$

Формули /2.5/ можна застосувати, якщо $\det A = |A| \neq 0$. Коли визначник матриці дорівнює нулю, матриця називається особливою.

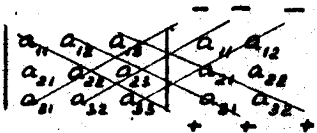
Розглянемо систему лінійних рівнянь 3-го порядку:

$$\begin{cases} a_1x+b_1y+c_1z=d_1 \\ a_2x+b_2y+c_2z=d_2 \\ a_3x+b_3y+c_3z=d_3. \end{cases} \qquad /2.6/$$

Відповідна квадратна матриця

$$A=\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}. \qquad /2.7/$$

Нехай $\det A = |A|$. Обчислюється визначник третього порядку за правилом Саррюса:



$$=a_{11}a_{22}a_{33}+a_{12}a_{23}a_{31}+a_{13}a_{21}a_{32}-$$
$$-a_{31}a_{22}a_{13}-a_{11}a_{23}a_{32}-a_{12}a_{21}a_{33}.$$

Корені системи /2.6/ згідно з формулами Крамера

$$x=\frac{|A_1|}{|A|},\quad y=\frac{|A_2|}{|A|},\quad z=\frac{|A_3|}{|A|}, \qquad /2.8/$$

Матриці $A_1, A_2, A_3$ у формулах /2.8/, а також у формулах /2.5/ дістають з матриці $A$ заміною відповідного стовпця на вільні члени.

Приклад 2.1. Розв"язати систему

$$\begin{cases} 2x - 3y + 5z = 1 \\ x - 2y - 3z = 7 \\ 3x - y - 2z = 5; \end{cases}$$

$$|A| = \begin{vmatrix} 2 & -3 & 5 \\ 1 & -2 & -3 \\ 3 & -1 & -2 \end{vmatrix} = -28; \qquad |A_1| = \begin{vmatrix} 1 & -3 & 5 \\ 7 & 2 & -3 \\ 5 & -1 & -2 \end{vmatrix} = -89;$$

$$|A_2| = \begin{vmatrix} 2 & 1 & 5 \\ 1 & 7 & -3 \\ 3 & 5 & -2 \end{vmatrix} = -85; \qquad |A_3| = \begin{vmatrix} 2 & -3 & 1 \\ 1 & 2 & 7 \\ 3 & -1 & 5 \end{vmatrix} = -21;$$

$$x = \frac{89}{28}; \quad y = \frac{85}{28}; \quad z = \frac{3}{4}.$$

### Визначник $n$-го порядку

Нехай маємо квадратну матрицю $n$-го порядку

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{vmatrix}, \qquad /2.9/$$

Тоді

$$\det A = |A| = a_{i1}A_{i1} + a_{i2}A_{i2} + \dots + a_{in}A_{in} = \sum_{i=1}^{n} a_{ik}A_{ik} \qquad /2.10/$$

або

$$\det A = |A| = a_{1k}A_{1k} + a_{2k}A_{2k} + \dots + a_{nk}A_{nk} = \sum_{i=1}^{n} a_{ki}A_{ki}. \qquad /2.10a/$$

де $A_{in}, A_{nk}$ – алгебраїчні доповнення матриці.

Алгебраїчним доповненням елемента квадратної матриці $n$-го порядку $a_{ij}$ називається визначник $(n-1)$-го порядку, який утворюється з даного визначника закресленням $i$-го рядка і $j$-го стовпця і множенням на $(-1)^{(i+j)}$.

Отже, визначник матриці $A$ на відміну від самої матриці є скалярна величина, значення якої можна обчислити, розклавши цей визначник за елементами $i$-го рядка чи $k$-го стовпця.

Враховуючи сказане, можна записати теорему розподілу: якщо $\det A = |a_{ik}|$ – визначник $n$-го порядку, то

$$\det A = \sum_{i=1}^{n} a_{ik} A_{ik} = \sum_{i=1}^{n} a_{ki} A_{ki}, \quad 1 \leq k \leq n, \qquad /2.11/$$

а саме: сума добутків всіх елементів довільного рядка /стовпця/ матриці на відповідні їм алгебраїчні доповнення дорівнює її визначнику.

Приклад 2.2. Обчислити визначник матриці, розклавши його по елементам третього стовпця.

$$\Delta = \begin{vmatrix} -x & 1 & 0 & -1 \\ -1 & -x & 0 & 1 \\ 1 & -1 & 0 & -x \\ 1 & 2 & 3 & 4 \end{vmatrix}; \quad \Delta = 0 \cdot A_{13} + 0 \cdot A_{23} + 0 \cdot A_{33} + 3 \cdot A_{43} =$$

$$= -3\Delta_{43} = -3 \begin{vmatrix} -x & 1 & -1 \\ -1 & -x & 1 \\ 1 & -1 & -x \end{vmatrix} = 3x^3 + 9x.$$

## Нормований визначник матриці

Нормований визначник матриці $A$ подається рівністю

$$\widetilde{\det} A = \mu_1 \mu_2 \ldots \mu_n \det A, \qquad /2.12/$$

де

$$\mu_i = \left( \sum_{j=1}^{n} a_{ij}^2 \right)^{-\frac{1}{2}}.$$

Приклад 2.3.

$$A = \begin{pmatrix} 5 & 4 & -2 \\ 3 & 6 & 5 \\ 1 & -3 & 3 \end{pmatrix}; \quad \det A = 179$$

$$\mu_1 = [5^2 + 4^2 + 2^2]^{-\frac{1}{2}} = \frac{1}{6,7}; \quad \mu_2 = [3^2 + 6^2 + 5^2]^{-\frac{1}{2}} = \frac{1}{8,35};$$

$$\mu_3 = [1^2 + 3^2 + 3^2]^{-\frac{1}{2}} = \frac{1}{4,35};$$

$$\widetilde{\det} A = \frac{179}{6,7\cdot 8,35\cdot 4,35} = 0,738;$$

$$-1 \le \widetilde{\det} A \le 1. \qquad /2.I3/$$

Зауважимо, що розв"язувати системи рівнянь, безпосередньо застосовуючи теорію визначників, можна лише тоді, коли число невідомих невелике. При збільшенні числа невідомих доводиться звертатися до різних спеціальних методів і прийомів. Зокрема, для випадку симетричної $(a_{ik} = a_{ki})$ матриці зручний метод Гаусса.

### Застосування методу Гаусса для обчислення визначника

Нехай дано матрицю

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}. \qquad /а/$$

Розглянемо лінійну систему

$$Ax = 0. \qquad /б/$$

При розв"язуванні цієї системи методом Гаусса матриця $A$ заміняється трикутною матрицею $B$ з одиничними коефіцієнтами по головній діагоналі:

$$B = \begin{vmatrix} 1 & b_{12} & b_{13} & \dots & b_{1n} \\ 0 & 1 & b_{23}^{(1)} & \dots & b_{2n}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix}. \qquad /2.I4/$$

У результаті маємо систему, еквівалентну системі /б/:

$$Bx = 0. \qquad /в/$$

Елементи матриці $B$ послідовно дістали з елементів матриці $A$ та допоміжних матриць $A_1, A_2, \dots, A_{n-1}$ за допомогою таких елементарних перетворень:

I/ ділення на головні елементи $a_{11}, a_{22}^{(1)}, \dots, a_{nn}^{(n-1)}$, які відмінні від нуля;

2/ віднімання від рядків матриці $A$ та допоміжних матриць $A_i$ $/ i = 1, 2, \dots, n-1 /$ чисел, пропорційних до елементів відповідних головних рядків.

Під час першої операції визначник матриці також ділиться на відповідний головний елемент, під час другої – залишається незмінним. Тому

$$\det B = 1 = \frac{\det A}{a_{11}\, a_{22}^{(1)} \ldots a_{nn}^{(n-1)}} . \qquad /2.I5/$$

Отже,

$$\det A = a_{11}\, a_{22}^{(1)} \ldots a_{nn}^{(n-1)} . \qquad /2.I6/$$

тобто визначник дорівнює добутку головних елементів для відповідної схеми Гаусса.

Приклад 2.4. Обчислити визначник матриці:

$$\Delta = \begin{vmatrix} 7,4 & 2,2 & -3,1 & 0,7 \\ 1,6 & 4,8 & -8,5 & 4,5 \\ 4,7 & 7,0 & -6,0 & 6,6 \\ 5,9 & 2,7 & 4,9 & -5,3 \end{vmatrix} .$$

| I | П | Ш | ІУ | Примітка |
|---|---|---|---|---|
| 7,4 | 2,2 | -3,I | 0,7 | A |
| I,6 | 4,8 | -8,5 | 4,5 | |
| 4,7 | 7,0 | -6,0 | 6,6 | |
| 5,9 | 2,7 | 4,9 | -5,3 | |
| I | 0,2973 | -0,4I89 | 0,0946 | |
| | 4,324 | -7,830 | 4,349 | $A_I$ |
| | 5,603 | -4,03I | 6,I55 | |
| | 0,946 | 7,37I | -5,858 | |
| | I | -I,8II | I,006 | |
| | | 6,II3 | 0,52I | $A_2$ |
| | | 9,084 | -6,809 | |
| | | I | 0,085 | |
| | | | -7,584 | $A_3$ |

Визначник дорівнює добутку головних елементів:

$$\Delta = 7,4 \cdot 4,324 \cdot 6,113 \cdot (-7,584) = -1483,619 .$$

# Розв"язування системи лінійних рівнянь методом Гаусса, ітераційним методом, методом Зейделя

Система лінійних рівнянь - основний "інструмент" розв"язування задач науки, техніки, промисловості. Зокрема, у будівельній справі численні фізичні задачі зводяться до системи лінійних алгебраїчних рівнянь /розрахунок балансу сил, що діють на будь-яку конструкцію, - міст, багато поверховий будинок, каркас літака тощо/. Скажімо, системи канонічних рівнянь методу сил, методу деформацій /класичних методів розрахунку будівельних конструкцій/ є системами саме таких рівнянь.

Розглянемо деякі методи їх розв"язування.

У загальному вигляді система лінійних рівнянь записується так:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = c_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = c_2 \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = c_n . \end{cases} \qquad \text{/2.I7/}$$

Необхідною і достатньою умовою того, щоб ця система мала єдиний розв"язок, є нерівність нулю визначника даної системи. Усі відомі методи розв"язування систем виду /2.I7/ поділяються на прямі та ітераційні. До прямих методів належать і розглянутий раніше метод Крамера /метод визначників/,і метод Гаусса, який подається далі, до ітераційних - метод ітерацій, метод Зейделя.

## Метод виключення Гаусса

Найбільш поширені прямі методи грунтуються на зведенні системи рівнянь до трикутного виду, коли останнє рівняння включає лише одну змінну, а в кожному наступному з"являється ще по одній. Розглядуваний метод Гаусса також грунтується на цьому зведенні.

Спочатку нормують перше рівняння, ділячи його коефіцієнти на $a_{11}$. Далі утворене рівняння множать по черзі на перші коефіцієнти $a_{i1}$ всіх інших рівнянь і послідовно віднімають від решти рівнянь. У результаті перша змінна буде виключена з усіх рівнянь, крім першого. На наступному етапі розв"язування така сама процедура застосовується до решти $n$ -I рівнянь, з яких виключається друга змінна. Процедура повторюється доти, поки після $n$ кроків система не буде зведена до трикутного виду.

Математично цю процедуру можна описати так: на $k$-му кроці процесу виключення нормовані коефіцієнти $k$-го рівняння мають вигляд

$$b_{kj}=\frac{a_{kj}}{a_{kk}},$$

а нові коефіцієнти в наступних рівняннях

$$b_{ij}=a_{ij}-a_{ik}b_{kj}, \quad i>k.$$

Головні коефіцієнти рівнянь $(a_{11}, a_{22}^{(1)}, \dots, a_{nn}^{(n-1)})$ не дорівнюють нулю. Цієї умови завжди можна додержати, виконавши відповідні перетворення.

Схема зведення системи рівнянь до трикутного вигляду називається схемою єдиного ділення. Процес знаходження коефіцієнтів $b_{ij}$ трикутної системи називається прямим ходом. З останнього рівняння трикутної системи, яке містить одну змінну, знаходимо її значення, а далі зворотним ходом обчислюємо значення решти змінних.

Отже, алгоритм Гаусса включає в себе два етапи:

1/ побудова допоміжної системи з трикутною матрицею /прямий хід/,

2/ знаходження розв"язків побудованої системи /зворотний хід/.

Розглянутий метод дає змогу розв"язувати і так звані погано обумовлені системи лінійних алгебраїчних рівнянь, які дістаємо при розрахунках будівельних конструкцій, коли невдало вибрані основні системи. Головні елементи системи в цьому разі інтерпретують безпосередньо твердість, або податливість, будівельних конструкцій, а тому відмінні від нуля.

Зауважимо, що оскільки йдеться про матриці твердості, то головна мета при виборі основної системи - отримати на головній діагоналі такі елементи, які були б значно більші за решту. При цьому, якщо найбільший елемент в кожному рядку матриці взяти за головний, то можна відразу зменшити можливість поганої обумовленості.

Розробка алгоритмів розв"язування задач і створення відповідних програм обчислень є насамперед справою спеціалістів з чисельних методів, а не з будівельної механіки. Але оскільки при розв"язуванні задач будівельної механіки на ЕОМ значна частина машинного часу витрачається на пошук розв"язків систем алгебраїчних рівнянь і обернення матриць, бажано, щоб інженер мав уявлення про те, за якими схемами реалізуються відповідні програми.

**Приклад 2.5.** Розв"язати методом Гаусса систему рівнянь:

$$\begin{aligned} x_1 + x_2 + x_3 - x_4 &= 2 \\ x_1 - x_2 - x_3 + x_4 &= 0 \\ 2x_1 + x_2 - x_3 + 2x_4 &= 9 \\ 3x_1 + x_2 + 2x_3 - x_4 &= 7 \end{aligned}$$

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | Вільні члени | |
|---|---|---|---|---|---|---|
| Прямий хід | $\boxed{\text{I}}$ | I | I | –I | 2 | A (4×4) |
| | I | –I | –I | I | 0 | |
| | 2 | I | –I | 2 | 9 | |
| | 3 | I | 2 | –I | 7 | |
| | I | I | I | –I | 2 | |
| | 0 | $\boxed{+2}$ | +2 | –2 | 2 | B (3×3) |
| | 0 | I | 3 | –4 | –5 | |
| | 0 | 2 | I | –2 | 2 | |
| | | I | I | –I | I | |
| | | | $\boxed{-2}$ | 3 | +6 | C (2×2) |
| | | | I | 0 | 3 | |
| | | | I | $-3/2$ | –3 | |
| | | | | $\boxed{-3/2}$ | –6 | D (1×1) |
| Зворотний хід | | | | I | 4 | Корені системи рівнянь |
| | | | I | | 3 | |
| | | I | | | 2 | |
| | I | | | | I | |

$$x_3\cdot 1 - \frac{3}{2}\cdot 4 = -3; \quad x_3 = -3 + 6 = 3;$$

$$x_2\cdot 1 + 1\cdot 3 - 1\cdot 4 = 1; \quad x_2 = 1 + 4 - 3 = 2;$$

$$1\cdot x_1 + 1\cdot 2 + 1\cdot 3 - 1\cdot 4 = 2; \quad x_1 = 2 + 4 - 3 - 2 = 1.$$

Виконавши прямий хід, дістанемо трикутну матрицю

$$\left|\begin{array}{cccc|c} 1 & 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & -1 & 1 \\ 0 & 0 & 1 & -1,5 & -3 \\ 0 & 0 & 0 & 1 & 4 \end{array}\right| .$$

Зворотним ходом знаходимо корені:

$$x_1 = 1; \; x_2 = 2; \; x_3 = 3; \; x_4 = 4.$$

## Ітераційні методи розв"язування системи лінійних рівнянь

Ітераційні схеми розв"язування систем лінійних алгебраїчних рівнянь застосовуються до систем, зведених до вигляду

$$\begin{cases} x_1 = \frac{1}{a_{11}}(-a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n + b_1) \\ x_2 = \frac{1}{a_{22}}(-a_{21}x_1 - a_{13}x_3 - \ldots - a_{2n}x_n + b_2) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \frac{1}{a_{nn}}(-a_{n1}x_1 - \ldots - a_{n,n-1}x_{n-1} + b_n) \end{cases} \qquad /2.18/$$

/перше рівняння розв"язане відносно $x_1$, друге відносно $x_2$ і т.д./

Праві частини рівнянь системи /2.18/ можна розглядати як функції від $n$ аргументів $x_1, x_2, \ldots, x_n$. Позначимо праву частину $i$-го рівняння через $L_i(x_1, x_2, \ldots, x_n)$/зберігаючи єдиний підхід, не зважаємо на те, що в правій частині $i$-го рівняння $x_i$ відсутнє/. Тоді система набере вигляду:

$$\begin{cases} x_1 = L_1(x_1, x_2, \ldots, x_n) \\ x_2 = L_2(x_1, x_2, \ldots, x_n) \\ \ldots\ldots\ldots\ldots\ldots \\ x_n = L_n(x_1, x_2, \ldots, x_n). \end{cases} \qquad /2.19/$$

Задамо початкові /нульові/ наближення коренів цієї системи: $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$. Тоді перші наближення дістанемо, підставивши в праву її частину початкові наближення:

$$\begin{cases} x_1^{(1)} = L_1(x_1^{(0)}, \ldots, x_n^{(0)}) \\ x_2^{(1)} = L_2(x_1^{(0)}, \ldots, x_n^{(0)}) \\ \ldots\ldots\ldots\ldots\ldots \\ x_n^{(1)} = L_n(x_1^{(0)}, \ldots, x_n^{(0)}). \end{cases} \qquad /2.20/$$

Знайдені перші наближення можна використати для знаходження других і т.д. Ітераційний процес продовжується доти, поки всі $x^{(k)}$ не стануть достатньо близькі до $x^{(k-1)}$, тобто почне виконуватися нерівність

$$M^k = \max(x_i^k - x_i^{k-1}) \leq \varepsilon, \qquad /2.21/$$

де $i = 1, 2, \ldots, n$; $\varepsilon$ – задана точність.

Краще порівнювати з $\varepsilon$ не абсолютні, а відносні різниці сусідніх величин, розглядаючи нерівність

$$\max \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}} \right| < \varepsilon. \qquad /2.22/$$

Розв"язуючи системи лінійних рівнянь ітераційними методами, потрібно розглядати питання про збіжність обчислюваних наближень.

Для виконання умови збіжності ітераційного процесу необхідно, щоб сума модулів відношень коефіцієнтів будь-якого рядка до діагонального коефіцієнта була строго менша від одиниці:

$$\sum_{\substack{j=1\\ j\neq i}}^{n} \left| \frac{a_{ij}}{a_{ii}} \right| < 1. \qquad /2.23/$$

Нерівність /2.23/ виконуватиметься, якщо діагональні елементи системи /2.18/ задовольняють умову:

$$|a_{ii}| > \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}|. \qquad /2.24/$$

<u>Приклад 2.6</u>. Дано систему лінійних рівнянь

$$\begin{cases} 18x_1 + 2x_2 - 4x_3 = 71 \\ 8x_1 - 19x_2 + x_3 = 8 \\ 9x_1 - 3x_2 + 17x_3 = 2. \end{cases}$$

Оцінити збіжність методу ітерацій для даної системи і звести її до виду /2.19/.

Перевіримо виконання умови збіжності:

$$\begin{aligned} &|18| > |2| + |-4| = 6; \\ &|-19| > |8| + |1| = 9; \\ &|17| > |9| + |-3| = 12. \end{aligned}$$

Таким чином, ця умова виконується.

Тепер зводимо дану систему до потрібного виду:

$$\begin{cases} x_1 = \frac{1}{18}(71 - 2x_2 + 4x_3) \\ x_2 = -\frac{1}{19}(8 - 8x_1 - x_3) \\ x_3 = \frac{1}{17}(2 - 9x_1 + 3x_2). \end{cases}$$

## Метод Зейделя

Звернемося знову до системи рівнянь /2.19/.

$$\begin{cases} x_1 = L_1(x_1, x_2, \ldots, x_n) \\ x_2 = L_2(x_1, x_2, \ldots, x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = L_n(x_1, x_2, \ldots, x_n). \end{cases}$$

Щоб обчислити $(k-1)$-ше наближення $x_i$ за методом ітерацій, потрібно використати $x_i^{(k)}$ де $i = 1,2,\ldots n$. Проте до моменту обчислення $x_i^{(k+1)}$ уже знайдено $x_1^{(k+1)},\ldots,x_{i-1}^{(k+1)}$ тому напрошується висновок: при обчисленні $x_i^{(k+1)}$ доцільно використовувати знайдені раніше $x_1^{(k+1)}, x_2^{(k+1)},\ldots,x_{i-1}^{(k+1)},\ldots,x_{i+1}^{(k)},\ldots,x_n^{(k)}$.

У цьому й полягає метод Зейделя: для обчислення $(k+1)$-го наближення $x_i$ використовують не лише $k$-те, але й $(k+1)$-ше наближення, тобто:

$$\begin{aligned} &x^{(k+1)} = L_1(x_1^{(k)},\ldots,x_n^{(k)}); \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &x_i^{(k+1)} = L_i(x_1^{(k+1)},\ldots,x_{i-1}^{(k+1)}, x_i^{(k)}); \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &x_n^{(k+1)} = L_n(x_1^{(k+1)},\ldots,x_{n-1}^{(k+1)}, x_n^{(k)}). \end{aligned} \qquad /2.25/$$

Можна вважати, що умови збіжності, отримані для методу ітерацій, достатні для методу Зейделя, який називають ще поліпшеним методом ітерацій.

Зауважимо, що за допомогою методу Гаусса можна розв"язати будь-яку систему лінійних алгебраїчних рівнянь. Що ж до системи рівнянь, розв"язуваної ітераційними методами, то для неї мають забезпечуватися умови збіжності. Проте час, потрібний для розв"язування системи за методом Гаусса, пропорційний до $n^3$, а за методом ітерацій - до $n^2$, де $n$ - порядок системи. До того ж, похибки округлень, які з"являються при використанні методу Гаусса, можуть призвести до беззмістовного результату. Ітераційні методи дають змогу раціонально використати пам"ять ЕОМ, потрібну для записування коефіцієнтів системи, оскільки для знаходження чергового наближення $k$-го кореня використовуються коефіцієнти лише $k$-го рівняння. Останнє особливо важливе при розв"язуванні системи, яка не вміщується в пам"яті ЕОМ.

## Обернення матриці методом Гаусса

Матриці рівнянь рівноваги, систем канонічних рівнянь в будівельній механіці - неособливі за своїм фізичним походженням.

Випадок, коли визначник такої матриці дорівнює нулю, указує або на геометричну змінюваність системи, або на втрачену нею стійкість, або просто свідчить про допущену помилку. В будь-якому разі потрібна перевірка обчислень.

Для неособливої квадратної матриці $n \times n$ $(\det A \neq 0)$ вводиться поняття оберненої матриці.

Якщо А – неособлива квадратна матриця $n$-го порядку $(\det A \neq 0)$, то існує така обернена до неї матриця $A^{-1}$ $n$-го порядку, що

$$AA^{-1} = A^{-1}A = E,$$

де

$$E = \begin{vmatrix} 1 & & & 0 \\ & 1 & & \\ & & \ddots & \\ 0 & & & 1 \end{vmatrix}.$$

одинична діагональна матриця.

Нехай дано неособливу матрицю

$$A = [a_{ij}] \quad (i,j = 1,2,\dots,n).$$

Для знаходження оберненої до неї матриці

$$A^{-1} = [x_{ij}]$$

використаємо основне співвідношення

$$AA^{-1} = E. \qquad /2.26/$$

Кожний стовпець матриці $A^{-1}$ можна розглядати як сукупність невідомих коренів, а кожний стовпець матриці $E$ – як відповідний стовпець вільних коефіцієнтів. Тоді таку систему, яка має матрицю $A$ і різні вільні члени, можна безпосередньо розв"язати методом Гаусса.

<u>Приклад 2.7</u>. Знайти матрицю, обернену до матриці

$$A = \begin{pmatrix} 10 & 6 & 8 \\ 3 & 1 & 7 \\ 5 & 4 & 2 \end{pmatrix}.$$

Складаємо схему єдиного ділення. При цьому маємо три стовпці вільних членів

| $x_1$ | $x_2$ | $x_3$ | Вільні коефіцієнти | | |
|---|---|---|---|---|---|
| | | | I | II | III |
| I0<br>3<br>5 | 6<br>I<br>4 | 8<br>7<br>2 | I<br>0<br>0 | 0<br>I<br>0 | 0<br>0<br>I |
| I | 0,6 | 0,8 | 0,I | 0 | 0 |
| | 0,8<br>-I | -4,6<br>2 | 0,3<br>0,5 | -I<br>0 | 0<br>-I |
| | I | -5,75 | 0,375 | -I,25 | 0 |
| | | -3,75 | 0,875 | -I,25 | -I |
| | | I | -0,2333 | 0,3333 | 0,2667 |
| | I | | -0,9665<br>0,8665 | 0,6665<br>-0,6665 | I,5335<br>-I,I333 |
| I | | | | | |

Варіант I $1\cdot x_2-5,75(-0,2333)=0,375;\quad x_2^{I}=-0,9665;$
Варіант II $1\cdot x_2-5,75(0,3333)=-1,25;\quad x_2^{II}=0,6666;$
Варіант III $1\cdot x_2-5,75(0,2667)=0;\quad x_2^{III}=1,5335.$

Варіант I $1\cdot x_1+0,6(-0,9665)+0,8(-0,2333)=0,1;\quad x_1^{I}=0,8665;$
Варіант II $1\cdot x_1+0,6(0,6665)+0,8(0,3333)=0;\quad x_1^{II}=-0,6665;$
Варіант III $1\cdot x_1+0,6(1.5335)+0,8(0,2667)=0;\quad x_3^{III}=-1,1333.$

$$A^{-1}=\begin{vmatrix} 0,8665 & -0,6665 & -1,1333 \\ -0,9665 & 0,6665 & 1,5335 \\ -0,2333 & 0,3333 & 0,2667 \end{vmatrix}.$$

## Обернення матриці з використанням мінорів та визначників

Алгебраїчне доповнення елемента $a_{ij}$ квадратної матриці обчислюється за формулою

$$(-1)^{i+j}M_{ij}, \qquad /2.27/$$

де $M_{ij}$ – мінор елемента $a_{ij}$, отриманий з даного визначника матриці шляхом викреслення $i$-го рядка і $j$-го стовпця.

$$\underset{(n\times n)}{A^{-1}}=\frac{1}{\det A}\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & & A_{nn} \end{pmatrix}^{T}. \qquad /2.28/$$

Приклад 2.8.

Нехай $\underset{(3\times 3)}{A}=\begin{pmatrix} 10 & 6 & 8 \\ 3 & 1 & 7 \\ 3 & 4 & 2 \end{pmatrix}$. Знайти $A^{-1}$.

Розв"язання. Обчислюємо $\det A=-30$, а також мінори матриці $A$:

$M_{11}=1\cdot2-4\cdot7=-26;\quad M_{32}=10\cdot7-3\cdot8=46;$
$M_{21}=6\cdot2-4\cdot8=-20;\quad M_{13}=3\cdot4-5\cdot1=7;$
$M_{31}=6\cdot7-1\cdot8=34;\quad M_{23}=10\cdot4-5\cdot6=10;$
$M_{12}=3\cdot2-5\cdot7=-29;\quad M_{33}=10\cdot1-3\cdot6=-8.$
$M_{22}=10\cdot2-5\cdot8=-20;$

Обчислюємо алгебраїчні доповнення за /2.27/

$$A_{11} = -26;\ A_{12} = 29;\ A_{13} = 7;$$
$$A_{21} = 20;\ A_{22} = -20;\ A_{23} = -10;$$
$$A_{31} = 34;\ A_{32} = -46;\ A_{33} = -8.$$

Матриця алгебраїчних доповнень має вигляд

$$A_{ij} = \begin{pmatrix} -26 & 29 & 7 \\ 20 & -20 & -10 \\ 34 & -46 & -8 \end{pmatrix}.$$

Транспонуємо її

$$A_{ij}^{T} = \begin{pmatrix} -26 & 20 & 34 \\ 29 & -20 & -10 \\ 7 & -10 & -8 \end{pmatrix}.$$

Обернену матрицю обчислюємо за /2.28/:

$$A^{-1} = \begin{vmatrix} \frac{-26}{-30} & \frac{20}{-30} & \frac{34}{-30} \\ \frac{29}{-30} & \frac{-20}{-30} & \frac{-46}{-30} \\ \frac{7}{-30} & \frac{-10}{-30} & \frac{-8}{-30} \end{vmatrix} = \begin{vmatrix} 0,8665 & -0,6665 & -1,1333 \\ -0,9665 & 0,6665 & 1,5335 \\ -0,2333 & 0,3333 & 0,2667 \end{vmatrix}.$$

## Обчислення власних значень і власних векторів матриці. Запис вікового рівняння. Чисельні методи розв"язування вікового рівняння

Цілий ряд інженерних задач зводиться до розгляду системи рівнянь, що має єдиний розв"язок лише в тому випадку, коли відоме значення певного спеціального параметра, який входить до нього. Цей параметр називається характеристичним, або власним, значенням системи.

Із задачами на власні значення інженер зустрічається в різних ситуаціях. Так, для тензорів напруг власні значення задають головні нормальні напруги, а власні вектори - напрями, пов"язані з цими значеннями.

При динамічному аналізі будівельних систем власні значення відповідають частотам власних коливань такої системи, а власні вектори характеризують форми цих коливань.

При розрахунках будівельних конструкцій на стійкість власні значення дають змогу знайти критичні навантаження, перевищення яких веде до втрати стійкості.

Нехай дано квадратну матрицю $A = [a_{ij}]$.

Ненульовий вектор називається власним вектором даної матриці, якщо в результаті відповідного лінійного перетворення цей вектор переходить в колінеарний йому, тобто якщо результуючий вектор відрізняється від вихідного лише скалярним множником.

Вектор $x \neq 0$ називають власним вектором матриці $A$, якщо

$$Ax = \lambda x, \qquad /2.29/$$

де $\lambda$ - власне значення.

Рівняння $Ax = \lambda x$ еквівалентне рівнянню

$$(A - \lambda E)x = 0. \qquad /2.30/$$

Рівняння /2.30/ являє собою однорідну систему лінійних рівнянь, записану в матричному вигляді. Тривіальний найпростіший розв"язок системи /2.30/ маємо тоді, коли вектор $\vec{x} = 0$. Але він задачу не задовольняє, оскільки не дає значень власних векторів. Система /2.30/ має розв"язок, коли

$$\det(A - \lambda E) = 0. \qquad /2.31/$$

Многочлен $\det(A - \lambda E)$ - це характеристичне /частотне/, або вікове, рівняння матриці $A$. У розгорнутому вигляді воно записується так:

$$\det \begin{vmatrix} a_{11-\lambda} & a_{12} \ldots & a_{1n} \\ a_{21} & a_{22-\lambda} \ldots & a_{2n} \\ \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} \ldots & a_{nn-\lambda} \end{vmatrix} = 0. \qquad /2.32/$$

Обчислити значення власних чисел матриці можна двома способами: 1/ знайти корені системи лінійних однорідних рівнянь /2.30/ методом ітерацій без попереднього розгортання визначника /2.31/; 2/ розгорнути вікове рівняння /2.32/ в поліном $n$-го степеня і знайти його корені.

Розв"язавши вікове рівняння першим способом, дістанемо всі значення власних чисел і відповідних їм власних векторів.

За другим способом після обчислення коренів полінома /власних значень матриці/ власні вектори знаходяться розв"язуванням системи однорідних рівнянь /2.30/.

Приклад 2.9. Знайти власні значення матриці $A = \begin{vmatrix} 3 & -2 \\ -4 & 1 \end{vmatrix}$, розгорнувши визначник /2.32/ в поліном.

Розв"язання.

$$\begin{vmatrix} 3-\lambda & -2 \\ -4 & 1-\lambda \end{vmatrix} = 0; \ \lambda^2-4\lambda-5=0; \lambda_1=5; \lambda_2=-1.$$

Знайдемо власний вектор для $\lambda_1 = 5$:

$$\begin{vmatrix} -2 & -2 \\ -4 & -4 \end{vmatrix} \begin{vmatrix} x_1 \\ x_2 \end{vmatrix} = 0; \begin{cases} -2x_1-2x_2=0; \\ -4x_1-4x_2=0; \end{cases} \ \vec{x}_{\lambda_1} = \begin{vmatrix} -1 \\ 1 \end{vmatrix}.$$

Аналогічно знаходимо власний вектор для $\lambda_2$ = -I:

$$\begin{vmatrix} 4 & -2 \\ -4 & 2 \end{vmatrix} \vec{x}_{\lambda_2} = 0; \quad \vec{x}_{\lambda_2} = \begin{vmatrix} 1 \\ 2 \end{vmatrix}.$$

Зауважимо, що ітераційні методи добре застосовні для знаходження максимального і мінімального власних значень. Різновид ітераційного методу – метод вичерпування – грунтується на принципі ортогональності власних векторів:

$$\sum_{i=1}^{n} x_i^{(j)} x_i^{(k)} = 0.$$

Приклад 2.I0. Для матриці $\underset{(3\times3)}{A} = \begin{vmatrix} 4 & 2 & 2 \\ 2 & 5 & 1 \\ 2 & 1 & 6 \end{vmatrix}$ знайти корені $\lambda_j$ характеристичного рівняння та власні вектори $\vec{x}^{(j)}$ / $j = \overline{1,2,3}$/.

Розв"язання. Складемо вікове рівняння для матриці $A$

$$\begin{vmatrix} (4-\lambda) & 2 & 2 \\ 2 & (5-\lambda) & 1 \\ 2 & 1 & (6-\lambda) \end{vmatrix} = 0,$$

звідки знайдемо власні значення $\lambda$.

Маємо систему для знаходження власних векторів $\vec{x}_j$:

$$\begin{vmatrix} (4-\lambda) & 2 & 2 \\ 2 & (5-\lambda) & 1 \\ 2 & 1 & (6-\lambda) \end{vmatrix} * \begin{vmatrix} x_1^{(j)} \\ x_2^{(j)} \\ x_3^{(j)} \end{vmatrix} = 0,$$

або, в розгорнутому вигляді,

$$\begin{cases} (4-\lambda)x_1^{(j)} + 2x_2^{(j)} + 2x_3^{(j)} = 0 \\ 2x_1^{(j)} + (5-\lambda)x_2^{(j)} + 1x_3^{(j)} = 0 \\ 2x_1^{(j)} + 1\cdot x_2^{(j)} + (6-\lambda)x_3^{(j)} = 0. \end{cases}$$

Зведемо цю систему до вигляду, зручного для розв"язування її методом ітерацій:

$$\begin{cases} \lambda_j x_1^{(j)} = 4x_1^{(j)} + 2x_2^{(j)} + 2x_3^{(j)} \\ \lambda_j x_2^{(j)} = 2x_1^{(j)} + 5x_2^{(j)} + x_3^{(j)} \\ \lambda_j x_3^{(j)} = 2x_1^{(j)} + x_2^{(j)} + 6x_3^{(j)} \end{cases} \quad (j = 1, 2, 3). \qquad /a/$$

Схема алгоритму реалізації поставленої задачі зображена на рис. 2.I. Поклавши $j$ = I та $x_3^{(1)}$ = I, запишемо систему /а/ відносно невідомих в лівій частині

$$\begin{cases} x_1^{(1)} = \frac{1}{\lambda_1}(4x_1^{(1)} + 2x_2^{(1)}) \\ x_2^{(1)} = \frac{1}{\lambda_1}(2x_1^{(1)} + 5x_2^{(1)} + 1) \\ \lambda_1 = 2x_1^{(1)} + x_2^{(1)} + 6 \end{cases} \qquad /б/$$

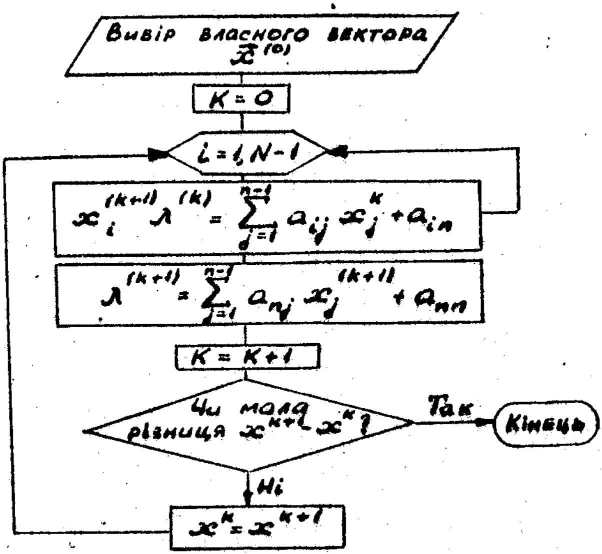


Рис. 2.I. Схема ітераційного методу знаходження найбільшого власного значення і відповідного власного вектора

Розв"яжемо систему /б/ методом ітерацій, взявши за початкові значення $x_1^{(1,0)} = I$ і $x_2^{(1,0)} = I$. Тоді з останнього рівняння цієї системи дістанемо $\lambda_1^{(0)} = 9$. Результати обчислень наведені в табл. 2.I.

Таблиця 2.I

До розв"язування системи /б/

| $k$ | $x_1^{(1k)}$ | $x_2^{(1k)}$ | $x_3^{(1k)}$ | $\lambda_1^{(k)}$ |
|---|---|---|---|---|
| 0 | I | I | I | 9 |
| I | 0,89 | 0,89 | I | 8,67 |
| 2 | 0,85 | 0,83 | I | 8,53 |
| . | . | . | . | . |
| II | 0,8077 | 0,7720 | I | 8,3874 |

Тепер маємо:

$$\left.\begin{aligned} x_1^{(1,1)} &= 1/9\,(4\cdot 1+2\cdot 1+2) = 0,89 \\ x_2^{(1,1)} &= 1/9\,(2\cdot 1+5\cdot 1+1) = 0,89 \\ \lambda_1^{(1)} &= 2\cdot 0,89+0,89+6 = 8,67 \end{aligned}\right\} \quad k=1,$$

$$\left.\begin{aligned} x_1^{(1,2)} &= 1/8,67\,(4\cdot 0,89+2\cdot 0,89+2) = 0,85 \\ x_2^{(1,2)} &= 1/8,67\,(2\cdot 0,89+5\cdot 0,89+1) = 0,83 \\ \lambda_1^{(2)} &= 2\cdot 0,85+1\cdot 0,83+6 = 8,53 \end{aligned}\right\} \quad k=2$$

і так далі, доти, поки не почне виконуватися нерівність $\left|\frac{z_i - z_{i-1}}{z_i}\right| < \varepsilon$, де $\varepsilon$ – наперед задана відносна величина різниці сусідніх значень.

З табл. 2.I беремо $\lambda_1 = 8,3874$ і $\vec{x}^{(1)} = \begin{vmatrix} 0,8077 \\ 0,7720 \\ 1 \end{vmatrix}$.

Підставимо в систему /а/ $j = 2$. З умови ортогональності векторів $x^{(1)}$ і $x^{(2)}$ дістанемо:

$$0,8077\,x_1^{(2)} + 0,7720\,x_2^{(2)} + x_3^{(2)} = 0,$$

звідки $x_3^{(2)} = -0,8077\,x_1^{(2)} - 0,7720\,x_2^{(2)}$.

Підставивши цей вираз в систему /а/ і взявши $x_2^{(2)} = 1$, отримаємо

$$\begin{cases} x_1^{(2)} = 1/\lambda_2\,(4\,x_1^{(2)} + 2\cdot 1 + 2\,(-0,8077\,x_1^{(2)} - 0,7720) \\ \lambda_2 = 2\cdot x_1^{(2)} + 5\cdot 1 - 0,8077\,x_1^{(2)} - 0,7720 \end{cases}$$

або

$$\begin{cases} x_1^{(2)} = 1/\lambda_2\,(2,3846\,x_1^{(2)} + 0,4560) \\ \lambda_2 = 1,1923\,x_1^{(2)} + 4,228. \end{cases}$$

Систему розв"язуємо методом ітерацій, покладаючи $x_1^{(20)} = I$ і $\lambda_2^{(0)} = 5,42$.

Результати обчислень записуємо в табл. 2.2.

Таблиця 2.2

До розв"язування системи /a/

| $k$ | $x_1^{(2k)}$ | $x_2^{(2k)}$ | $\lambda_2^{(k)}$ |
|---|---|---|---|
| 0 | I | I | 5,42 |
| I | 6,25 | I | 4,85 |
| . | . | . | . |
| II | 0,217 | I | 4,4867 |

Звідси: $\lambda_2 = 4,4867$ і $\vec{x}^{(2)} = \begin{vmatrix} 0,2170 \\ 1 \\ -0,9473 \end{vmatrix}$ .

Третій власний вектор $\vec{x}^{(3)}$ безпосередньо знаходимо з двох співвідношень ортогональності векторів

$$\begin{cases} \vec{x}^{(2)}\,\vec{x}^{(3)} = 0 \\ \vec{x}^{(1)}\,\vec{x}^{(3)} = 0 \end{cases}$$

або

$$\begin{cases} 0,8077\,x_1^{(3)} + 0,7720\,x_2^{(3)} + x_3^{(3)} = 0 \\ 0,2170\,x_1^{(3)} + x_2^{(3)} - 0,9473\,x_3^{(3)} = 0. \end{cases}$$

Поклавши $x_1^{(3)} = I$, дістанемо $x_2^{(3)} = -0,5673$, $x_3^{(3)} = -0,3698$, звідки

$$\vec{x}^{(3)} = \begin{vmatrix} 1 \\ -0,5673 \\ -0,3698 \end{vmatrix} .$$

Значення $\lambda_3$ знайдемо, підставивши $\vec{x}^{(3)}$ в перше рівняння системи /a/:

$$\lambda_3 = 2,1260.$$

Зауважимо, що корені вікового рівняння, які знаходяться методом ітерацій, як правило, розміщені в порядку зменшення їх модулів.

Власні вектори матриці знаходяться з точністю до коефіцієнтів пропорційності, а тому всі розв"язки вихідної системи такі:

| $\lambda_i$ | $x_1^{(j)}$ | $x_2^{(j)}$ | $x_3^{(j)}$ |
|---|---|---|---|
| 8,3874 | $0,8077c_1$ | $0,7720c_1$ | $c_1$ |
| 4,4867 | $0,2170c_2$ | $c_2$ | $-0,9473\,c_2$ |
| 2,1260 | $c_3$ | $-0,5673c_3$ | $-0,3698\,c_3$ |

де $c_1, c_2, c_3$ – вільні сталі, відмінні від нуля.

## 2.2. Чисельні методи розв"язування нелінійних алгебраїчних рівнянь

### Метод виділення коренів

Для знаходження коренів поліномів першого, другого чи третього степеня існують точні методи розв"язування рівнянь. В інших випадках звертаються до наближених методів.

Розв"язування рівнянь наближеними методами поділяється на два етапи: I/ визначення інтервала, де міститься корінь /знаходження початкового наближення/; 2/ уточнення кореня до заданої точності.

Часто наближене значення кореня відоме з фізичних міркувань. У противному разі його знаходять, аналізуючи властивості функції.

Щоб виділити на певному відрізку корені того чи іншого полінома, необхідно скористатися теоремою: якщо значення неперервної функції $F(x)$ на кінцях відрізка $[a, b]$ мають різні знаки, то в середині відрізка є принаймні один корінь рівняння $F(x) = 0$.

Початкове наближення кореня рівняння $F(x) = 0$ можна також знайти графічним способом. У прямокутній декартовій системі координат будується графік функції $y = F(x)$. Абсциси точок перетину його з віссю $x$ і дадуть шукані початкові наближення.

### Комбінований спосіб розв"язування нелінійних рівнянь /метод хорд і дотичних/

Нехай потрібно розв"язати рівняння

$$F(x) = 0, \qquad /2.33/$$

для якого вдалося знайти інтервал, де міститься корінь /інтервал його ізоляції/:

$$\bar{x} \in [a_1, b_1].$$

Розглянемо геометричну інтерпретацію комбінованого методу хорд і дотичних.

Дійсним значенням кореня $\bar{x}$ розглядуваного рівняння є абсциса точки перетину кривої $y = F(x)$ з віссю $x$. Кожне із зображень рис. 2.2,а–г, відповідає одному з чотирьох випадків:

1/ $F'(x) > 0;\ F''(x) > 0;$
2/ $F'(x) > 0;\ F''(x) < 0;$
3/ $F'(x) < 0;\ F''(x) > 0;$
4/ $F'(x) < 0;\ F''(x) < 0.$

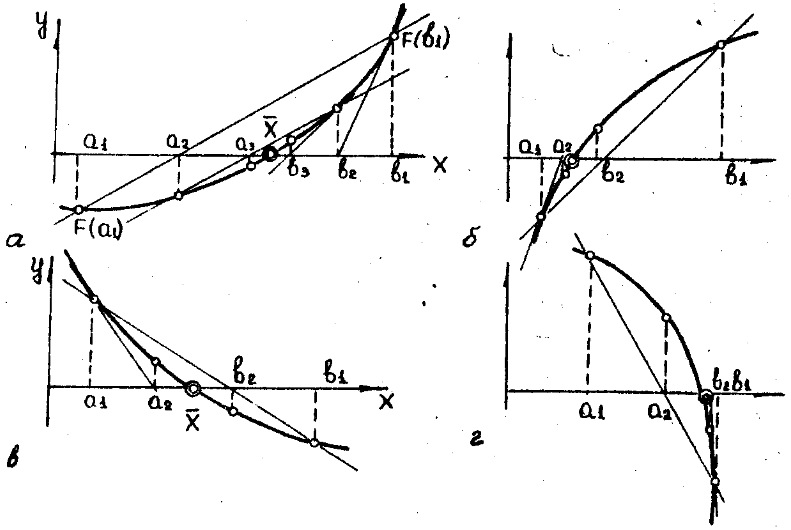


Рис. 2.2. Геометрична інтерпретація методу хорд і дотичних

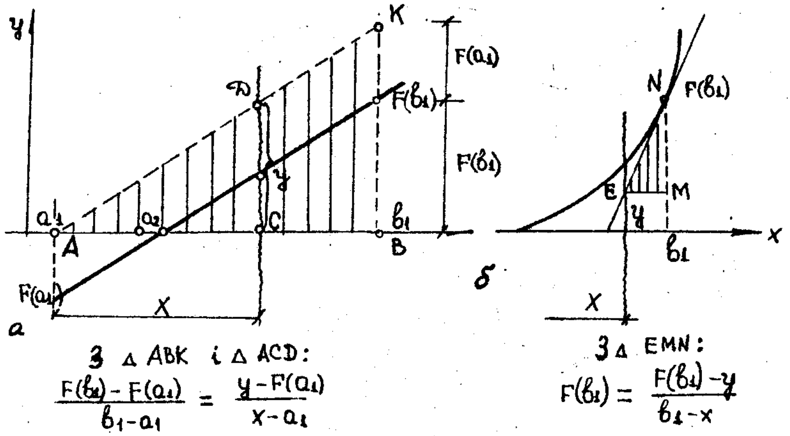


Рис. 2.3. Знаходження кореня рівняння $F(x)=0$ методом хорд і дотичних

Розглянемо перший випадок. Міркування в решті випадків аналогічні.

Проведемо через точки $(a_1, F(a_1))$ та $(b_1, F(b_1))$ пряму /рис. 2.3,а/. З подібних трикутників $\Delta ABK$ і $\Delta ACD$ записуємо рівняння прямої, проведеної через ці точки:

$$\frac{F(b_1)-F(a_1)}{b_1-a_1} = \frac{y-F(a_1)}{x-a_1}.$$

Абсциса точки перетину цієї прямої з віссю $x$ дає перше уточнене значення лівої межі інтервалу ізоляції кореня:

$$a_2 = a_1 - \frac{b_1 - a_1}{F(b_1)-F(a_1)} F(a_1).$$

Отже,

$$\bar{x} \in [a_2, b_1].$$

Тепер проведемо через точку $(b_1, F(b_1))$ дотичну до кривої $y = F(x)$ /рис. 2.3,б/. З $\Delta EMN$:

$$F'(b_1) = \frac{F(b_1)-y}{b_1-x}.$$

Це буде рівняння дотичної. Абсциса точок перетину дотичної з віссю $x$ дає уточнену праву межу інтервалу ізоляції кореня:

$$b_2 = b_1 - \frac{F(b_1)}{F'(b_1)}.$$

Отже,

$$\bar{x} \in [a_2, b_2].$$

На $i$-му кроці ліва межа інтервалу ізоляції визначається виразом

$$a_{i+1} = a_i - \frac{b_i - a_i}{F(b_i) - F(a_i)} F(a_i), \qquad /2.34/$$

права - виразом

$$b_{i+1} = b_i - \frac{F(b_i)}{F'(b_i)}. \qquad /2.35/$$

Інтервал ізоляції кореня слід звужувати доти, поки буде досягнуто нерівності

$$b_i - a_i < \varepsilon.$$

Шуканим значенням кореня буде середнє арифметичне отриманих меж інтервалу ізоляції кореня:

$$\bar{x} = \frac{1}{2} (b_i + a_i). \qquad /2.36/$$

При знаходженні коренів рівняння $F(x) = 0$ в інших випадках /див. рис. 2.2,б–г/ слід уточнити, через яку з точок $(a_1, F(a_1))$ чи $(b_1, F(b_1))$ необхідно провести дотичну.

Якщо на інтервалі ізоляції кореня виконується нерівність $F'(x) F''(x) > 0$, то межі уточнюються за /2.34/ і /2.35/. Якщо $F'(x) F''(x) < 0$, формули /2.34/ і /2.35/ міняють місцями:

$$b_{i+1} = b_i - \frac{b_i - a_i}{F(b_i) - F(a_i)} F(b_i); \qquad /2.34а/$$

$$a_{i+1} = a_i - \frac{F(a_i)}{F'(a_i)}. \qquad /2.35а/$$

## Метод послідовних наближень

Нехай рівняння $F(x) = 0$ переписано в вигляді

$$x = f(x). \qquad /2.37/$$

Це перетворення завжди можна виконати, якщо до лівої та правої частин даного рівняння додати $x$ . Наприклад, якщо $F(x) = 3x - \sin^2 x$, то до рівняння виду /2.37/ можна прийти одним із способів:

1/ $x = 3x - \sin^2 x + x = 4x - \sin^2 x;$

2/ $x = -2x + \sin^2 x;$

3/ $x = 1/3 \sin^2 x.$

Нехай $[a, b]$ відомий інтервал ізоляції шуканого кореня. Уявши за початкове його наближення значення $x_0 = \frac{1}{2}(a+b)$, наступні наближення знаходимо з таких рівностей:

$$x_1 = f(x_0), \; x_2 = f(x_1) \quad \text{і т.д.}$$

Нарешті, дістаємо

$$x_i = f(x_{i-1}).$$

Чисельний метод, в якому відбувається послідовне, крок за кроком уточнення кореня, називається методом ітерацій.

Якщо в результаті ітерацій дістають значення, які все ближче підходять до шуканого кореня рівняння, то відповідний метод називають збіжним, у противному разі – розбіжним.

## Класичний метод ітерацій

Дамо геометричну інтерпретацію класичного методу ітерацій. Розв"язком рівняння /2.37/ є абсциса точки перетину прямої $y=x$ і кривої $y=f(x)$ /рис. 2.4/. Задамося значенням $x_0$, тоді $x_1=f(x_0)$. Проведемо через точку $(x_0, f(x_0))$ пряму, паралельну осі абсцис, до перетину з прямою $y=x$. /Рівнянням $y=f(x_0)$ задається пряма, паралельна осі абсцис, яка відокремлює на осі ординат відрізок $y=f(x_0)$./ Абсциса точки перетину прямих $y=x$ і $y=f(x_0)$ дасть значення $x_1$. Процес повторюється далі в порядку, який показано стрілками на рис. 2.4.

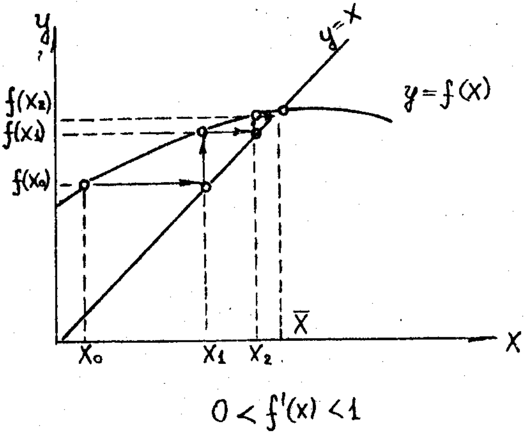


Рис. 2.4. Геометрична інтерпретація класичного методу ітерацій

Очевидно, що послідовність значень $x_0, x_1, x_2 \ldots$ прямує до кореня рівняння. Метод ітерацій збіжний, коли

$$f'(x) < 1, \qquad /2.38/$$

і розбіжний, коли $|f'(x)| \geq 1$.

Важливо, що ітераційні методи забезпечують незалежність точності розв"язку рівняння від похибок округлення на $i$-му кроці ітерацій, оскільки ці похибки не нагромаджуються.

Метод хорд і дотичних збігається швидше, ніж метод послідовних наближень.

Зауважимо, проте, що розв"язати нелінійне рівняння методом хорд і дотичних часто буває неможливо, оскільки згідно з ним потрібно обчислювати значення похідної, яка в кількох точках може не існувати.

Вибір методу зумовлюється конкретним типом розв"язуваного рівняння.

<u>Приклад 2.II</u>. Знайти корені нелінійного рівняння

$$y = x^4 - 3x^3 + 2x^2 - 8$$

з точністю $\varepsilon = 0,04$.

<u>Розв"язання</u>. Для знаходження інтервалів ізоляції коренів скористаємось графічним методом. Щоб побудувати графік, знайдемо точки екстремуму досліджуваної функції:

$$F'(x) = \frac{dy}{dx} = 4x^3 - 9x^2 + 4x.$$

Для цього прирівняємо $F'(x)$ до нуля:

$$4x^3 - 9x^2 + 4x = 0,$$

звідки $x_1 = 0$, $x_2 = 0,6096$, $x_3 = 1,64$.

Інтервали ізоляції коренів розглядуваної функції визначаються, як показано на рис. 2.5. Маємо:

$$x \in [-1,5; -1]\ ;\ x \in [2,5, 3].$$

Наступний етап – уточнення значень коренів до заданого ступеня точності. Скористаємося чисельним методом хорд і дотичних.

Оскільки на інтервалах ізоляції коренів $x$ тангенс кута нахилу графіка функції до осі абсцис, більший за одиницю, то метод послідовних наближень дає розбіжний процес /нерівність $0 < f' <$ I не задовольняється/.

Маємо: нехай $x \in (-1,5; -1)$; тоді

$$F(b_1) = F(-1) = -2;$$
$$F(a_1) = F(-1,5) = 11,6875;$$
$$F'(b_1) = F'(-1) = -17.$$

| x | y |
|---|---|
| -2 | 40 |
| -1,5 | 11,6876 |
| -1,2 | 5,6 |
| -1 | -2,0 |
| -0,5 | -7,0625 |
| 0 | -8, |
| +0,5 | -7,81 |
| +0,6096 | -7,7985 |
| +1,64 | -8,82 |
| +2 | -8, |
| +2,5 | -3,3125 |
| +3 | +10,0 |

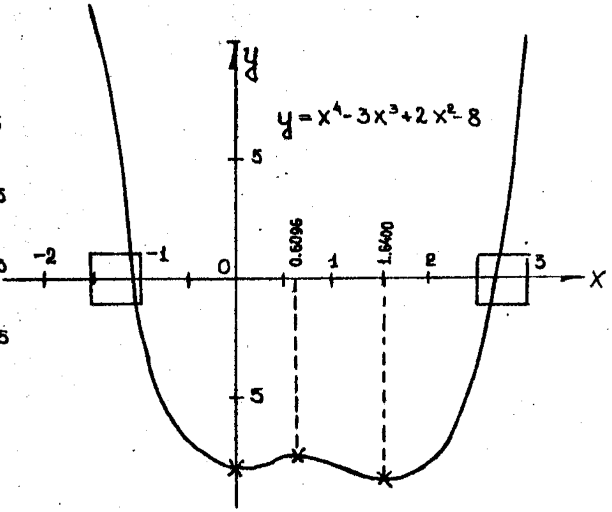


Рис. 2.5. Виділення інтервалів ізоляції коренів функції $y=x^4-3x^3+2x^2-8$

Отже,

$$a_2=\frac{-1,5-(-1,5)}{-2-11,6875}\cdot 11,6875=-1,073;\quad b_2=-1-\frac{-2}{-17}=-1,1176;$$

$$b_2-a_2=0,0386<\varepsilon;$$

звідки

$$x_1=\frac{a_2+b_2}{2}=-1,0983.$$

Аналогічно: нехай $x\in[2,5;3]$; тоді

$$F(b_1)=F(3)=10;$$
$$F(a_1)=F(2,5)=-3,3126;$$
$$F'(b_1)=F'(3)=39.$$

Отже,

$$a_2=2,5-\frac{(3-2,5)(-3,9125)}{10-(-3,3125)}=2,6244,\quad b_2=3-\frac{10}{39}=2,7435,$$

звідки $x\in[2,6244;2,7435]$;

$$F(b_2) = F(2,7435) = 1,7569;$$
$$F(a_2) = F(2,6244) = -1,0142;$$
$$F'(b_2) = F'(2,7435) = 25,83;$$

$$a_3 = 2,6244 - \frac{2,7435-2,6244}{1,7569+1,0142}\cdot(-1,0142) = 2.668; \quad b_3 = 2,7435 - \frac{1,7569}{25,83} = 2,674;$$

$$b_3 - a_3 = 0,007 < \varepsilon,$$

звідки

$$x_2 = \frac{a_3 + b_3}{2} = 2,6715.$$

## 2.3. Методи обробки числових даних

Інженеру часто доводиться мати справу з великими масивами чисел. Отже, йому потрібно ознайомитися з методами їх обробки, такими як інтерполяція, апроксимація кривих, числове диференціювання, числове інтегрування.

### Інтерполяція

Дані, з якими стикається інженер, часто записані в вигляді таблиць /наприклад, коли ці дані отримані експериментально і лише для деяких значень аргументу, або коли обсяг таблиці обмежений і в ній можна навести лише окремі дані/.

Суть інтерполяції полягає в пошуку значень функції в точках, що є проміжними відносно точок, в яких значення функції відомі.

Найпростіший вид інтерполяції – лінійна. Це апроксимація кривої на інтервалі між точками $(x_k, y_k)$ і $(x_{k+1}, y_{k+1})$ прямою, яка проходить через ті самі точки. Рівняння прямої можна записати, розглянувши подібність трикутників $ABC$ і $AED$ /рис. 2.6/. Маємо:

$$\frac{BC}{AC} = \frac{ED}{AD}; \quad \frac{y_{k+1} - y_k}{x_{k+1} - x_k} = \frac{y - y_k}{x - x_k},$$

або

$$y = \frac{y_k(x - x_{k+1}) - y_{k+1}(x - x_k)}{x_k - x_{k+1}}. \qquad /2.39/$$

Якщо використати більше число сусідніх точок і апроксимувати справжню криву складнішою лінією, можна уточнити отриманий результат. Методи пошуку єдиного многочлена $n$-го степеня $P(x)$, який апроксимує функцію $f(x)$ кривою, що проходить через всі $n+1$ задані в таблиці точки $(x_i, y_i)$ / $i = 0, I, \ldots, n$ /, поділяються на три групи: I/ методи Лагранжа; 2/ різницеві методи; 3/ ітераційні методи /метод Ейткіна та інші/. При цьому передбачається, що апроксимуючий многочлен задовольняє таку умову:

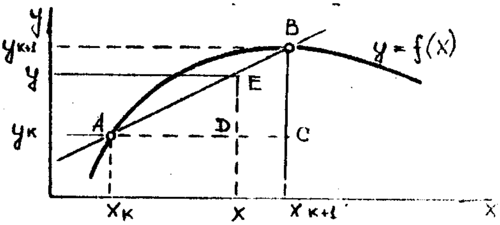


Рис. 2.6. Лінійна інтерполяція

$$P(x_i) = y_i \text{ при } i = 0, I, \ldots, n. \qquad /2.40/$$

## Інтерполяція за Лагранжем

При цій інтерполяції задається $n+1$ табличне значення $(x_i, y_i)$, де $i = 0, I, 2, \ldots, n$. Передбачається, що точки $(x_i, y_i)$ належать кривій $y = f(x)$ в інтервалі $x_0 \leq x \leq x_n$. Інтерполяційний многочлен для цього методу має вигляд:

$$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \ldots + y_n b_n(x), \qquad /2.4I/$$

де $b_i(x)$ – многочлени степеня $n$, коефіцієнти яких можна знайти за допомогою $n+1$ рівняння виду /2.40/:

$$P_n(x_i) = y_i, \text{ де } i = 0, I, \ldots, n.$$

У результаті дістаємо систему рівнянь

$$\begin{cases} y_0 b_0(x_0) + y_1 b_1(x_0) + \ldots + y_n b_n(x_0) = y_0 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ y_0 b_0(x_n) + y_1 b_1(x_n) + \ldots + y_n b_n(x_n) = y_n. \end{cases} \qquad /2.42/$$

Якщо значення $b_j(x_i)$ вибрано так, що

$$b_j(x_i) = \begin{cases} 1 \text{ при } i = j, \\ 0 \text{ при } i \neq j, \end{cases} \qquad /2.43/$$

то рівності /2.42/ задовольняються. Отже, в загальному вигляді

$$b_j(x)=c_j(x-x_0)(x-x_1)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n). \qquad /2.44/$$

Оскільки $b_j(x_j)=1$, то

$$c_j=1/(x_j-x_0)(x_j-x_1)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_n). \qquad /2.45/$$

Нарешті, для шуканого многочлена маємо

$$P_n(x)=\sum_{j=0}^{n} y_j\frac{(x-x_0)(x-x_1)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n)}{(x_j-x_0)(x_j-x_1)\ldots(x_j-x_{j-1})(x_j-x_{j+1})\ldots(x_j-x_n)}. \qquad /2.46/$$

Ввівши позначення

$$L_j(x)=(x-x_0)(x-x_1)\ldots(x-x_{j-1})(x-x_{j+1})\ldots(x-x_n), \qquad /2.47/$$

запишемо отриманий многочлен компактніше:

$$P_n(x)=\sum_{j=0}^{n} y_j\,\frac{L_j(x)}{L_j(x_j)}. \qquad /2.48/$$

## Метод розділених різниць

Найбільш поширений метод Ньютона для інтерполювання вперед відомий в літературі як метод Ньютона – Грегорі. Інтерполяційний многочлен для цього методу

$$P_n(x)=c_0+c_1(x-x_0)+c_2(x-x_0)(x-x_1)+\ldots+c_n(x-x_0)(x-x_1)\ldots(x-x_{n-1}). \qquad /2.49/$$

Коефіцієнти $c_j$ визначаються з рівнянь виду /2.40/

$$P_n(x)=y_i\,,\quad i=0,1,2,\ldots,n.$$

Отже, можна записати систему

$$\begin{cases} c_0=y_0 \\ c_0+c_1(x_1-x_0)=y_1 \\ c_0+c_1(x_2-x_0)+c_2(x_2-x_0)(x_2-x_1)=y_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ c_0+\ldots+c_n(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})=y_n\,. \end{cases} \qquad /2.50/$$

Це лінійна система рівнянь з трикутною матрицею і знайти $c_j$ в ній неважко.

Існує ще простіший спосіб знаходження $c_j$, який грунтується на використанні правих скінченних різниць. Якщо $x$ задано через рівні проміжки аргументу $x_{i+1}-x_i=h$, то в загальному вигляді

$$x_i = x_0 + ih, \text{ де } i = 1,2,\dots,n.$$

Останній вираз дає змогу звести рівняння /2.50/ до вигляду

$$\begin{cases} y_0 = c_0 \\ y_1 = c_0 + c_1 h \\ y_2 = c_0 + c_1(2h) + 2h^2 c_2 \\ \dots\dots\dots\dots\dots\dots \\ y_i = c_0 + c_1 ih + c_2 ih[(i-1)h] + \dots + c_i(i!)h^i, \end{cases} \qquad /2.51/$$

звідки

$$c_0 = y_0$$

$$c_1 = \frac{y_1 - c_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}. \qquad /2.52/$$

Вираз $\Delta y_0$ називається правою скінченною різницею.

Далі маємо

$$c_2 = \frac{1}{2h^2}(y_2 - c_0 - 2hc_1) = \frac{1}{2h^2}[(y_2-y_1)-(y_1-y_0)] = \frac{1}{2h^2}[\Delta(\Delta y_0)] = \frac{\Delta^2 y_0}{2h^2},$$

де $\Delta^2 y_0$ – друга права різниця, тобто різниця різниць,

$$c_j = \frac{\Delta^j y_0}{(j!)h^j}, \qquad /2.53/$$

де

$$\Delta^j y_i = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i \quad (i = 0,1,\dots,n,n-j). \qquad /2.54/$$

Результати обчислення різниць заносимо в табл. 2.3.

Таблиця 2.3

Праві різниці

| $x_i$ | $y_i$ | $\Delta y_i = y_{i+1} - y_i$ | $\Delta^2 y_i = \Delta y_{i+1} - \Delta y_i$ | $\Delta^3 y_i = \Delta^2 y_{i+1} - \Delta^2 y_i$ | $\Delta^4 y_i = \Delta^3 y_{i+1} - \Delta^3 y_i$ |
|---|---|---|---|---|---|
| $x_0$ | $y_0$ | $\Delta y_0$ | | | |
| $x_1$ | $y_1$ | $\Delta y_1$ | $\Delta^2 y_0$ | | |
| $x_2$ | $y_2$ | $\Delta y_2$ | $\Delta^2 y_1$ | $\Delta^3 y_0$ | $\Delta^4 y_0$ |
| $x_3$ | $y_3$ | $\Delta y_3$ | $\Delta^2 y_2$ | $\Delta^3 y_1$ | |
| $x_4$ | $y_4$ | | | | |
| ⋮ | | | | | |

Приклад 2.I2. Маємо таблицю даних $y_i = \sin x_i$/ $i$ = 0,I,2,3,4,5/. Знайти $y$ при $x = 23^0$ методом розділених різниць.

| $i$ | $x_i$, град | $y_i$ |
|---|---|---|
| 0 | IO | 0,I7365 |
| I | 20 | 0,34202 |
| 2 | 30 | 0,50000 |
| 3 | 40 | 0,64279 |
| 4 | 50 | 0,76604 |
| 5 | 60 | 0,86603 |

Розв"язання. За допомогою вихідних даних складемо таблицю різниць /табл. 2.4/.

Таблиця 2.4

| $i$ | $x_i$ | $y_i$ | $\Delta y_i$ | $\Delta^2 y_i$ | $\Delta^3 y_i$ | $\Delta^4 y_i$ | $\Delta^5 y_i$ |
|---|---|---|---|---|---|---|---|
| 0 | IO | 0,I7365 | | | | | |
| | | | 0,I6837 | | | | |
| I | 20 | 0,34202 | | -0,0I039 | | | |
| | | | 0,I5798 | | 0,00480 | | |
| 2 | 30 | 0,50000 | | -0,0I5I9 | | 0,00045 | |
| | | | 0,I4279 | | -0,00435 | | 0,000I8 |
| 3 | 40 | 0,64279 | | -0,0I954 | | 0,00063 | |
| | | | 0,I2325 | | -0,00372 | | |
| 4 | 50 | 0,76604 | | -0,02326 | | | |
| | | | 0,09999 | | | | |
| 5 | 60 | 0,86603 | | | | | |

За $x_0$ можна взяти будь-яке $x_i$ , наприклад $x_1 = 20^0$. Необхідні різниці стоять на діагоналі, яка йде від $x_0$ вниз. Число використаних різниць вищих порядків може бути довільним. Якщо відомо, скільки членів потрібно взяти, їх кількість можна збільшувати доти, поки їх внесок не стане настільки малим, що ним можна буде знехтувати.

Використавши лише першу різницю, дістанемо:

$$y(23) = y + \frac{\Delta y_0}{h}(23 - x_0) = 0,34202 + \frac{0,15798}{10} 3 = 0,38941,$$

Ввівши другу різницю, отримаємо:

$$y(23) = 0,38941 + \frac{\Delta^2 y_0}{1 \cdot 2 h^2}(23 - x_0)(23 - x_1) = 0,39100.$$

За допомогою третьої різниці запишемо:

$$y(23)=0{,}39100+\frac{\Delta^3 y_0}{6h^3}(23-x_0)(23-x_1)(23-x_2)=0{,}39074.$$

Перевага методу буде в тім, що він дозволяє уточнити результат, при цьому немає необхідності починати обчислення спочатку.

### Ітераційні методи інтерполяції

Засновані на повторному використанні простої інтерполяційної схеми. Найбільш відомий метод Ейткіна, суть якого в повторному використанні лінійної інтерполяції. Лінійна інтерполяція виконується за формулою

$$y_{i1}(x)=\frac{1}{x_i-x_0}\left[y_0(x_i-x)-y_i(x_0-x)\right]. \qquad /2.55/$$

Скористаємось нею, щоб, задавшись значенням $x_i$, скласти таблицю функцій $y_i(x)$, $i=1,2,\dots,n$. Використовуючи ці функції, за допомогою лінійної інтерполяції

$$y_{i2}(x)=\frac{1}{x_i-x_1}\left[y_{11}(x)(x_i-x)-y_{i1}(x)(x_1-x)\right] \qquad /2.56/$$

дістаємо нову сім"ю співвідношень. Безпосередньою підстановкою можна показати, що вираз для $y_i(x)$ – це многочлен другого степеня, який описує криві, що проходять через точки $(x_0,y_0),(x_1,y_1)$ і $(x_i,y_i)$. Отримавши многочлени для $y_i$, можна записати вираз для многочлена третього степеня:

$$y_{i3}(x)=\frac{1}{x_i-x_2}\left[y_{22}(x)(x_i-x)-y_{i2}(x)(x_2-x)\right]. \qquad /2.57/$$

Він описує криві, що проходять через точки $(x_0,y_0),(x_1,y_1)$, $(x_2,y_2)$ і $(x_i,y_i)$. Продовжуючи цей процес, діставатимемо значення $y_i(x)$, які наближатимуться до значення $y_{ij}(x)$. Метод Ейткіна не потребує того, щоб значення функції, використовувані для інтерполяції, були розміщені через рівні інтервали.

Нехай, наприклад, різниці задано табл. 2.5.

Таблиця 2.5

| $i$ | $x_i$ | $y_i$ | $y_{i1}$ | $y_{i2}$ | $y_{i3}$ |
|---|---|---|---|---|---|
| 0 | 10 | 0,17365 | | | |
| 1 | 20 | 0,34202 | 0,39253 | | |
| 2 | 30 | 0,50000 | 0,38578 | 0,39051 | – |
| 3 | 40 | 0,64279 | 0,37694 | 0,39019 | 0,39073 |
| 4 | 50 | 0,76604 | 0,36618 | 0,38990 | 0,39072 |
| 5 | 60 | 0,86603 | 0,38962 | 0,39072 | 0,39072 |

Звідси маємо

$$y_{11} = \frac{1}{10}[0,17365(20-23)-0,34202(10-23)] = 0,39253.$$

## Апроксимація кривих

На практиці бувають випадки, коли зв"язок між функцією $y$ і аргументом $x$ явно не задано, тобто його не можна записати у вигляді деякої залежності $y=f(x)$. А інколи ця залежність настільки громіздка, що скористатись нею на практиці дуже важко. Тому буває зручно задати такий зв"язок у вигляді таблиці.

Коли неявний, громіздкий, або табличний зв"язок між функцією $y$ та її аргументом $x$ потрібно записати наближено деякою аналітичною залежністю, говорять, що функцію потрібно апроксимувати.

Існують два основних підходи до апроксимації табличних даних кривими. Перший передбачає, що апроксимуюча крива проходить через усі точки, задані таблицею. Цього вдається досягти за допомогою методів інтерполяції, розглянутих вище. Другий підхід полягає в тому, що табличні дані апроксимуються простою функцією, яка може використовуватися на всьому їх діапазоні, але графік якої не обов"язково проходить через усі точки. Така дія називається підгонкою кривої. Її намагаються провести так, щоб відхилення значень апроксимуючої функції від табличних даних був мінімальним. Найчастіше мінімізується сума квадратів різниць між значеннями апроксимуючої функції, які належать вибраній кривій, і табличними значеннями. Відповідний метод підгонки називається методом найменших квадратів.

## Метод найменших квадратів

Нехай в таблиці задано $n+1$ точку $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ і потрібно знайти апроксимуючу криву $q(x)$ в діапазоні $x_0 \leq x \leq x_n$. У цьому разі похибка в кожній табличній точці

$$\mathcal{E}_i = q(x_i) - y_i . \qquad /2.58/$$

Тоді сума квадратів похибок

$$E = \sum_{i=0}^{n} [q(x_i) - y_i]^2 . \qquad /2.59/$$

Звичайно, функцію $q(x)$ вибирають у вигляді лінійної комбінації "придатних" функцій, тобто таких, які найкраще описують табличні дані:

$$q(x) = c_1 q_1(x) + c_2 q_2(x) + \ldots + c_k q_k(x). \qquad /2.60/$$

Для того щоб значення $E$ було мінімальним, потрібно, щоб

$$\frac{\partial E}{\partial c_1} = \frac{\partial E}{\partial c_2} = \ldots = \frac{\partial E}{\partial c_k} = 0. \qquad /2.61/$$

Оскільки

$$E = \sum_{i=0}^{n} [c_1 q_1(x_i) + c_2 q_2(x_i) + \ldots + c_k q_k(x_i) - y_i]^2 , \qquad /2.62/$$

то ця умова еквівалентна системі рівнянь

$$\left\{ \begin{array}{l} \dfrac{\partial E}{\partial c_1} = 2\sum [c_1 q_1(x_i) + \ldots + c_k q_k(x_i) - y_i] q_1(x_i) = 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dfrac{\partial E}{\partial c_k} = 2\sum [c_1 q_1(x_i) + \ldots + c_k q_k(x_i) - y_i] q_k(x_i) = 0. \end{array} \right. \qquad /2.63/$$

Ці $k$ рівнянь можна записати у вигляді перемножуючих матриць

$$\begin{pmatrix} \sum q_1^2(x_i) & \sum q_1(x_i) q_2(x_i) \ldots & \sum q_1(x_i) q_k(x_i) \\ \vdots & & \vdots \\ \sum q_1(x_i) q_k(x_i) & & \sum q_k^2(x_i) \end{pmatrix} * \begin{pmatrix} c_1 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} \sum q_1(x_i) y_i \\ \vdots \\ \sum q_k(x_i) y_i \end{pmatrix} . \qquad /2.64/$$

Оскільки елементи матриці зліва і елементи вектора справа в /2.64/ знаходяться з табличних даних, то систему $k$-го порядку з $k$ змінними можна розв"язати. Тут придатна будь-яка функція $q(x)$, тільки б вона була лінійна відносно своїх коефіцієнтів.

Іноді таблицю розбивають на кілька частин і для кожної частини добирають окрему апроксимуючу криву. Такий підхід виправданий, коли апроксимуючі дані відповідають різним фізичним станам системи /наприклад, перехід конструкції від стійкого стану до нестійкого/.

## Сплайни

У прикладній математиці почали використовуватися недавно. Як апарат машинобудівного креслення відомі давно. Сплайн - це деяка фізична модель многочлена - гнучка лінійка, яку деформують так, щоб по ній можна було провести криву через задані точки $(x_i, y_i)$. Якщо закріпити таку гнучку лінійку в цих точках, тобто деформувати її відповідним чином, то вона набере форми, при якій нагромаджена в ній пружна енергія /потенціальна енергія/ буде мінімальною.

Використовуючи теорію прогинання бруса при малих деформаціях, можна строго показати, що сплайн - це група сполучених кубічних многочленів, у місцях з"єднання яких перша й друга похідні неперервні. Такі функції називаються кубічними сплайнами. Щоб побудувати кубічний сплайн, потрібно задати коефіцієнти, які однозначно визнача- кубічний многочлен на інтервалі між даними точками.

## Чисельне диференціювання

Іноді потрібно знайти наближене значення похідної функції, заданої таблично. Відповідні формули можна дістати за допомогою ряду Тейлора

$$y(x_0+h)=y(x_0)+hy'(x_0)+\frac{h^2}{2!}y''(x_0)++\frac{h^3}{3!}y'''(x_0)+\ldots+\frac{y^{(n)}(x_0)}{n!}h^n. \quad /2.65/$$

або диференціюванням інтерполяційного виразу.

## Інтерполяційні многочлени

Нехай функцію $y=f(x)$ задано графіком або таблицею значень в деяких вузлових точках. Найпростішим способом наближеного обчислення її похідних є заміна даної функції $y$ многочленом, який у вузлах набуває тих самих значень, що й ця функція. Похідні многочлена й стають наближеними значеннями відповідних похідних функції $y$.

Знайдемо, наприклад, другу похідну $y''$ функції $y$, заданої в трьох точках $l, i, r$, ліва й права з яких на $h$ віддалені від $i$ /мал. 2.7/.

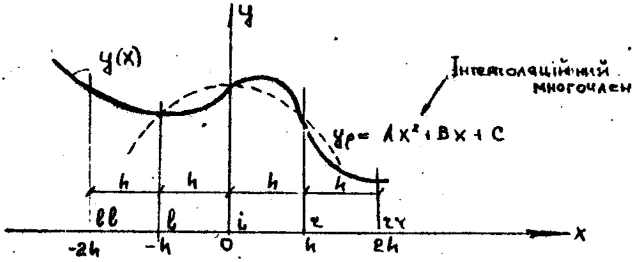


Рис. 2.7. Апроксимація диференційовної функції квадратною параболою

Нехай $y_l, y_i, y_r$ – задані значення функції $y$ відповідно в точках $l, i, r$.

Розглянемо многочлен другого степеня

$$y_P = Ax^2 + Bx + C, \qquad /a/$$

значення якого в точках $l, i, r$ збігається із значенням $y$ в цих точках. Для простоти припустимо, що точка $i$ збігається з початком координат. Тоді

$$\begin{cases} y(-h) = y_l = Ah^2 - Bh + C \\ y(0) = y_i = C \\ y(h) = y_r = Ah^2 + Bh + C. \end{cases}$$

Додамо почленно ці три рівняння, попередньо домноживши друге на 2. Дістанемо:

$$y_l - 2y_i + y_r = 2Ah^2.$$

Запишемо першу й другу похідні многочлена /а/:

$$y' = 2Ax + B, \quad y'' = 2A.$$

Оскільки $y''=2A$, то $y''_i$ апроксимується виразом

$$y''_i = \frac{1}{h^2}(y_l - 2y_i + y_r). \qquad /2.66/$$

Скориставшись інтерполяційними многочленами вищих степенів, аналогічно можна дістати наближені значення похідних вищих порядків. Наприклад, апроксимувавши диференційовну функцію кубічним многочленом

$$y_p = Ax^3 + Bx^2 + Cx + D,$$

дістанемо таке наближення розв"язку:

$$y''_i = \frac{1}{h^3}(-y_l + 3y_i - 3y_r + y_{rr}). \qquad /2.67/$$

## Чисельне інтегрування

Інженеру часто доводиться обчислювати визначені інтеграли чисельними методами. Це буває в тих випадках, коли або не вдається виразити інтеграл в замкненій формі, або вона настільки складна, що доцільніше скористатися чисельним інтегруванням. Чисельне інтегрування являє собою стійкий процес, воно зменшує вплив похибок у вихідних даних на кінцевий результат. В його основі лежить наближене обчислення площі під кривою, яка описує підінтегральну функцію. У загальному вигляді задача формулюється як знаходження значення

$$I = \int_a^b f(x)\,dx. \qquad /2.68/$$

Методи чисельного інтегрування класифікуються залежно від того, через рівні чи нерівні проміжки задані значення аргументу. Так, метод Ньютона-Котеса потребує, щоб значення $x$ були задані із сталим кроком, а метод Гаусса не накладає такого обмеження.

### Метод прямокутників

У цьому методі площу під інтегральною кривою шукаємо як суму прямокутників. Чим менший крок розбиття області визначення незалежної змінної, тим точніше значення інтеграла /рис. 2.8,а/

$$I = \int_a^b f(x)dx = h(y_0 + y_1 + y_2 + \ldots + y_{n-1}). \qquad /2.69/$$

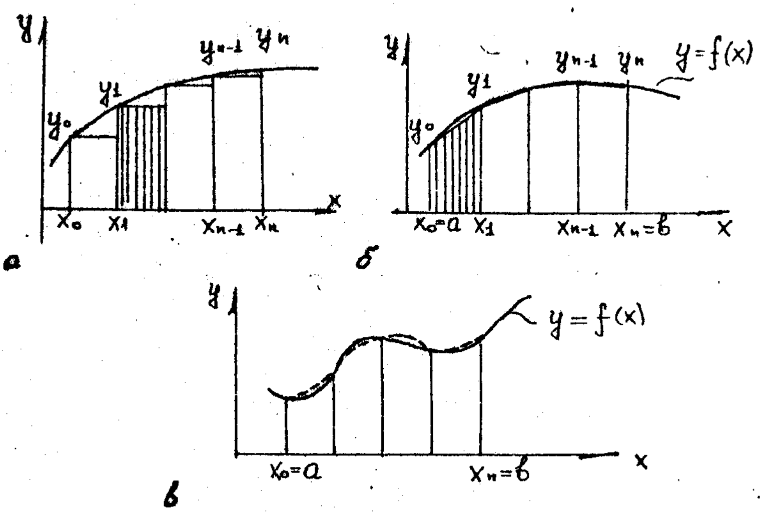


Рис. 2.8. Чисельне інтегрування за методом прямокутників /а/; трапецій /б/; Сімпсона /в/

**Приклад 2.14.** Обчислити визначений інтеграл $I = \int\limits_2^4 \frac{dx}{\sqrt{x^2-1}}$ методом прямокутників.

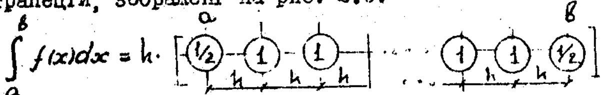
**Розв"язання.** Візьмемо крок розбиття $h = 0,5$. Позначимо $y = \frac{1}{\sqrt{x^2-1}}$. Знайдемо значення $y$, коли $x \in [2,4]$ з кроком $h = 0,5$.

| $x$ | 2 | 2,5 | 3 | 3,5 | 4 |
|---|---|---|---|---|---|
| $y$ | 0,577 | 0,436 | 0,354 | 0,298 | 0,258 |

$$I = 0,5\,(0,577 + 0,434 + 0,354 + 0,298) = 0,8315.$$

Метод Ньютона-Котеса

Дістаємо, інтегруючи методом трапецій, суть якого становить лінійна апроксимація підінтегральної функції /рис. 2.8,б/. Коефіцієнти, з якими значення $f(x)$ у вузлових точках входять до формули трапецій, зображені на рис. 2.9.



$$\int\limits_a^b f(x)dx = h\cdot\Big[\overset{a}{\textcircled{\tiny 1/2}}\underbrace{-\textcircled{1}}_{h}\underbrace{-\textcircled{1}}_{h}\underbrace{-}_{h}\cdots\textcircled{1}\underbrace{-\textcircled{1}}_{h}\underbrace{-\overset{b}{\textcircled{\tiny 1/2}}}_{h}\Big] + O(h^2) \qquad /2.70/$$

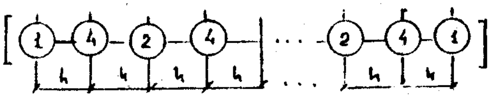
Рис. 2.9. Формула трапецій

Значення інтеграла $I = \int_2^4 \frac{dx}{\sqrt{x^2-1}}$ за методом трапецій таке:

$$I = \frac{0,5}{2} /0,577+2\cdot 0,436+2\cdot 0,354+2\cdot 0,298+0,258/ = 0,7527.$$

Природно напрощується думка, що можна підвищити точність результату, замінивши лінійну апроксимацію апроксимацією кривих вищого порядку, наприклад параболами. Існує відповідна формула, відома як правило Симпсона /мал. 2.8,в/. Коефіцієнти, з якими значення $f(x)$ у вузлових точках входять до формули Симпсона, зображені на мал. 2.10.

$\int_a^b f(x)\,dx = \frac{h}{3}$



$+ O(h^4)$ /2.71/

Рис. 2.10. Формула Симпсона /парабол/

Значення інтеграла $I = \int_2^4 \frac{dx}{\sqrt{x^2-1}}$ за методом Симпсона таке:

$$I = \frac{1}{3} \cdot 5 /0,574+4\cdot 0,436+2\cdot 0,354+4\cdot 0,298+0,258/ = 0,7465.$$

### Метод Гаусса

Знявши обмеження, що значення функції мають задаватися зі сталим кроком по осі $x$, можна вибрати крок так, щоб похибки апроксимації були якомога менші. У цьому й суть методу Гаусса.

## 2.4. Чисельні методи розв"язування лінійних диференціальних рівнянь з початковими і крайовими умовами

Відомо, яку велику роль відіграють диференціальні рівняння для розв"язування прикладних задач. Адже переважна більшість законів техніки й фізики формулюється саме у вигляді диференціальних рівнянь. Розрахунок потоків енергії або руху тіл зводиться врешті до розв"язування диференціальних рівнянь. Лише окремі з них можна розв"язати в замкненому вигляді. Найчастіше доводиться звертатися до чисельних методів та сучасних ЕОМ.

Диференціальним називається рівняння, яке об"єднує незалежну змінну $x$, шукану функцію $y = f(x)$ і її похідні $y', y'', y''', \dots, y^{(n)}$. Записується воно у вигляді

$$F(x, y, y', y'', \dots, y^{(n)}) = 0. \qquad /2.72/$$

Диференціальні рівняння діляться на звичайні та в частинних похідних.

Якщо $y = f(x)$ є функцією однієї незалежної змінної і похідні беруться за нею, то диференціальне рівняння називається звичайними.

Диференціальне рівняння в частинних похідних містить кілька незалежних змінних і похідні за ними, які називають частинними.

Порядком диференціального рівняння називається порядок найвищої похідної, яка входить до нього.

Розв"язком, або інтегралом, диференціального рівняння називається будь-яка функція $y = f(x)$, підставивши яку в рівняння /2.72/, дістанемо тотожність.

## Задача Коші і крайова задача

Для розв"язування звичайного диференціального рівняння потрібні значення залежної змінної і /або/ її похідних при деяких значеннях незалежної змінної. Якщо ці додаткові умови задаються при одному й тому самому значенні незалежної змінної, то це задача з початковими умовами, або задача Коші. Якщо додаткові умови задаються при двох або більше значеннях незалежної змінної, то така задача називається крайовою.

У задачі Коші додаткові умови називаються початковими, а в крайовій – граничними. Часто в задачі Коші роль незалежної змінної відіграє час. Прикладом може служити задача про вільні коливання тіла, підвішеного на пружині.

Якщо додаткові умови задаються у вигляді значень переміщень і швидкості при $t = 0$, то також маємо задачу Коші, а коли однією з умов задається переміщення наприкінці деякого проміжку часу, маємо крайову задачу.

У крайових задачах незалежною змінною часто буває довжина. Відомим прикладом такого роду є диференціальне рівняння, що описує деформацію пружного стержня:

$$\frac{y''}{(1+y'^2)^{3/2}} = -\frac{M}{EJ}, \qquad /2.73/$$

або, як його часто називають, наближене диференціальне рівняння вигнутої осі стержня. У будівельних конструкціях величина $(y')^2$ мала порівняно з одиницею, і нею здебільшого нехтують, записуючи /2.73/ у вигляді

$$y'' = -\frac{M}{EJ}. \qquad /2.74/$$

Знак "–" ставиться, коли вісь $y$ напрямлена вниз. Тоді граничні умови звичайно задаються на обох кінцях стержня.

При чисельному розв"язуванні задачі розв"язок шукають в окремих /дискретних/ точках області визначення незалежного аргументу $x$. Звичайно крок вважають сталим $(h = const)$.

Чисельне розв"язування задачі Коші, як широко використовуване в різних галузях науки і техніки, протягом багатьох років було об"єктом пильної уваги вчених, і кількість розроблених для нього методів дуже велика. Спинимося на двох групах чисельних методів розв"язування задачі Коші.

1. Однокрокові методи. Вони потребують інформації лише про один попередній крок. Це метод Ейлера і метод Рунге – Кутта.

2. Методи прогнозу й корекції /багатокрокові/. У них для пошуку наступної точки кривої $y=f(x)$ потрібна інформація більш, як про одну з попередніх точок. Це методи Мілна, Адамса – Башфорта, Хемінга, Крилова. Багатокрокові методи грунтуються на заміні диференціального рівняння

$$y'(x)-f(x,y(x))=0;\ y(a)=S \qquad /2.75/$$

при сталому кроці $h$ різницевим рівнянням $k$-го порядку

$$\sum_{j=0}^{k} a_j^{(k)} y_{n+j} - h\sum_{j=0}^{k} b_j^{(k)} f(x_{n+j}, y_{n+j}) = 0; \qquad /2.76/$$

$$(a_0^k)^2 + (b_0^k)^2 \neq 0, \quad n = 0,1,2,\ldots,n. \qquad /2.77/$$

При цьому значення $y_0 = S, y_1, \ldots, y_{k-1}$ будуть заданими, тобто для виконання багатокрокових методів потрібна початкова процедура.

## Однокрокові методи. Метод Ейлера

Застосовується для розв"язування диференціальних рівнянь першого порядку, які мають вигляд

$$y' = f(x,y), \text{ де } y' = dy/dx \qquad /2.78/$$

при початкових умовах $y(x_0) = y_0$.

Метод Ейлера базується на розвиненні $y$ у ряд Тейлора в околі точки $x_0$ /див. /2.65// /рис. 2.11/:

$$y(x_0+h)=y(x_0)+hy'(x_0)+\frac{1}{2}h^2y''(x_0)+\ldots+\frac{y^{(n)}(x_0)}{n!}h^n. \qquad /2.79/$$

Якщо $h$ мале, то члени, які мають $h$ в другому та вищих степенях, будуть малими більш високих порядків і ними можна знехтувати.

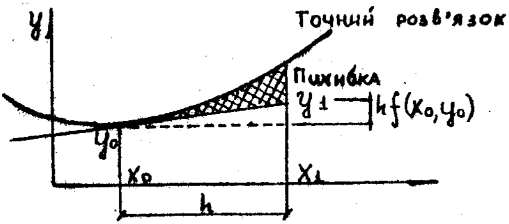


Рис. 2.11. До розв"язування диференціальних рівнянь методом Ейлера

Тоді

$$y(x_0+h)=y(x_0)+hy'(x_0) \qquad /2.80/$$

або

$$y_{n+1}=y_n+hf(x_n,y_n),\ n=1,2,\ldots .$$

Значення $y'(x)$ знаходимо з диференціального рівняння, підставивши в нього початкові умови. Похибка методу Ейлера має порядок $h^2$, оскільки члени, що містять $h$ в другому та вищому степенях відкидаються.

## Метод Рунге - Кутта

Щоб дістати в ряду Тейлора член $n$-го порядку, необхідно якось обчислити $n$-ну похідну залежної змінної. Очевидно, чим вищий порядок обчислюваної похідної, тим більше додаткових обчислень потрібно зробити всередині інтервала.

Метод Рунге - Кутта дає набір формул для знаходження координат внутрішніх точок, потрібних для реалізації цієї ідеї. Маємо:

$$y_{n+1}=y_n+\frac{k_0+2k_1+2k_2+k_3}{6}, \qquad /2.81/$$

де

$$K_0 = hf(x_n, y_n); \qquad /2.82/$$

$$k_1 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_0); \qquad /2.83/$$

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1); \qquad /2.84/$$

$$k_3 = hf(x_{n+h}, y_n + k_2). \qquad /2.85/$$

Приклад 2.15. Розв"язати лінійне диференціальне рівняння типу $y' = f(x, y)$

$$\frac{dy}{dx} = 2x^2 + 2y$$ при початкових умовах:

$$y(0) = 1, \quad 0 \le x \le 1 \quad і \quad h = 0,1.$$

Розв"язання. Дане рівняння має точний розв"язок $y = 1,5e^{2x} - x^2 - x - 0,5$. Це саме рівняння можна розв"язати чисельно – методом Ейлера чи Рунге – Кутта, порівнявши точності знайдених розв"язків /табл.2.6/.

Таблиця 2.6

| $n$ | $x_n$ | Чисельний метод | | Точний розв"язок |
|---|---|---|---|---|
| | | Ейлера | Рунге – Кутта | |
| 0 | 0 | I,0 | I,0 | I,0 |
| I | 0,I | I,2 | I,222I | I,222 |
| 2 | 0,2 | I,442 | I,4977 | I,4977 |
| 3 | 0,3 | I,7384 | I,8432 | I,8432 |
| 4 | 0,4 | 2,I04I | 2,2783 | 2,2783 |
| 5 | 0,5 | 2,5569 | 2,8274 | 2,8274 |
| 6 | 0,6 | 3,II83 | 3,520I | 3 5202 |
| 7 | 0,7 | 3,8I39 | 4,3927 | 4,3928 |
| 8 | 0,8 | 4,6747 | 5,4894 | 5,4895 |
| 9 | 0,9 | 5,7376 | 6,8643 | 6,8645 |
| I0 | I,0 | 7,0472 | 8,5834 | 8,5836 |

## Багатокрокові методи розв"язування задачі Коші /Методи прогнозу та корекції/. Метод Адамса

Застосовується, якщо визначено "початковий" відрізок, тобто коли відомі функції в перших $n$ точках / $x = 0, 1, 2, \ldots, n$ /.

Нехай на відрізку $x_0 \le x \le x_n$ потрібно знайти розв"язок диференціального рівняння /2.78/ $y' = f(x,y)$ при початковій умові $y(x_0) = y_0$. Нехай відомі $p$ значень наближеного розв"язку $y_1, y_2, \ldots, y_n$ у точках сітки $x_1, x_2, \ldots, x_n$, причому $x_i = x_0 + ih$.

Значення наближеного розв"язку можна знайти, використовуючи однокрокові методи /Ейлера, Рунге – Кутта тощо/.

Подамо рівняння /2.78/ в інтегральній формі:

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))\,dx, (n = p, p+1, \ldots). \qquad /2.86/$$

Замінимо підінтегральну функцію $f(x, y(x))$ поліномом $P(x)$, який має степінь $p$ і збігається в $p+1$ точці з відповідними значеннями $f_i = f(x_i, y_i)$. У результаті маємо

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} P(x)\,dx. \qquad /2.87/$$

Узявши за $P(x)$ поліном Ньютона інтерполювання назад

$$P(x) = f_n + t\Delta f_n + \frac{t(t+1)}{2!}\Delta^2 f_n + \ldots + \frac{1}{p!} t(t+1)\ldots(t+p-1)\Delta^p f_n, \qquad /2.88/$$

/$\Delta f_n = f_n - f_{n-1}$ – праві різниці, обчислювані згідно з алгоритмом у табл. 2.3; $t = \frac{x - x_n}{h}$ /, дістанемо таку формулу послідовного знаходження наближень:

$$y_{n+1} = y_n + h\sum_{k=0}^{p} \alpha_k \Delta^k f_n, \ (n = p, p+1, \ldots), \qquad /2.89/$$

де

$$\alpha_0 = 1; \ \alpha_k = \frac{1}{k!}\int_0^1 t(t+1)\ldots(t+k-1)\,dt \quad (k = 1, 2, \ldots, p).$$

Найбільш уживану розрахункову формулу дістаємо при $p = 3$:

$$y_{n+1} = y_n + h\left(f_n + \frac{1}{2}\Delta f_n + \frac{5}{12}\Delta^2 f_n + \frac{3}{8}\Delta^3 f_n\right). \qquad /2.90/$$

Алгоритм, заданий формулою /2.89/ або /2.90/, являє собою екстраполяційний метод Адамса.

Розглянемо таблиці обчислень правих /табл. 2.7/, і лівих /табл. 2.8/ скінченних різниць величин.

Праві різниці позначаються $\Delta$ /"дельта"/, ліві – $\nabla^n$ /"набла"/.

Таблиця 2.7

| $i$ | $y_i$ | $\Delta y_i$ | $\Delta^2 y_i$ | $\Delta^3 y_i$ |
|---|---|---|---|---|
| 0 | $y_0$ | $\Delta y_0 = y_1 - y_0$ | $\Delta^2 y_0 - \Delta y_1 - \Delta y_0$ | $\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$ |
| 1 | $y_1$ | $\Delta y_1 = y_2 - y_1$ | $\Delta^2 y_1 = \Delta y_2 - \Delta y_1$ | |
| 2 | $y_2$ | $\Delta y_2 = y_3 - y_2$ | | |
| 3 | $y_3$. | | | |

Таблиця 2.8

| $i$ | $y_i$ | $\nabla y_i$ | $\nabla^2 y_i$ | $\nabla^3 y_i$ |
|---|---|---|---|---|
| 0 | $y_0$ | $\nabla y_1 = y_0 - y_1$ | | |
| 1 | $y_1$ | $\nabla y_2 = y_1 - y_2$ | $\nabla^2 y_2 = \nabla y_1 - \nabla y_2$ | |
| 2 | $y_2$ | $\nabla y_3 = y_2 - y_3$ | $\nabla^2 y_3 = \nabla y_2 - \nabla y_3$ | $\nabla^3 y_3 = \nabla^2 y_2 - \nabla^2 y_3$ |
| 3 | $y_3$ | | | |

Приклад 2.16. Розв"язати диференціальне рівняння

$$y' = \frac{sh(0,5y+x)}{1,5} + 0,5y$$ на інтервалі $[0;0,3]$, якщо $y(0) = 0$.

Розв"язання. Нехай знайдено "початковий" відрізок /значення функції в перших чотирьох точках/ за допомогою методу Ейлера:

| $k$ | $x_k$ | $y_k$ |
|---|---|---|
| 0 | 0 | 0 |
| I | 0,05 | 0,000846 |
| 2 | 0,I0 | 0,003432 |
| 3 | 0,I5 | 0,007838 |

Тоді знаходимо

$$y_1 = y(x_1) = y(x_0 + h);$$
$$y_2 = y(x_2) = y(x_0 + 2h);$$
$$y_3 = y(x_3) = y(x_0 + 3h).$$

За допомогою $y_1, y_2, y_3$ обчислюємо

$$q_0 = hy'_0 = hf(x_0, y_0); \quad q_1 = hy'_1 = hf(x_1, y_1);$$
$$q_2 = hy'_2 = hf(x_2, y_2); \quad q_3 = hy'_3 = hf(x_3, y_3).$$

Далі обчислюємо праві скінченні різниці величини $q$ згідно в табл. 2.8:

1-ша скінченна різниця $\Delta q_{k-1} = q_k - q_{k-1};$

2-га скінченна різниця $\Delta^2 q_{k-2} = \Delta q_{k-1} - \Delta q_{k-2};$

3-тя скінченна різниця $\Delta^3 q_{k-3} = \Delta^2 q_{k-2} - \Delta^2 q_{k-3}.$

Приріст значень функції знаходимо за екстраполяційною формулою Адамса /2.69/:

$$\Delta y_k = q_k + \frac{1}{2}\Delta q_{k-1} + \frac{5}{12}\Delta^2 q_{k-2} + \frac{3}{8}\Delta^3 q_{k-3}. \qquad /а/$$

Значення функції на наступному кроці

$$y_{k+1} = y_k + \Delta y_k. \qquad /б/$$

У разі потреби значення $\Delta y_k$ коректують за інтерполяційною формулою Адамса:

$$\Delta y_k = q_k + \frac{1}{2}\Delta q_k - \frac{1}{12}\Delta^2 q_{k-1} - \frac{1}{24}\Delta^3 q_{k-2}. \qquad /в/$$

Формули /а/, /б/ мають достатньо велику точність.

Для ЕОМ екстраполяційна та інтерполяційна формули Адамса мають відповідно такий вигляд:

$$y_{k+1} = y_k + \frac{h}{24}(55y'_k - 59y'_{k-1} + 37y'_{k-2} - 9y'_{k-3}, \qquad /2.91/$$

$$y_{k+1} = y_k + \frac{h}{24}(9y'_{k+1} + 19y'_k - 5y'_{k-1} + y'_{k-2}). \qquad /2.92/$$

Результати обчислень наведені в табл. 2.9.

Таблиця 2.9

| $k$ | $x_k$ | $y_k$ | $\Delta y_k$ | $q_k=hf(x_k,y_k)$ | $\Delta q_k$ | $\Delta^2 q_k$ | $\Delta^3 q_k$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 |  | 0 | 0,00I702 | 0,000078 | 0,000007 |
| I | 0,05 | 0,000846 |  | 0,00I702 | 0,00I780 | 0,000085 | 0,000009 |
| 2 | 0,I0 | 0,003432 |  | 0,003482 | 0,00I865 | 0,000094 | 0,0000II |
| 3 | 0,I5 | 0,007838 | 0,0063I8 | 0,005347 | 0,00I959 | 0,0000I05 | 0,0000II |
| 4 | 0,20 | 0,0I4I56 | 0,00833 | 0,00730 | 0,002064 | 0,0000II6 |  |
| 5 | 0,25 | 0,022500 | 0,0I045 | 0,00940 | 0,002I80 |  |  |
| 6 | 0,30 | 0,03290 | 0,0I269 | 0,0II50 |  |  |  |

## 2.5. Метод скінченних різниць

### Методи розв"язування крайових задач

Прикладом крайової задачі може бути звичайне диференціальне рівняння другого порядку

$$\frac{d^2y}{dx^2}=f(x,y,y') \qquad /2.93/$$

за таких граничних умов:

$$y(a)=A,\ y(b)=B.$$

Методи розв"язування рівнянь вищих степенів поділяються на дві групи:

I/ методи, засновані на заміні розв"язування крайової задачі розв"язуванням кількох задач Коші;

2/ методи, в яких застосовується скінченно-різницева форма диференціального рівняння.

### Скінченно-різницеві методи розв"язування диференціальних рівнянь вищих порядків

При розв"язуванні задач цим методом розглядається не континуум точок площини, а зчисленну множину окремих точок $P_{ij}=(x_i,y_i)$.

Важливо, що скінченно-різницеві методи дають змогу звести розв"язування крайової задачі до розв"язування системи алгебраїчних рівнянь. Якщо розв"язується крайова задача

$$y''=f(x,y,y')$$

за умов $y(a)=A$ та $y(b)=B$, інтервал $[a,b]$ можна розбити на $n$ рівних частин:

$$x_i = x_0 + ih, \; i = 1,2,\ldots,n, \qquad /2.94/$$

де $x_0 = a; x_n = b$, а $h = \frac{b-a}{n}$.

У точках $x_i$, названих вузлами, потрібно дістати значення $y_i$.

Похідні, які зустрічаються в розглядуваному диференціальному рівнянні, необхідно апроксимувати. Апроксимацію можна по-перше, виконати, розклавши функцію в степеневий ряд Тейлора. Крім того, можна замінити похідні функції похідними многочлена, який збігається з функцією у вузлових точках. Обома способами дістаємо формули для наближеного обчислення похідних через різницеві відношення, причому перший додатково дає змогу оцінити похибку.

Різницеві відношення бувають ліві, центральні, праві [8].

Коефіцієнти, з якими вузлові значення функції входять до виразу для похідних, обчислених відповідно за лівими, центральними, правими різницями, показані на рис. 2.12-2.14.

Через $D$ позначений широко уживаний в математиці оператор

$$D = \frac{d}{dx}, \qquad /2.95/$$

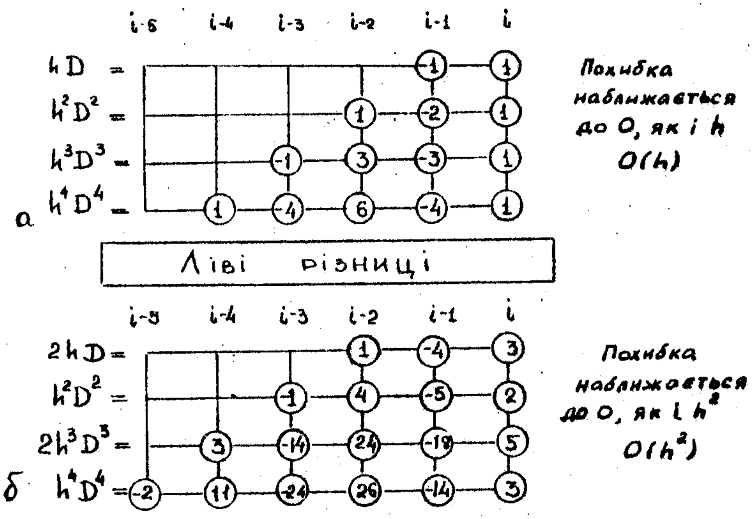
який підпорядкований основним алгебраїчним законам і часто застосовується як символічне число [7].

Рис. 2.12, 2.14 включають формули для одностороннього /або лівого, або правого/ чисельного диференціювання з похибками порядку $h$ та $h^2$.

Центральні різниці, обчислені за вузлами, які містяться симетрично відносно $i$, приводять до формул, на порядок точніших /рис. 2.13/.

Центральні різниці часто застосовуються при розв"язуванні крайових задач. Геометричний зміст центральних різниць полягає в тому, що нахил кривої в точці $P_i$ замінюється нахилом хорди $P_r P_l$, а не $P_r P_i$ або $P_i P_l$, як у випадку правих або лівих різниць /рис. 2.15/. Застосування центральних різниць приводить до точніших формул для заміни похідних, ніж застосування правих чи лівих.

Коефіцієнти, з якими значення функції у вузлових точках входять до формул для обчислення похідних через центральні різниці, показані на рис. 2.13 /а – з точністю $O(h^2)$; б – з подвійною точністю $O(h^4)$/.



а

| | i-5 | i-4 | i-3 | i-2 | i-1 | i |
|---|---|---|---|---|---|---|
| $hD =$ | | | | | -1 | 1 |
| $h^2D^2 =$ | | | | 1 | -2 | 1 |
| $h^3D^3 =$ | | | -1 | 3 | -3 | 1 |
| $h^4D^4 =$ | | 1 | -4 | 6 | -4 | 1 |

Похибка наближається до 0, як і $h$ $O(h)$

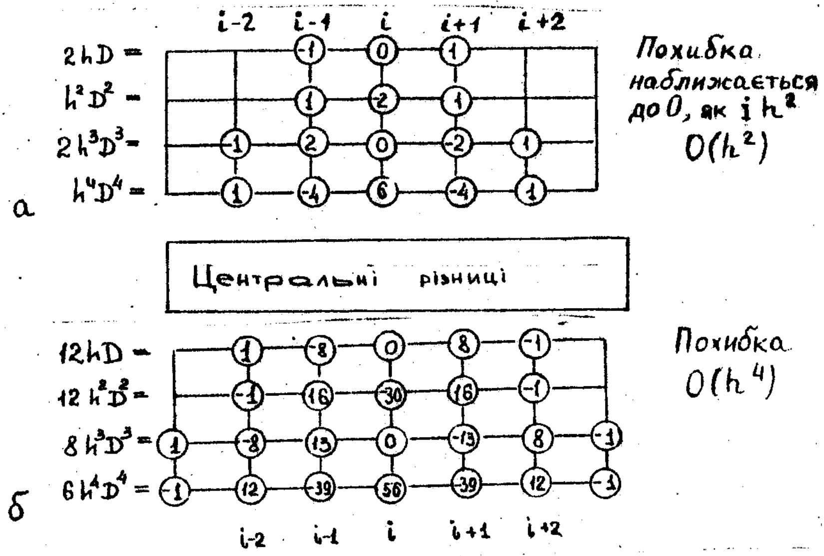
Ліві різниці

б

| | i-5 | i-4 | i-3 | i-2 | i-1 | i |
|---|---|---|---|---|---|---|
| $2hD =$ | | | | 1 | -4 | 3 |
| $h^2D^2 =$ | | | -1 | 4 | -5 | 2 |
| $2h^3D^3 =$ | | 3 | -14 | 24 | -18 | 5 |
| $h^4D^4 =$ | -2 | 11 | -24 | 26 | -14 | 3 |

Похибка наближається до 0, як і $h^2$ $O(h^2)$

Рис. 2.12. Ліві різницеві оператори



а

| | i-2 | i-1 | i | i+1 | i+2 |
|---|---|---|---|---|---|
| $2hD =$ | | -1 | 0 | 1 | |
| $h^2D^2 =$ | | 1 | -2 | 1 | |
| $2h^3D^3 =$ | -1 | 2 | 0 | -2 | 1 |
| $h^4D^4 =$ | 1 | -4 | 6 | -4 | 1 |

Похибка наближається до 0, як і $h^2$ $O(h^2)$

Центральні різниці

б

| | | i-2 | i-1 | i | i+1 | i+2 | |
|---|---|---|---|---|---|---|---|
| $12hD =$ | | 1 | -8 | 0 | 8 | -1 | |
| $12h^2D^2 =$ | | -1 | 16 | -30 | 16 | -1 | |
| $8h^3D^3 =$ | 1 | -8 | 13 | 0 | -13 | 8 | -1 |
| $6h^4D^4 =$ | -1 | 12 | -39 | 56 | -39 | 12 | -1 |

Похибка $O(h^4)$

Рис. 2.13. Центральні різницеві оператори

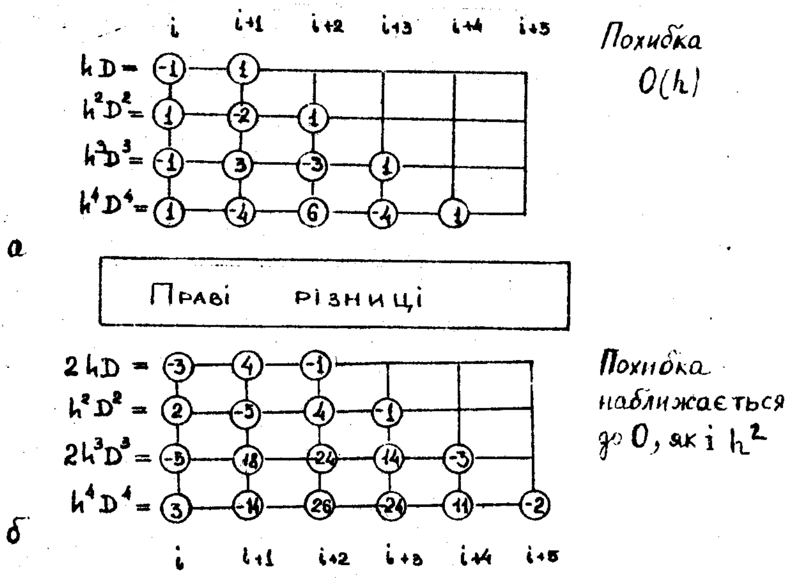


Рис. 2.14. Праві різницеві оператори

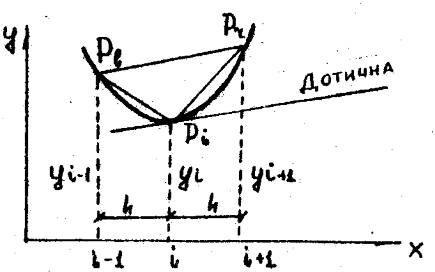


Рис. 2.15. Геометричний зміст центральних різниць

Наприклад, $y''$ можна замінити через центральні різниці виразом з подвійною точністю /див. рис. 2.13/:

$$y''_i = -y_{i-2} + 16 y_{i-1} - 30 y_i + 16 y_{i+1} - y_{i+2}.$$

Якщо відомі координати вузлів, то, застосовуючи скінченно-різницеві вирази для похідних, диференціальне рівняння можна подати у вигляді різницевого. Записавши це різницеве рівняння для $i$ = 1, 2, ..., $n$ при двох граничних умовах, зведемо задачу до розв"язування системи $n$ −1 алгебраїчних рівнянь з $n$ −1 змінною $y_i$.

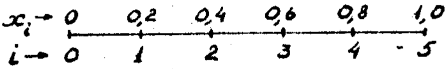
Якщо вихідне диференціальне рівняння лінійне, то задача зводиться до розв"язування системи лінійних алгебраїчних рівнянь, а якщо нелінійне – до розв"язування системи нелінійних алгебраїчних або трансцендентних рівнянь.

Приклад 2.17. Розв"язати диференціальне рівняння $y'' = 2x + 3y$, якщо задано граничні умови $y(0) = 0$; $y(1)$ = 1 та крок $h$ = 0,2.

Розв"язання. У різницевій формі це рівняння має вигляд

$$\frac{1}{0{,}04}(y_{i+1} - 2y_i + y_{i-1}) = 2x_i + 3y_i.$$

Розіб"ємо область визначення аргументу $x$ на рівні частини з кроком $h$ = 0,2.



Дістанемо ряд окремих точок, в яких і будемо шукати розв"язок диференціального рівняння. Для окремих точок 1, 2, 3, 4 запишемо відповідні різницеві рівняння, скориставшись центральними різницями.

При $i = 1$ $\quad \frac{1}{0{,}04}(y_2 - 2y_1 + y_0) = 2x_1 + 3y_1$;

$$y_2 - 2y_1 - y_0 = 0{,}08x_1 + 0{,}12y_1 = 0{,}08 \cdot 0{,}2 + 0{,}12y_1;$$

$$-2{,}12y_1 + y_2 = 0{,}016.$$

Користуючись цією формулою та граничними умовами, запишемо систему чотирьох рівнянь з чотирма змінними:

$$\begin{array}{ll} i=1 & \\ i=2 & \\ i=3 & \\ i=4 & \end{array} \begin{cases} -2{,}12y_1 + y_2 = 0{,}016 \\ y_3 - 2{,}12y_2 + y_1 = 0{,}032 \\ y_4 - 2{,}12y_3 + y_2 = -0{,}064 \\ -2{,}12y_4 + y_3 = -0{,}936. \end{cases}$$

Корені цієї системи дадуть значення $y$ в окремих точках. Порівняємо їх з відповідними значеннями точного розв"язку

$$y(x) = \frac{5}{3}\,\frac{sh\sqrt{3x}}{sh\sqrt{3}} - \frac{2}{3}$$

/табл. 2.10/.

Таблиця 2.10

| $x_i$ | Значення розв"язку $y_i$ | |
|---|---|---|
| | точного | чисельного |
| 0 | 0 | 0 |
| 0,4 | 0,1912 | 0,1897 |
| 0,8 | 0,6088 | 0,6073 |
| I | I | I |

Приклад 2.18. Дано диференціальне рівняння прогину бруса зі сталим поперечним перерізом /рис. 2.16/.

$$\frac{M_x}{EJ} = \frac{PL^2}{EJ}\left(\frac{1}{L} - \frac{x}{L^2}\right),$$

а також початкові умови: $y(0) = 0$, $y'(0) = 0$, $L$ = I м, $PL^2/EJ = 2$, де $L$ – довжина бруса, $PL^2/EJ$ – узагальнююча характеристика твердості бруса.

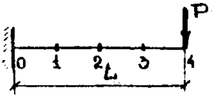


Рис. 2.16. Розрахункова схема бруса

Побудувати лінію прогину $y(x)$ бруса, користуючись наближеним виразом для кривизни

$$y'' = \frac{PL^2}{EJ}\left(\frac{1}{L} - \frac{x}{L^2}\right). \qquad /2.96/$$

Розв"язання. Різницевий аналог рівняння /2.96/ має вигляд

$$\frac{1}{h^2}(y_{i+1} - 2y_i + y_{i-1}) = \frac{PL^2}{EJ}\left(\frac{1}{L} - \frac{x}{L^2}\right)$$

Нехай $h$ = 0,25 м. Тоді маємо:

$$8(y_{i+1} - 2y_i + y_{i-1}) = 1 - x_i.$$

Записуємо систему:

$$\begin{array}{l} i=1 \\ i=2 \\ i=3 \end{array} \begin{cases} 8y_2 - 16y_1 + 8y_0 = 0,75 \\ 8y_3 - 16y_2 + y_1 = 0,5 \\ 8y_4 - 16y_3 + y_2 = 0,25. \end{cases}$$

Розглянемо умову $y'(0) = 0$. Використавши праві різниці $\frac{y_1 - y_0}{h} = 0$; $y_1 = y_0 = 0$, дістанемо результат, який не відповідає фізичному змісту задачі. Тому, щоб записати в різницевій формі рівняння $y'(0) = 0$, скористаємось виразом для правих різниць, записаних з подвійною точністю:

$$y_0' = \frac{1}{2h}(-3y_0 + 4y_1 - y_2) = 0; \quad y_2 = 4y_1.$$

Розв"яжемо систему алгебраїчних рівнянь четвертого порядку з чотирма невідомими: $y_0 = 0, y_1 = 0{,}0469, y_2 = 0{,}1875, y_3 = 0{,}4316, y_4 = 0{,}8715.$

Лінію прогину бруса показано на рис. 2.17.

Приклад 2.19. Знайти розв"язок крайової задачі прогину балки на пружній основі /рис. 2.18/:

$$y^{IV} + \frac{k}{EJ}\,y = \frac{q}{EJ}, \qquad /2.97/$$

де $k$ – коефіцієнт "постелі" /вважаємо $k = const$ /; $EJ$ – прогинна жорсткість балки /вважаємо $EJ = const$ /, при крайових умовах

$$y(0) = y'(0) = y(\ell) = 0; \; y''(\ell) = 0. \qquad /2.98/$$

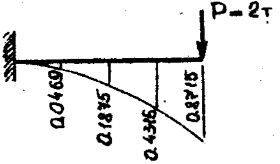


Рис. 2.17. Лінія прогину бруса

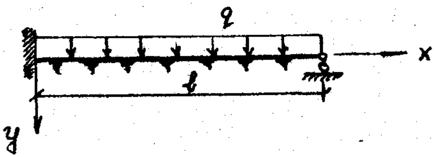


Рис. 2.18. Розрахункова схема балки на пружній основі

Припускаємо основу Вінклера

$$p(x) = k(x)\,y(x). \qquad /2.99/$$

Розв"язання. Розглянувши систему диференціальних рівнянь рівноваги вигнутої балки

$$\begin{cases} M''(x) + q(x) - k(x)y(x) = 0 \\ y''(x) + M(x)/EJ(x) = 0, \end{cases} \qquad /2.100/$$

дістанемо диференціальне рівняння четвертого порядку виду /2.97/.

Довжину балки $\ell$ розіб"ємо на $n$ рівних частин точками

$$x_i = 0 + ih \; (i = 0, 1, 2, \ldots, n)$$

де $h = \frac{\ell - 0}{n}$ /рис. 2.19/.

Похідні $y', y'', y''', y^{IV}$ в кожній серединній точці $x_1, x_2, \ldots, x_{n-1}$ можуть бути апроксимовані певними різницевими аналогами. Якщо скористатися центрально-різницевими відношеннями /похибка порядку $h^2$/ /див. рис. 2.13, а/ можна записати такі залежності:

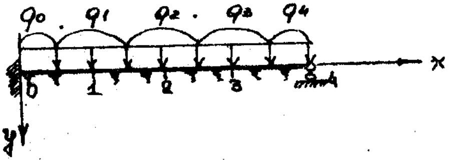


Рис. 2.19. Розбиття балки на ділянку і збирання навантаження

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2h}; \qquad /2.101\text{а}/$$

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}; \qquad /2.101\text{б}/$$

$$y_i''' = \frac{-y_{i-2} + 2y_{i-1} - 2y_{i+1} + y_{i+2}}{2h^3}; \qquad /2.101\text{в}/$$

$$y_i^{IV} = \frac{y_{i-2} - 4y_{i-1} + 6y_i - 4y_i + y_{i+2}}{h^4}, \qquad /2.101\text{г}/$$

а згідно з центральними різницевими відношеннями /похибка порядку $h^4$/ /див. рис. 2.13, б/ – такі залежності:

$$y_i' = \frac{y_{i-2} - 8y_{i-1} + 8y_{i+1} - y_{i+2}}{12h}; \qquad /2.102\text{а}/$$

$$y_i'' = \frac{-y_{i-2} + 16y_{i-1} - 30y_i + 16y_{i+1} - y_{i-2}}{12h^2}; \qquad /2.102\text{б}/$$

$$y_i''' = \frac{y_{i-3} - 8y_{i-2} + 13y_{i-1} - 13y_i + 8y_{i+2} - y_{i+3}}{8h^3}; \qquad /2.102\text{в}/$$

$$y_i^{IV} = \frac{-y_{i-3} + 12y_{i-2} - 39y_{i-1} + 56y_i - 39y_{i+1} + 12y_{i+2} - y_{i+3}}{6h^4}. \qquad /2.102\text{г}/$$

Через центральні різниці похідні обчислюються на порядок точніше, ніж через ліві або праві різниці.

У кожній серединній точці $x_1, x_2, \ldots, x_{n-1}$ замінимо похідні їх різницевими аналогами, скориставшись центральними різницевими відношеннями:

$$\begin{array}{ll} i=1 & \left\{ \begin{array}{l} \frac{1}{h^4}(y_{-1}-4y_0+6y_1-4y_2+y_3)+\frac{k}{EJ}y_1=\frac{q}{EJ}; \\ \frac{1}{h^4}(y_0-4y_1+6y_2-4y_3+y_4)+\frac{k}{EJ}y_2=\frac{q}{EJ}; \\ \frac{1}{h^4}(y_1-4y_2+6y_3-4y_4+y_5)+\frac{k}{EJ}y_3=\frac{q}{EJ}. \end{array} \right. \\ i=2 & \\ i=3 & \end{array} \qquad /2.I03/$$

Додаткові невідомі прогини фіктивних законтурних вузлів $(y_{-1}; y_5)$, які з"явилися в рівнянні для передконтурних вузлів / $i$ = I, 3/, легко знайти з граничних умов на кінцях балок, використавши рівняння /2.I0I/ та основні диференціальні залежності прогину балок:

$$\frac{d^2y}{dx^2}=-\frac{M}{EJ}; \qquad /2.I04а/$$

$$\frac{d^3y}{dx^3}=-\frac{Q}{EJ}; \qquad /2.I04б/$$

$$\frac{d^4y}{dx^4}=-\frac{q}{EJ}; \qquad /2.I04в/$$

$$\frac{dQ}{dx}=q; \qquad /2.I04г/$$

$$\frac{d^2M}{dx^2}=q. \qquad /2.I04д/$$

Розглянемо основні види граничних умов на кінцях балки.

I. Кінець балки вільно лежить на пружній основі /рис. 2.20/:

$$M_i=0; \quad Q_i=0.$$

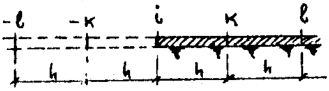


Рис. 2.20. Граничні умови на вільному кінці балки

Використовуючи /2.I0Iб/ та /2.I04а/, /2.I0Iв/ та /2.I04б/, дістаємо:

$$y_{-k}=2y_i-y_k; \qquad /2.I05/$$

$$y_{-\ell}=4y_i-4y_k+y_\ell. \qquad /2.I06/$$

2. Кінець балки шарнірно спирається на нерухому зосереджену опору /рис. 2.21/:

$$M_i = 0; \; y_i = 0.$$

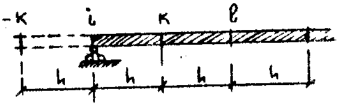


Рис. 2.21. Граничні умови для шарнірно опертого кінця балки

Тоді з /2.101б/ і /2.104а/ маємо:

$$y_{-k} = -y_k. \qquad /2.107/$$

3. Кінці балки жорстко закріплені і не допускають переміщень у вертикальному напрямі /рис. 2.22/:

$$y_i' = \theta_i = 0; \; y_i = 0.$$

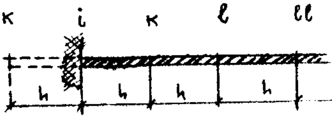


Рис. 2.22. Граничні умови жорстко закріпленого кінця балки

Використавши /2.101а/ та рівність $y' = \varphi$, де $\varphi$ – кут повороту, дістанемо

$$y_{-k} = y_k. \qquad /2.108/$$

Для розглядуваного випадку

$$y_{-1} = y_1; \; y_5 = -y_3.$$

Розглянемо похідні на кінцях інтервала.

На лівому його кінці значення похідних беремо за правими різницями /з подвійною точністю/ або різницями, взятими вперед:

$$y_0' = \frac{-3y_0 + 4y_1 - y_2}{2h}; \qquad /2.109а/$$

$$y_0'' = \frac{2y_0 - 5y_1 + 4y_2 - y_3}{h^2}; \qquad /2.109б/$$

$$y_0''' = \frac{-5y_0 + 18y_1 - 24y_2 + 14y_3 - 3y_4}{2h^3}; \qquad /2.109в/$$

$$y_0^{IV} = \frac{3y_0 - 14y_1 + 26y_2 - 24y_3 + 11y_4 - 2y_5}{h^4}. \qquad /2.109г/$$

На правому кінці інтервала значення похідних беремо за лівими різницями з подвійною точністю /або різницями, взятими назад/:

$$y_n' = \frac{y_{n-2} - 4y_{n-1} + 3y_n}{2h}; \qquad /2.110а/$$

$$y''_n = \frac{y_{n-3}+4y_{n-2}-5y_{n-1}+2y_n}{h^2}; \qquad /2.110б/$$

$$y'''_n = \frac{3y_{n-4}-14y_{n-3}+24y_{n-2}-18y_{n-1}+5y_n}{2h^3}; \qquad /2.110в/$$

$$y^{IV}_n = \frac{-2y_{n-5}+11y_{n-4}-24y_{n-3}+26y_{n-2}-14y_{n-1}+3y_n}{h^4}. \qquad /2.110г/$$

Згідно з граничними умовами /2.98/ і з урахуванням виразів /2.109а/ і /2.110б/ маємо:

$$y'_0 = \frac{-3y_0+4y_1-y_2}{2h}=0; \quad y''_5=\frac{y_1+4y_2-5y_3+2y_4}{h^2}=0.$$

Разом з рівняннями /2.103/ маємо систему $n+1$ лінійних алгебраїчних рівнянь з $n+1$ змінною. Розв"язати її можна будь-яким з розглянутих раніше методів.

## Розв"язування нелінійних та трансцендентних диференціальних рівнянь методом скінченних різниць

За допомогою методу скінченних різниць /МСР/ легко можна розв"язати будь-яке диференціальне нелінійне рівняння. Хід розв"язування аналогічний наведеній вище схемі. Область визначення незалежної змінної розбиваємо зі сталим кроком $h$ на $n$ частин. Розв"язок шукаємо в точках $x_i = x_0 + ih$, $i = 1,2,\dots,n$. Похідні в диференціальному рівнянні замінюємо їх скінченно-різницевими аналогами /центральними, правими або лівими/.

Записавши для кожної точки відповідне рівняння, дістаємо систему лінійних рівнянь. Її розв"язок буде розв"язком диференціального рівняння в окремих точках /фіксованих точках області визначення/.

Приклад 2.20. Розв"язати рівняння $y''=\sin x + y + \ln x$, якщо граничні умови такі: $y_0 = 1;\ y_4 = 2;\ h = 0,25$.

Записуємо скінченно-різницевий аналог даного диференціального рівняння:

$$i=1,2,3 \qquad \frac{y_{i-1}-2y_i+y_{i+1}}{h^2}=\sin x_i + y_i + \ln x;$$

$$i=1 \quad y_2 - 2y_1 + y_0 = 0{,}25^2(\sin 1{,}25 + y_1 + \ln 1{,}25);$$

$$-2{,}0625y_1 + y_2 = -0{,}927;$$

$$i=2 \quad y_3 - 2y_2 + y_1 = 0{,}25^2(\sin 1{,}5 + y_2 + \ln 1{,}5);$$

$$y_1 - 2{,}0625y_2 + y_3 = +0{,}0876;$$

$$i=3 \quad y_4 - 2y_3 + y_2 = 0{,}25^2(\sin 1{,}75 + y_3 + \ln 1{,}75);$$

$$y_2 - 2{,}0625y_3 = -1{,}9035$$

Розв"язуємо систему лінійних рівнянь:

$$\begin{cases} -2{,}0625y_1 + y_2 = -0{,}927 \\ y_1 - 2{,}0625y_2 + y_3 = 0{,}0876 \\ y_2 - 2{,}0625y_3 = -1{,}9035 \end{cases} \quad \begin{aligned} y_1 &= 0{,}2783; \\ y_2 &= 0{,}3531; \\ y_3 &= 1{,}0941. \end{aligned}$$

## 2.6. Розв"язування диференціальних рівнянь в частинних похідних

Інженеру часто доводиться розв"язувати задачі, в яких шукана величина залежить від кількох змінних. У цьому разі розв"язувані рівняння містять частинні похідні і називаються диференціальними рівняннями в частинних похідних.

Більшість з таких рівнянь не мають аналітичного розв"язку, і їх доводиться розв"язувати чисельними методами.

Якщо для розв"язування звичайних диференціальних рівнянь розроблено багато різних методів, то для розв"язування диференціальних рівнянь в частинних похідних /через їх складність/ застосовуються такі три методи: скінченних різниць, скінченних елементів, граничних елементів.

### Типи диференціальних рівнянь в частинних похідних

Такі рівняння класифікуються залежно або від їх математичної природи /еліптичні, параболічні тощо/, або від фізичної суті розв"язуваної за їх допомогою задачі /рівняння дифузії, хвильове, теплопровідності тощо/.

З математичної точки зору диференціальні рівняння другого порядку в частинних похідних з двома незалежними змінними

$$A(x,y)\frac{\partial^2 u}{\partial x^2}+B(x,y)\frac{\partial^2 u}{\partial x\partial y}+C(x,y)\frac{\partial u}{\partial x}+E(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y})=0, \qquad /2.III/$$

класифікуються за характером функцій $A$, $B$ і $C$. Ці функції залежать від змінних $x$ та $y$.

Рівняння називається еліптичним, якщо $B^2-4AC<0$, параболічним, якщо $B^2-4AC=0$, гіперболічним, якщо $B^2-4AC>0$.

Залежність функцій $A, B, C$ від $x$ і $y$ ускладнює задачу, оскільки тип рівняння при переході від однієї частини розглядуваної області до другої може змінитися.

Еліптичні рівняння описують сталі /стаціонарні/ процеси. Задача ставиться в замкненій області і в кожній точці межі цієї області задаються граничні умови.

Параболічні та гіперболічні рівняння описують еволюційні процеси /процеси "розповсюдження"/. У таких задачах на одній частині межі ставляться граничні умови, на другій - початкові.

У практичній діяльності інженера найчастіше зустрічаються рівняння в частинних похідних, подані в табл. 2.II.

Таблиця 2.II

Типи диференціальних рівнянь в частинних похідних

| Рівняння | Математична форма | Приклади задач |
|---|---|---|
| Лапласа | $\Delta f=0$ | Сталий потік рідини<br>Стаціонарні теплові поля |
| Пуассона | $\Delta f=-k$ | Теплопередача з внутрішніми джерелами теплоти |
| Дифузії | $\Delta f=\frac{1}{h^2}\frac{\partial f}{\partial t}$ | Нестаціонарна теплопровідність |
| Хвильове | $\Delta f=\frac{1}{c^2}\frac{\partial^2 f}{\partial t^2}$ | Розповсюдження звукових хвиль |
| Бігармонічне | $\Delta^2 f=F(x,y)$ | Деформації пластин |

Символом $\Delta$ позначається оператор Лапласа, причому у випадку однієї незалежної змінної $x$

$$\Delta = \frac{\partial^2}{\partial x^2}, \qquad /2.112/$$

у випадку двох незалежних змінних $x$ і $y$

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \qquad /2.113/$$

Оператор $\Delta^2$ називається бігармонічним і у випадку двох незалежних змінних записується у вигляді

$$\Delta^2 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial^2 x\, \partial^2 y} + \frac{\partial^4}{\partial y^4}. \qquad /2.114/$$

## МСР чисельного розв"язування диференціальних рівнянь в частинних похідних

Важливою областю застосування МСР є розв"язування диференціальних рівнянь в частинних похідних. Прикладами таких рівнянь можуть бути рівняння, які застосовуються для розв"язування задач, що описують напружено-деформований стан плит:

$$\Delta^2 f = \frac{\partial^4 f}{\partial x^4} + 2\frac{\partial^4 f}{\partial x^2 \partial y^2} + \frac{\partial^4 f}{\partial y^4} = F(x, y). \qquad /2.115/$$

Пружні деформації пластин під дією сталих сил розраховуються за допомогою рівняння Пуассона /перша крайова задача Діріхле/:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = F(x, y). \qquad /2.116/$$

Ця задача формулюється таким чином: знайти функцію $f = f(x, y)$, яка задовольняє усередині деякої області $G$ рівняння типу /2.115 – 2.116/, а на межі $\Gamma$ – умову $f|_{\Gamma} = \varphi(x,y)$, де $\varphi(x, y)$ – задана неперервна функція.

Такі задачі розв"язуються методом сіток або МСР. В його основі лежить ідея апроксимації похідних скінченно-різницевими відношеннями. Заміна рівнянь в частинних похідних системою алгебраїчних рівнянь виконується так само, як і для звичайних диференціальних рівнянь /див. підрозд. 2.5/. Це пояснюється тим, що частинні і звичайні похідні обчислюються завдяки одним і тим самим граничним переходам /при розв"язуванні задачі в частинних похідних всі незалежні змінні, крім однієї, слід вважати сталими/.

Апроксимація похідних здійснюється в три етапи.

І-й етап. В області, де відшукують розв"язок, вводять рівномірну сітку вузлових точок, які відповідають характеру задачі та граничним умовам. Найчастіше в розрахунках застосовується декартова система координат, а в ній прямокутна координатна сітка вузлів /рис. 2.23,а/. У косокутній системі координат застосовується координатна сітка з паралелограмів /рис. 2.23,б/. У деяких задачах вживається полярна система координат і відповідна їй полярна сітка /мал. 2.23,в/.

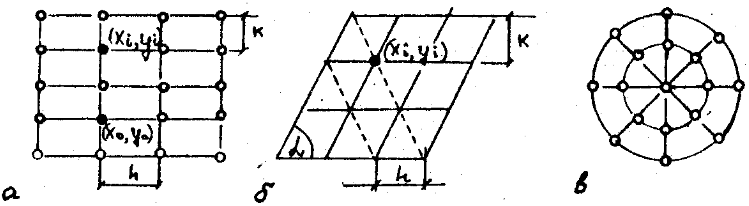


Рис. 2.23. Форми сіток, які використовуються в МСР

2-й етап. Розв"язуване рівняння в частинних похідних записують у системі координат, що відповідає задачі, і, подаючи похідні скінченно-різницевими відношеннями, зводять це рівняння до системи лінійних алгебраїчних рівнянь. Останні виражають функціональний зв"язок між сусідніми вузлами сітки.

3-й етап. Різницеві рівняння записують для всіх вузлів сітки і дістають систему $n$ рівнянь з $n$ змінними. Розв"язання цієї системи виконують одним з відомих методів, наприклад методом Гаусса, Зейделя тощо.

Розглянемо випадок двох незалежних змінних. Нехай у площині $xOy$ є область $G$ з межею $Г$ /рис. 2.24/. Побудуємо на площині дві сім"ї паралельних прямих:

$$x = x_0 + ih \quad (i = 0, \pm 1, \pm 2, \dots);$$

$$y = y_0 + kl \quad (k = 0, \pm 1, \pm 2, \dots).$$

Точки перетину цих прямих назвемо вузлами, точки $B$ і $C$ – граничними вузлами, точку $A$ – внутрішнім вузлом. У кожному внутрішньому вузлі замінимо частинні похідні різницевими відношеннями, які отримані або в результаті розвинення функції в ряд Тейлора, або диференцюванням інтерполяційного многочлена.

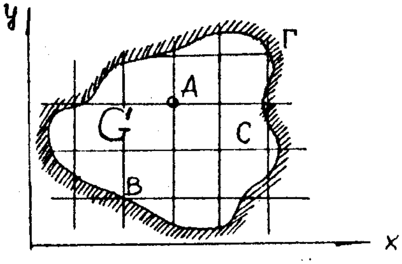


Рис. 2.24. Нанесення сітки на розглядувану область $G$

Згідно з позначеннями на рис. 2.25 запишемо скінченно-різницеві відношення для частинних похідних в порядку їх зростання в напрямі осі $x$ :

$$\frac{\partial u}{\partial x}=\frac{u_r-u_l}{2h}; \qquad /2.117а/$$

$$\frac{\partial^2 u}{\partial x^2}=\frac{u_r-2u_i+u_l}{h^2}; \qquad /2.117б/$$

$$\frac{\partial^3 u}{\partial x^3}=\frac{u_{rr}-2u_r+2u_l-u_{ll}}{2h^3}; \qquad /2.117в/$$

$$\frac{\partial^4 u}{\partial x^4}=\frac{u_{rr}-4u_r+2u_l-u_{ll}}{4^4}, \qquad /2.117г/$$

у напрямі осі $y$ :

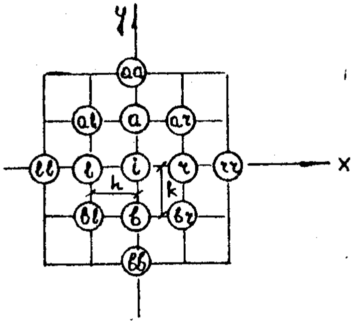


Рис. 2.25. Індексні позначення вузлових точок в прямокутній сітці

$$\frac{\partial u}{\partial y}=\frac{u_a-u_l}{2k}; \qquad /2.118а/$$

$$\frac{\partial^2 u}{\partial y^2}=\frac{u_a-2u_i+u_b}{k^2}; \qquad /2.118б/$$

$$\frac{\partial^3 u}{\partial y^3}=\frac{u_{aa}-2u_a+2u_b-u_{bb}}{2k^3}; \qquad /2.118в/$$

$$\frac{\partial^4 u}{\partial y^4}=\frac{u_{aa}-4u_a+6u_i-4u_b+u_{bb}}{k^4}. \qquad /2.118г/$$

Вираз для мішаної похідної:

$$\frac{\partial^2 u}{\partial x^2}=\frac{\partial u\,\partial u}{\partial x\,\partial y}=\frac{(u_{ar}-u_{ab})-(u_{br}+u_{bb})}{4hk}=D_{xy}, \qquad /2.119/$$

звідки

$$\frac{\partial^4 u}{\partial x^2 \partial y^2} = \frac{\partial^2 u \, \partial^2 u}{\partial x^2 \partial y^2} = \frac{(u_{az}+u_{ab}+u_{bz}+u_{bb})-2(u_a+u_b+u_z+u_b)+4u_i}{h^2 k^2} = D^2_{xy}. \quad /2.120/$$

Оператор Лапласа

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = D^2_x + D^2_y$$

у прямокутній сітці зі сторонами клітки $k$ і $h$ набирає згідно з /2.117/ – /2.10/ такого вигляду:

$$h^2 k^2 \Delta u_i = k^2 (u_z - 2u_i + u_\ell) + h^2 (u_a - 2u_i + u_b) + O(h^2). \quad /2.121/$$

Якщо сітка має форму квадрата, гармонічний оператор $\Delta$ можна спростити так:

$$h^2 \Delta u_i = u_a + u_b + u_z + u_\ell - 4u_i + O(h^2). \quad /2.122/$$

Вігармонічний оператор

$$\Delta^2 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^2}$$

в квадратній сітці набирає вигляду [10] :

$$\begin{aligned} h^4 \Delta^2 u_i &= h^4 \Delta(\Delta u_i) = h^4 (\Delta u_a + \Delta u_b + \Delta u_z + \Delta u_\ell - 4\Delta u_i) = \\ &= (u_{aa} + u_{bb} + u_{zz} + u_{\ell\ell}) + 2(u_{ab} + u_{az} + u_{bz} + u_{b\ell}) - \quad /2.123/ \\ &- 8(u_a + u_b + u_z + u_\ell) + 20u_i + O(h^2). \end{aligned}$$

Оператори $D_{xy}, \Delta, \Delta^2$ якщо сітка квадратна, можна записати у вигляді схем, зображених на рис. 2.26. Записавши ці самі оператори через центрально-різницеві співвідношення /рис. 2.27/, дістанемо точніші результати при розрахунках. Тоді похибка $O(h^4)$ наближатиметься до нуля пропорційно $h^4$.

Отже, розв"язками рівнянь /2.115/, /2.116/ будуть функції $u(x,y)$, які задовольняють рівняння /2.121/ або /2.123/ усередині області $G$ та граничні умови на межі області. Законтурні точки, згідно з граничними умовами, враховуються за співвідношеннями /2.105/ – /2.108/.

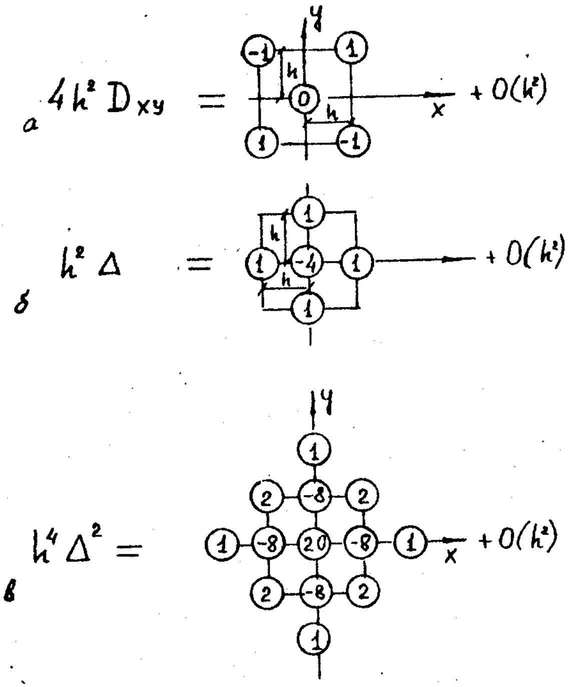


Рис. 2.26. Коефіцієнти МСР при вузлових точках для операторів $D_{xy}$ /а/; $\Delta$ /б/; $\Delta^2$ /в/

**Приклад 2.21.** Розглянемо крайову задачу про вигин квадратної пластини зі стороною $a$, жорстко закріпленої по краях і навантаженої розподіленим навантаженням $q$ /рис. 2.28/. Задача описується рівнянням

$$\Delta^2 W = q/D, \qquad /2.124/$$

при $W = 0$ /на межі/ $\partial W/\partial \nu = 0$, де $q$ – навантаження на одиницю площі; $D=Eh^3/12(1-\mu^2)$ – циліндрична жорсткість; $E, \mu$ – відповідно модуль пружності та коефіцієнт Пуассона матеріалу пластини; $h$ – товщина пластини; $\nu$ – напрям нормалі до межі /вісь $x$ або $y$ /.

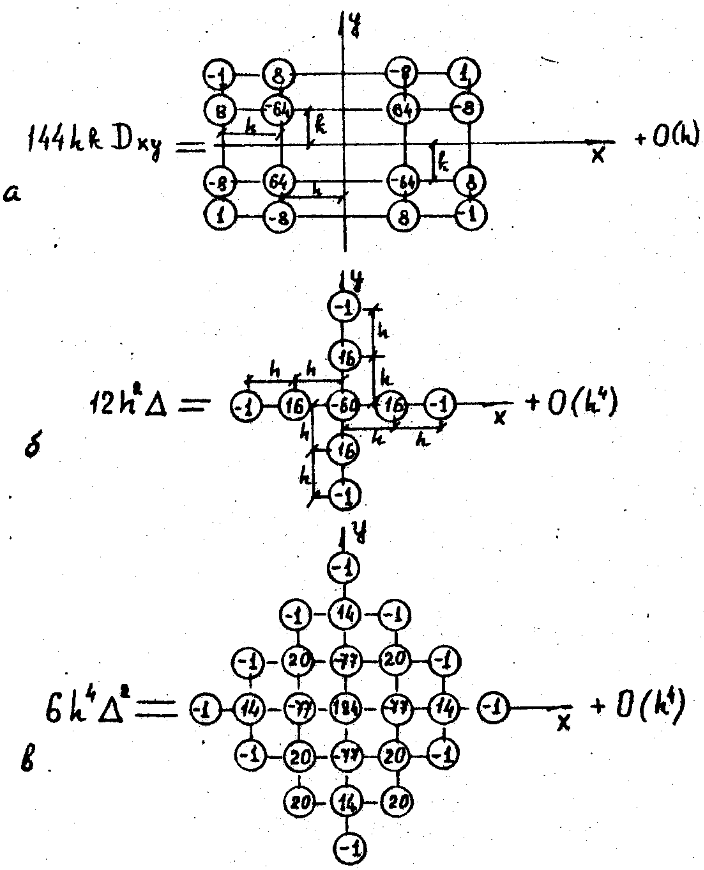


Рис. 2.27. Коефіцієнти МСР при вузлових точках, записані з подвійною точністю через центральні різниці для операторів $D_{xy}$ /а/; $\Delta$ /б/; $\Delta^2$ /в/

**Розв"язання.** Зведемо рівняння /2.I24/ до безрозмірної форми заміною $x = \xi a,\ y = \eta a$:

$$\frac{\partial^4 W}{\partial x^4} + 2\frac{\partial^4 W}{\partial x^2 \partial y^2} + \frac{\partial^4 W}{\partial y^4} = \frac{q}{D};$$

$$\frac{\partial^4 W}{a^4 \partial \xi^4} + 2\frac{\partial^4 W}{a^4 \partial \xi^2 \partial \eta^2} + \frac{\partial^4 W}{a^4 \partial \eta^4} = \frac{q}{D}.$$

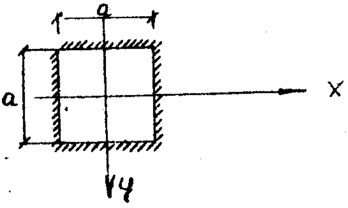


Рис. 2.28. Жорстко закріплена по контуру пластина

Домножимо ліву і праву частини рівняння на $\frac{q}{D}$ і введемо позначення

$$z(\xi,\eta) = \frac{W(x,y) D}{a^4 q}. \qquad /2.I25/$$

Тоді

$$\frac{\partial^4 z}{\partial \xi^4} + 2\frac{\partial^4 z}{\partial \xi^2 \partial \eta^2} + \frac{\partial^4 z}{\partial \eta^4} = 1;$$

$$\Delta^2 z = 1, \qquad /2.I26/$$

при $z = 0$ /на межі/ $\partial z / \partial \nu = 0$, /$\Delta^2$ береться відносно $\xi$ та $\eta$ /.

Обчисливши з /2.I26/ функцію $z(\xi,\eta)$, прогин пластини знаходимо з /2.I25/:

$$W(x,y) = \frac{q a^4}{D} z(\xi,\eta). \qquad /2.I27/$$

Чисельний розв"язок /2.I22/ дістанемо, замінивши $\Delta^2$ різницевим аналогом /2.I23/ для квадратної сітки з кроком $h = n$.

Граничні умови дають $z = 0$ на межі. Значення $z$ в перших законтурних точках для жорсткого закріплення згідно з /2.I08/: $y_{-k} = y_k$.

Нехай $n = 2$ /рис. 2.29/.

Рівняння /2.I26/ замінимо на /2.I23/ – його скінченно-різницевий аналог:

$$(z_0 + z_0 + z_0 + z_0) + 2\cdot 0 - 8\cdot 0 + 20 z_0 = \frac{1}{h^4},$$

звідки

$$z_0 \Big|_{n=2} = \frac{1}{384}.$$

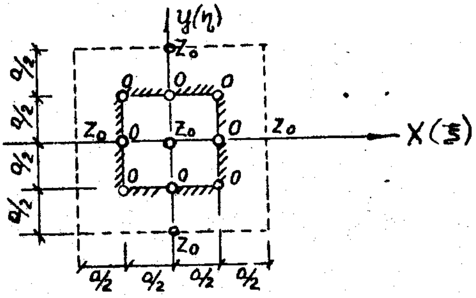


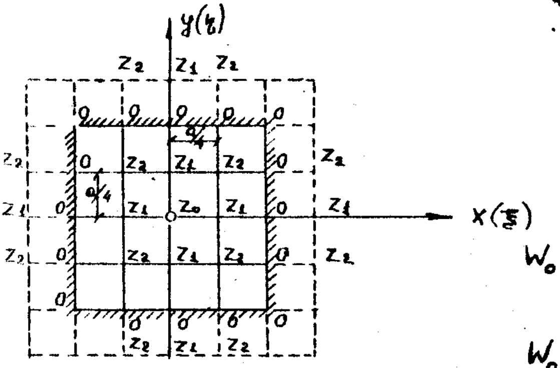
Рис. 2.29. Розбиття пластини на дві ділянки вздовж осей $x$ і $y$

Нехай $n = 4$ /рис. 2.30/. З рівняння /2.126/ для точок на рис. 2.30 маємо /індекс точки відповідає індексу функції $z_i$ /:

$$\begin{matrix} (z_0) \\ (z_1) \\ (z_2) \end{matrix} \begin{cases} 20\,z_0 - 32\,z_1 + 8\,z_2 = \dfrac{1}{4^4}; \\ -8\,z_0 + 26\,z_1 - 16\,z_2 = \dfrac{1}{4^4}; \\ 2\,z_0 - 16\,z_1 + 24\,z_2 = \dfrac{1}{4^4}. \end{cases}$$

Розв"язавши цю систему, дістанемо: $z_0\big|_4 = \dfrac{1}{555}$; $z_1\big|_4 = \dfrac{1}{828}$;

$z_2\big|_4 = \dfrac{1}{1225}$.



При $n = 8$ $z_0\big|_8 = 1/699$.

Прогин пластини в центрі при $\mu = 0{,}3$ і $n = 2,4,8$ згідно з /2.127/:

$$W_0\Big|_2 = \frac{12(1-0{,}3^2)qa^4}{384\,Eh^3} = 0{,}0284\,\frac{qa^4}{Eh^3};$$

$$W_0\Big|_4 = \frac{12(1-0{,}3^2)qa^4}{384\,Eh^3} = 0{,}0197\,\frac{qa^4}{Eh^3};$$

Рис. 2.30. Розбиття пластини на чотири ділянки вздовж кожної осі

$$W_0\Big|_8 = \frac{12(1-0{,}3^2)qa^4}{384\,Eh^3} = 0{,}0156\,\frac{qa^4}{Eh^3}.$$

Чим менший крок $h$, тим точніший розв"язок рівняння /2.124/.

## 2.7. Короткі відомості з варіаційного числення. Поняття функціоналу та необхідні умови екстремуму

Розглянемо деякий визначений інтеграл

$$I = \int_a^b F(x, y, y')\,dx \qquad /2.128/$$

з граничними умовами

$$y(a) = \alpha, \quad y(b) = \beta. \qquad /2.129/$$

Вираз типу /2.I28/ у варіаційному численні називають функціоналом /функція від функції/.

Пошук умов, при яких функціонали набувають стаціонарного значення, і становлять зміст задачі варіаційного числення. Говорять, що функція в точці має стаціонарне значення, якщо швидкість її зміни в цій точці дорівнює нулю /тобто дорівнюють нулю перші частинні похідні функції за всіма незалежними змінними/.

Нехай маємо деяку функцію $y(x)$, яка надає інтегралу /2.I28/ стаціонарного значення. Для того щоб показати, що справді маємо стаціонарне значення, розглянемо деяку функцію $\tilde{y}(x)$, відмінну від $y(x)$, але водночас нескінченно близьку до неї /рис. 2.3I/

$$\tilde{y}(x)=y(x)+\mathcal{E}\varphi(x), \qquad /2.I30/$$

де $\mathcal{E}$ – довільна як завгодно мала величина, $\varphi(x)$ – деяка функція, що задовольняє умови неперервності й диференційовності.

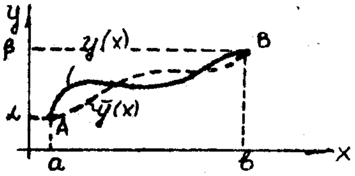


Рис. 2.3I. До поняття варіації функції

Різниця $\tilde{y}(x)-y(x)=\mathcal{E}\varphi(x)$ називається варіацією функції $y(x)$. Лагранж ввів для неї позначення $\delta y$. Дуже важливо, що величина $\delta y$ має так званий віртуальний /можливий/ характер, тобто не пов"язана з жодними умовами і може бути довільною. Якщо функцію $y(x)$ тлумачити як переміщення, то $\delta y$ називатиметься можливим переміщенням.

Більшість методів розв"язування варіаційних задач грунтуються на використанні необхідних умов екстремуму, якими звичайно бувають диференціальні рівняння /рівняння Ейлера/ і граничні умови. Отже варіаційна задача виявляється еквівалентною крайовій задачі для диференціального рівняння.

Диференціальне рівняння

$$\frac{\partial F}{\partial y}-\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial y'}\right)=0,\ x\in[a,b], \qquad /2.I3I/$$

виражає необхідну й достатню умову стаціонарності інтеграла $I$ при заданих граничних умовах. Тобто коли підінтегральна функція задовольняє рівняння /2.I3I/, тоді функціонал /2.I28/ має стаціонарне значення.

Рівняння /2.3I/ дістали незалежно Л.Ейлер та Ж.Лагранж. У варіаційному численні воно називається рівнянням Ейлера – Лагранжа і відіграє фундаментальну роль.

Як приклад можна навести варіаційну задачу Лагранжа про пошук мінімуму потенціальної енергії пружного стержня, еквівалентну диференціальному рівнянню рівноваги, записаному в переміщеннях з відповідними граничними умовами. Рівняння рівноваги вигнутого стержня /балки/ має вигляд /2.74/:

$$y'' = \pm \frac{M}{EJ} . \qquad /a/$$

Знак "+" відповідає випадку, коли вісь $y$ спрямована вгору. Тоді знаки момента $M$ та другої похідної $y''$ збігаються.

Функціоналом для рівняння /а/ при розв"язуванні цієї задачі буде вираз повної потенціальної енергії пружної балки. Розв"язок рівняння /а/ - знайдена пружна лінія балки $y(x)$ - забезпечить мінімум потенціальної енергії балки. Іншими словами, рівняння рівноваги являють собою рівняння Ейлера - Лагранжа для відповідного функціоналу. До них потрібно ще приєднати граничні умови, які можна дістати з виразу енергії.

Отже, повна потенціальна енергія системи дає повну інформацію про систему, оскільки дозволяє дістати не тільки рівняння рівноваги, але й конкретні граничні умови задачі.

## Суть варіаційних методів розв"язування диференціальних рівнянь. Форма апроксимуючої функції

Суть варіаційних методів розв"язування диференціальних рівнянь полягає в тому, що функція, яка задовольняє диференціальне рівняння при заданих граничних умовах, замінюється наближеним аналітичним виразом. Цей вираз добирається так, щоб він найкраще апроксимував дану функцію, тобто щоб відхилення від істинного значення функції було найменшим.

При розв"язуванні цієї задачі звичайно виникають два питання:

1/ про вибір форми апроксимуючої функції;

2/ про засіб наближення цієї функції.

Кожне з цих питань потрібно досліджувати окремо, оскільки від форми апроксимуючої функції та від засобу наближення залежить швидкість процесу збіжності результатів.

Практичне застосування варіаційних методів показує, що найзручнішою формою вираження апроксимуючої функції для будь-якого числа незалежних змінних є подання функції у вигляді ряду

$$W_n = \sum_1^n a_i u_i \quad (i = 1,2,3,\dots,n), \qquad /2.132/$$

де $a_i$ - невизначені сталі параметри, які варіюються згідно з прийнятим способом наближення; $u_i$ - "підходящі" функції, які в сукупності найкраще зображують досліджувану функцію.

Розглянемо перше питання. Припустимо для визначеності, що йдеться про функцію від однієї незалежної змінної в проміжку $[a, b]$ для якої відоме диференціальне рівняння

$$F(x, W, W', \dots, W^{IV}) = 0 \qquad /2.I33/$$

та граничні умови

$$W(a) = W_a;\ W(b) = W_b;\ W'(a) = W'_a;\ W'(b) = W'_b,$$

або інакше,

$$W(a) - W_a = 0;\quad W(b) - W_b = 0;$$
$$W'(a) - W'_a = 0;\quad W'(b) - W'_b = 0.$$

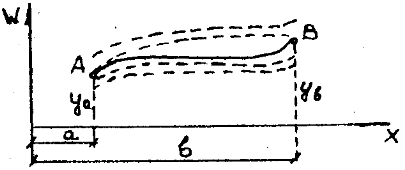


Рис. 2.32. Точне і наближене розв"язування диференціального рівняння /2.I33/

Нехай $W = f(x)$ - точний розв"язок цього рівняння /рис. 2,32, суцільна лінія $AB$ /, а $W_n = \sum_{i=1}^{n} a_i u_i$ - довільний з наближених розв"язків /пунктир на рис. 2.32/. Яку з ліній слід узяти за наближений розв"язок?

<u>I-й спосіб</u>. Вибираємо функції $u_i$ так, щоб кожна з них окремо задовольняла частину граничних умов. Диференціальне рівняння при цьому не задовольняється. Параметри $a_i$ визначаємо з умови, що весь ряд в цілому найкраще апроксимує функцію як всередині, так і на кінцях проміжку.

<u>2-й спосіб</u>. Вибираємо функції $u_i$ так, щоб кожна з них окремо задовольняла всі граничні умови. Диференціальне рівняння при цьому не задовольняється. Параметри $a_i$ знаходимо з умови, що весь ряд в цілому найкраще апроксимує функцію всередині проміжку.

<u>3-й спосіб</u>. Вибираємо функції $u_i$ так, щоб кожна з них окремо задовольняла диференціальне рівняння, тобто була його частинним інтегралом. Граничні умови при цьому не задовільняються. Параметри $a_i$ знаходимо з умови, що весь ряд в цілому найкраще апроксимує функцію на кінцях проміжку.

З указаних способів побудови ряду перший найпростіший, оскільки він накладає на функцію найменше число обмежень. Проте умова для знаходження параметрів при цьому виходить менш простою.

Наведені міркування для випадку однієї змінної поширюються на випадок двох незалежних змінних. Відмінність лише в тому, що тут проміжок замінюється областю площини, граничні точки - замкненим контуром цієї області, а інтегральна лінія - інтегральною поверхнею.

Завжди весь ряд в цілому повинен якомога повніше відповідати очікуваному характеру зміни функції.

Зауважимо, що при розв"язуванні практичних задач потрібні певні навички. Добре мати під руками довідкову таблицю так званих "підходящих" координатних функцій.

Так, наприклад, для вигину шарнірно опертої балки координатну функцію можна записати у вигляді

$$W_y = \sum_m \sin\frac{m\pi x}{\ell}, \quad m = 1,2,3,\ldots .$$

Для більшості практичних випадків рівняння Ейлера - Лагранжа не можливо розв"язати точно і доводиться використовувати чисельні методи. У таких випадках докладають зусиль, щоб уникнути розв"язування диференціальних рівнянь, а спробувати побудувати мінімізуючу послідовність.

## Метод Рітца

Метод Рітца - один з чисельних методів розв"язування диференціального рівняння Ейлера. За його допомогою варіаційна задача зводиться до задачі пошуку екстремуму функції. Основна ідея полягає ось у чому. Нехай $y_1, y_2, \ldots$ - повна система функцій в області визначення функціоналу $I(y)$, тобто кожна функція $y$ з цієї області може бути наближена з будь-яким ступенем точності лінійною комбінацією деяких функцій $u_i$, які називають координатними:

$$y_n(x) = \sum_1^n a_i u_i \qquad /2.134/$$

/ $n$ залежить від потрібної точності/.

Якщо цю лінійну комбінацію підставити у функціонал $I(y)$ то він буде функцією лише параметрів $a_i$ :

$$I\left(\sum_1^n a_i u_i\right) = f(a_1, a_2, \ldots, a_n). \qquad /2.135/$$

Необхідною умовою того, що ця функція набуває екстремального значення відносно параметрів $a_1, a_2, \ldots, a_n$, є виконання систем співвідношень

$$f'(a_1)=0;\ f'(a_2)=0; \ldots f'(a_n)=0. \qquad /2.136/$$

З цієї системи нелінійних рівнянь знаходяться параметри $a_i$. Їх ще називають ступенями вільності. Як правило, за фізичним змістом вони являють собою вузлові лінійні та кутові переміщення.

Ступінь вільності системи - це число незалежних геометричних параметрів, які повністю визначають можливі переміщення всіх її точок. Реальна пружна система завжди має нескінченне число ступенів вільності. Щоб чисельно розв"язати реальну задачу, деформований стан слід охарактеризувати переміщеннями скінченного числа точок, тобто число ступенів вільності вважається скінченним.

Координатні функції $u_i$ добираються так, щоб функція $y_n = \sum_1^n a_i u_i$ задовольняла граничні умови задачі. Так, координатні функції описують розподіл переміщень по області прогнутого стержня, коли одне з переміщень дорівнює одиниці, а інші дорівнюють нулю. Вирази для координатних функцій, застосовних в основних випадках розрахунків, вміщені в табл. 2.12.

Приклад 2.22. Знайти екстремум функціоналу

$$I(y) = \int_0^1 (y'^2 + y^2 + 2xy)dx, \quad y(0)=y(1)=0.$$

Розв"язання. Припустимо, що використовуються координатні функції

$$u_1(x) = x^2 - x,\ u_2(x) = x^3 - x^2, \ldots, u_n(x) = x^{n+1} - x^n.$$

Нехай $n = 2$, тобто

$$y_2(x) = a_1(x^2-x) + a_2(x^3-x^2),\ y_2(0) = y_2(1) = 0.$$

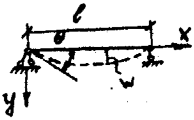
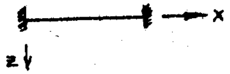
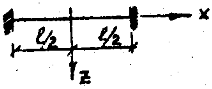
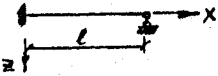
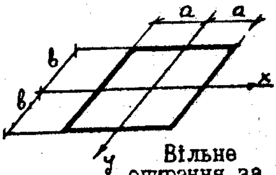
Тоді

$$I(y_2(x)) = \int_0^1 [a_1(2x-1) + a_2(3x^2-2x))^2 + (a_1(x^2-x) +$$
$$+ a_2(x^3-x^2))^2 + 2x(a_1(x^2-x)+a_2(x^3-x^2))]dx = f(a_1, a_2).$$

З умови $f(a_1)'=0$ і $f(a_2)'=0$ маємо:

$$\begin{cases} \frac{11}{15}a_1 + \frac{11}{30}a_2 = \frac{1}{6}; \\ \frac{11}{30}a_1 + \frac{2}{7}a_2 = \frac{1}{10}. \end{cases}$$

Таблиця 2.12

| № п/п | Граничні умови | Координатні функції |
|---|---|---|
| I |  | $W,Q=a_1x(l-x)+a_2x^2(l-x)+a_3x(l-x)^2+$ $+a_4x^2(l-x)^2+\dots$<br>/довільне навантаження/<br>$W,Q=a_1x(l-x)+a_2x^2(l-x)^2+a_3x^3(l-x)^3+\dots$<br>/симетричне навантаження/<br>$W=\sum\sin\frac{m\pi x}{l};\ (m=1,2,3,\dots)$ |
| 2 |  | $W,Q=a_1\sin\frac{\pi x}{l}+a_2\sin\frac{3\pi x}{l}+a_3\sin\frac{5\pi x}{l}+\dots$<br>/симетричне навантаження/<br>$W,Q=a_1\sin\frac{\pi x}{l}+a_2\sin\frac{2\pi x}{l}+a_3\sin\frac{3\pi x}{l}+\dots$<br>/довільне навантаження/ |
| 3 |  | $W_m=\sum\frac{1}{2}(1-\cos\frac{2m\pi x}{l})\ (m=1,3,5,\dots)$<br>$W_m=\frac{x}{l}(\frac{x}{l}-1)^2-\sum(-1)^m\frac{x^2}{b^2}(\frac{x}{a}-1)-\sum_m\frac{1}{m\pi}\sin\frac{m\pi x}{l}$ |
| 4 |  | $W_m=\sum_m(l^2-4x^2)^2x^m\ (m=0,1,2,3,\dots)$<br>$W_m=\sum_m[1-(-1)^m\cos\frac{2m\pi x}{l}]\ (m=1,3,5,\dots)$ |
| 5 |  | $W_m=\frac{x}{l}(\frac{x}{l}-1)(\frac{x}{2l}-1)-\sum_m\frac{1}{m\pi}\sin\frac{m\pi x}{l}$ |
| 6 |  | $W_m=\sum\frac{x}{2l}(\frac{x^2}{l^2}-1)(-1)^m-\sum_m\frac{1}{m\pi}\sin\frac{m\pi x}{l}$ |
| 7 |  Вільне опирання за контуром | $W=a_1\cos\frac{\pi x}{2a}\cos\frac{\pi y}{2b}+a_2\cos\frac{\pi x}{2a}\cos\frac{3\pi y}{2b}+$ $+a_3\cos\frac{3\pi x}{2a}\cos\frac{\pi y}{2b}+a_4\cos\frac{3\pi x}{2a}\cos\frac{3\pi y}{2b}+\dots$<br>/довільне навантаження/<br>$W=a_1(x^2-a^2)(y^2-b^2)+a_2(x^2-a^2)x^2(y^2-b^2)y^2+\dots$<br>/симетричне навантаження/ |

Звідси
$$a_1 = 69/473;$$
$$a_2 = 77/473;$$
$$y_2(x) = (77x^3 - 8x^2 - 69x)/473.$$

Отже, точний розв"язок
$$y(x) = \frac{e}{e^2-1}(e^x - e^{-x}) - x.$$

| $i$ | $y_2(2i)$ | Точний розв"язок $y(2i)$ |
|---|---|---|
| 1 | -0,0285 | -0,0287 |
| 2 | -0,0506 | -0,0505 |

Приклад 2.23. Крайова задача прогину балки.

Знайти лінію прогинів для балки, зображеної на рис. 2.33. Граничні умови:

$$W(-l/2)=0;\ W(l/2)=0;$$
$$W(-l/2)=0;\ W'(l/2)=0.$$

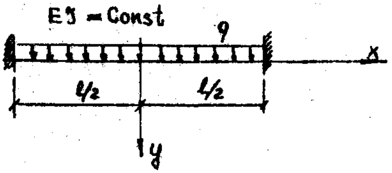


Рис. 2.33. Розрахункова схема навантаження на балку

Розв"язання. Розв"язати задачу у варіаційній постановці – означає знайти таку систему переміщень, яка мінімізує функціонал повної потенціальної енергії системи.

Цей функціонал для даної задачі
$$I(y) = \Pi = \frac{EJ}{2}\int_{-l/2}^{l/2}\left(\frac{d^2W_y}{dx^2}\right)^2 dx - \frac{1}{2}\int_{-l/2}^{l/2} q\,W_y\,dx. \qquad /2.137/$$

Перший доданок в /2.137/ являє собою потенціальну енергію пружних деформацій, що дорівнює роботі внутрішніх сил, які діють на балку:
$$A = \sum\int_0^s \frac{M^2 ds}{2EJ} + \sum\int_0^s \frac{N^2 ds}{2EF} + \mu\sum\int_0^s \frac{Q^2 ds}{2GF}. \qquad /2.138/$$

У даному разі другим і третім доданками в /2.138/ можна знехтувати, оскільки за модулем вони досить малі. Другий доданок у /2.137/ являє собою потенціал зовнішніх сил. Щоб розв"язати задачу, можна скористатися двома способами: 1/ мінімізувати функціонал; 2/ розв"язати диференціальне рівняння Ейлера – Лагранжа.

Наведемо розв"язання першим способом. Доберемо координатну функцію у вигляді:

$$u_n(x)=\sum_n(\ell^2-4x^2)^2x_n \quad (n=0,1,2,3,\ldots).$$

Нехай $n$ = 1, тоді

$$u_1(x)=(\ell^2-4x^2)^2$$

і переміщення $W_n=\sum^n u_i\,a_i$.

Координатна функція задовольняє такі граничні умови:

$$W_1(x)=a_1(\ell^2-4x^2)^2;$$

$$W_1(x)=a_1(\ell^4-8\ell^2x^2+16x^4);$$

$$W_1'(x)=-16a_1\ell^2x+64x^3a_1;$$

$$W_1'(x)\Big|_{-\ell/2}=+16a_1\ell^2\frac{\ell}{2}-64\frac{a_1\ell^3}{8}=8a_1\ell^3-8a_1\ell^3=0;$$

$$W_1'(x)\Big|_{\ell/2}=0;$$

$$W_1''(x)=-16a_1\ell^2+192a_1x^2=16[a_1(12x^2-\ell^2)];$$

$$W_1(x)\Big|_{x=-\ell/2}=a_1\left(\ell^2-4\frac{\ell^2}{4}\right)^2+a_2\left(\ell^2-4\frac{\ell^2}{4}\right)\left(-\frac{\ell}{2}\right)=0;$$

$$W_1(x)\Big|_{x=\ell/2}=0;$$

$$\Pi=\frac{EJ}{2}\int_{-\ell/2}^{\ell/2}[16a_1(12x^2-\ell^2)]^2dx-\frac{q}{2}\int_{-\ell/2}^{\ell/2}a_1(\ell^2-4x^2)^2dx=$$

$$=128\,EJ\int_{-\ell/2}^{\ell/2}a_1^2(12x^2-\ell^2)dx-q/2\int_{-\ell/2}^{\ell/2}a_1(\ell^4-8\ell^2x^2+16x^4)dx=$$

$$=128EJ\int_{-\ell/2}^{\ell/2}a_1^2(144x^4+24x^2\ell^2+\ell^4)dx-q/2\int_{-\ell/2}^{\ell/2}a_1(\ell^4-8\ell^2x^2+16x^4)dx=$$

$$=128\,EJ\Big|_{-\ell/2}^{\ell/2}a_1^2\left(144\frac{x^5}{5}-24\frac{\ell^2x^3}{3}+\ell^4x\right)-\frac{q}{2}\Big|_{-\ell/2}^{\ell/2}a_1\left(\ell^4x-8\ell^2\frac{x}{3}+16\frac{x^5}{6}\right)=$$

$$=128\,EJ\left[a_1^2\left(\frac{144}{5}\frac{\ell^5}{32}-\frac{8\ell^2\ell^3}{8}+\frac{\ell^4\ell}{2}\right)-a_1^2\left(\frac{144}{5}\left(-\frac{\ell^5}{32}\right)+8\frac{\ell^2\ell^3}{8}-\frac{\ell^4\ell}{2}\right)\right]=$$

$$-\frac{q}{2}\left[a_1\left(\ell^4\frac{\ell}{2}-\frac{8\ell^2}{3}\frac{\ell^3}{8}+\frac{16}{5}\frac{\ell^5}{32}\right)-a_1\left(\ell^4\left(-\frac{\ell}{2}\right)+\frac{8}{3}\ell^2\left(\frac{\ell^3}{8}\right)-\frac{16}{5}\frac{\ell^5}{32}\right)\right]=$$

$$=128\,EJ\left[a_1^2\left(\frac{9\cdot2\ell^5}{10}-2\ell^5+\frac{\ell^5}{2}\cdot2\right)\right]-\frac{q}{2}\left[a_1\left(\frac{\ell^5}{2}\cdot2-\frac{2\ell^5}{3}+\frac{2\ell^5}{10}\right)\right]=$$

$$=128\,EJ\,[a_1^2(1{,}8\ell^5-\ell^5)]-\frac{q}{2}[a_1(1{,}2\ell^5-\frac{2}{3}\ell^5)=$$

$$=128\,EJ\,a_1^2\cdot0{,}8\ell^5-\frac{qa_1\ell^5\cdot16}{30}=f(a_1);$$

$$\frac{df}{da_1}=0; \quad \frac{2a_1\cdot 128EJ\cdot 8l^5}{10}-\frac{ql^5\cdot 16}{30}=0;$$

$$a_1=\frac{ql^5\cdot 16\cdot 10}{30\cdot 2\cdot 128EJ\cdot 8l^5}=\frac{q}{3\cdot 128EJ};$$

$$W_1(x)=\frac{q}{384EJ}(l^2-4x^2)^2.$$

При $x=0$ /в середині балки/ $W_1(0)=\frac{ql^4}{384EJ}$, що повністю відповідає точному розв"язку задачі класичним методом сил. Наведемо це розв"язання.

Балка, зображена на рис. 2.33, згідно з методом сил тричі статично невизначена.

Запишемо канонічні рівняння за цим методом:

$$\begin{cases}\delta_{11}x_1+\delta_{12}x_2+\delta_{13}x_3+\Delta_{13}=0\\ \delta_{21}x_1+\delta_{22}x_2+\delta_{23}x_3+\Delta_{23}=0\\ \delta_{31}x_1+\delta_{32}x_2+\delta_{33}x_3+\Delta_{33}=0.\end{cases}$$

Щоб визначити коефіцієнти цих рівнянь, виберемо основну систему методу сил і побудуємо одиничні й вантажні епюри.

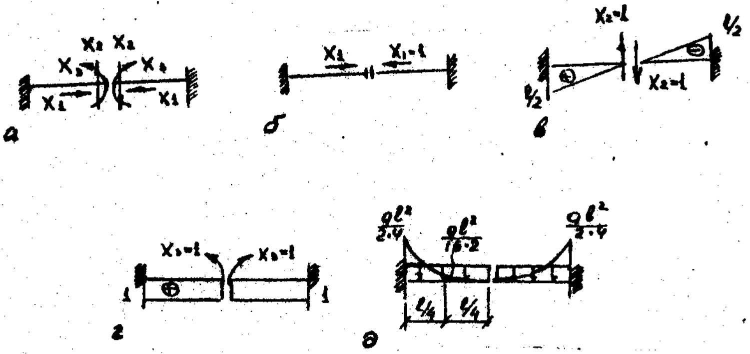


Рис. 2.34. Основна система методу сил ($a$) та епюри: $M_1^0$ /б/; $M_2^0$/в/; $M_3^0$/г/; $M_p^0$/д/

Визначаємо коефіцієнти канонічних рівнянь:

$$\delta_{11}=M_1^0M_1^0=0;\ \Delta_{1p}=0;\ \Delta_{2p}=0;\ \delta_{23}=\delta_{32}=0;$$

$$\delta_{12}=0;\ \delta_{13}=0;\ \Delta_{2p}=0;$$

$$\delta_{33}=M_3M_3=\frac{1\cdot l\cdot 1}{EJ}=\frac{l}{EJ};\ \delta_{22}=(\frac{l}{2}\cdot 1)1\cdot 1\frac{2}{EJ}=\frac{l}{EJ};$$

$$\Delta_{3p} = 2\left(-\frac{\ell}{2\cdot 6EJ}\left[\frac{q\ell^2}{8}\cdot 1 + 4\cdot 1\frac{q\ell^2}{32}\right]\right) = -\frac{2\ell\, q\ell^2 2}{2\cdot 6EJ\cdot 8} = \frac{-q\ell^3 2}{6\cdot 8EJ} = \frac{-q\ell^3}{24EJ}.$$

Отже, канонічна система запишеться у вигляді

$$\begin{cases} 0\cdot x_1 + 0\cdot x_2 + 0\cdot x_3 + 0 = 0; \\ 0\cdot x_1 + \ell/EJ x_2 + 0\cdot x_3 + 0 = 0; \\ 0\cdot x_1 + 0\cdot \frac{1}{2} + \frac{\ell}{EJ}x_3 + \left(\frac{-q\ell^3}{24EJ}\right) = 0. \end{cases}$$

Корені системи: $x_1 = 0;\ x_2 = 0;\ x_3 = \frac{q\ell^2}{24}$.

Далі маємо: $M = M_p^0 + M_1 x_1 + M_2 x_2 + M_3 x_3;$

$$M_{on} = \frac{q\ell^2}{8} - \frac{q\ell^2}{24} = \frac{q\ell^2}{12};$$

$$M_{(x=\ell/4)} = \frac{q\ell^2}{24} - \frac{q\ell^2}{32} = \frac{q\ell^2}{96}.$$

Шукані епюри зображено на рис. 2.35.

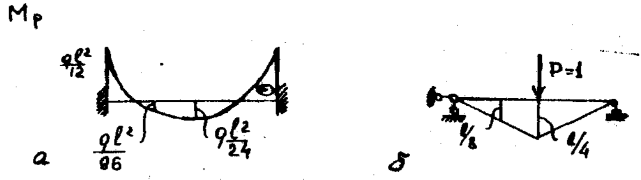


Рис. 2.35. Остаточний вантажний епюр $M_p$ /а/ та одиничний епюр $M_1$ в основній системі /б/

Щоб визначити переміщення середини балки, скористаємося формулою Симпсона-Карнаухова

$$\Delta = \frac{\ell}{6EJ}\left[M_p^A M_1^A + 4M_p^c M_1^c + M_p^B M_1^B\right], \qquad /2.139/$$

де $A, C, B$ – відповідно початок, середина та кінець стержня.

Побудуємо одиничний епюр в основній системі /див. рис. 2.35,б/ і перемножимо за /2.139/ з вантажним епюром моментів $M_p$ /див. рис. 2.35,а/:

$$\Delta = 2\left\{\frac{\ell}{2\cdot 6EJ}\left[4\frac{\ell}{8}\frac{q\ell^2}{96} + \frac{q\ell^2}{24}\frac{\ell}{4}\right]\right\} = \frac{q\ell^4}{384EJ}.$$

**Приклад 2.24.** Визначення стійкості системи з двома ступенями вільності.

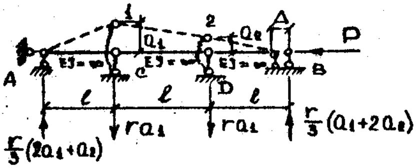


Рис. 2.36. Розрахункова схема системи з двома ступенями вільності

Для системи з двома ступенями вільності, показаної на рис. 2.36, знайти критичні сили, при яких можливе відхилення від рівноважного стану системи. Опори $C$ і $D$ – пружноподатливі з коефіцієнтом твердості $r$ /$r$ – це реакція в опорі при одиничному переміщенні її по вертикалі/.

Розв"язання. Розв"яжемо задачу так званим методом варіацій функції

$$u = A + W$$

/варіацією називають нескінченно малу зміну функції при фіксованому значенні незалежної змінної $x$ /.

Міркуємо за таким планом:

1. Обумовлюємо форму втрати стійкості з точністю до двох ступенів вільності $a_1$ і $a_2$ /їх фізична суть – це віртуальні лінійні переміщення точок C і $D$ /. Тоді робота зовнішніх сил $A = -P\Delta$, де $\Delta$ – поворот системи при зміщенні її кінця $\Delta = \ell/2\,(y')^2$.

Для першої ланки маємо $y' = a_1/\ell$, для другої $y' = a_1 - a_2/\ell$, для третьої $y' = a_2/\ell$.

Отже,

$$\Delta = \frac{\ell}{2}\left(\frac{a_1^2}{\ell^2} + \frac{a_2^2}{\ell^2} + \frac{(a_1-a_2)^2}{\ell^2}\right) = \frac{a_1^2 - a_1a_2 + a_2^2}{\ell}.$$

Робота внутрішніх сил $W = \frac{1}{2} r a_1^2 + \frac{1}{2} r a_2 a_2 = \frac{r}{2}(a_1^2 + a_2^2)$.

2. Записуємо вираз для потенціальної енергії системи

$$u = F(a_1, a_2, \ldots, a_n)$$

у вигляді

$$u = A + W = -P\,\frac{(a_1^2 - a_1a_2 + a_2^2)}{\ell} + \frac{r}{2}(a_1^2 + a_2^2).$$

3. З умов екстремуму функції $u$ маємо: $\frac{\partial u}{\partial a_1} = 0$; $\frac{\partial u}{\partial a_2} = 0$.

Тому дістаємо систему однорідних лінійних рівнянь відносно $a_1$ і $a_2$:

$$\begin{cases} -\frac{P}{\ell}(2a_1 - a_2) + ra_1 = 0; \\ -\frac{P}{\ell}(-a_1 + 2a_2) + ra_2 = 0, \end{cases}$$

звідки

$$\begin{cases} (\frac{2P}{\ell}-r)a_1-\frac{P}{\ell}a_2=0; \\ -\frac{P}{\ell}a_1+(\frac{2P}{\ell}-r)a_2=0. \end{cases}$$

Щоб ця система мала відмінний від нуля розв"язок, необхідно, щоб її визначник дорівнював нулю, тобто

$$D=\begin{vmatrix} \frac{2P}{\ell}-r & -\frac{P}{\ell} \\ -\frac{P}{\ell} & \frac{2P}{\ell}-r \end{vmatrix}=0,$$

або $(\frac{2P}{\ell}-r)^2-\frac{P^2}{\ell^2}=0$. Отже, $P_1=\frac{r\ell}{3}$; $P_2=r\ell$.

## 2.8. Математичні основи методу скінченних елементів. Розв"язування крайової задачі вигину балки

Метод скінченних елементів /МСЕ/ для описування суцільних середовищ вперше був використаний в середині 50-х років нашого століття і з тих пір завоював популярність як виключно корисний інженерний метод.

МСЕ і МСР застосовуються для розв"язування одних і тих самих задач, але грунтуються вони на різних ідеях. Згідно з МСР виконується різницева апроксимація похідних, які входять до диференціального рівняння. Математична основа МСЕ - варіаційне числення. Диференціальне рівняння, що описує розв"язувану задачу, і відповідні граничні умови використовуються для постановки варіаційної задачі, яка потім безпосередньо розв"язується.

Отже, МСЕ передбачає неявне застосування методу Рітца на окремих відрізках. За цим методом фізична задача замінюється деякою кусково-неперервною моделлю, побудова якої потребує, крім фактичних знань, певної інженерної інтуїції.

Розглянемо основні етапи застосування МСЕ.

### I. Розподіл тіла на елементи /дискретизація/.

МСЕ полягає в тому, що суцільне тіло /континуум/ уявляється як сукупність окремих скінченних елементів /СЕ/, які взаємодіють між собою в обмеженому числі вузлових точок. Приклад такої системи подано на рис. 2.37.

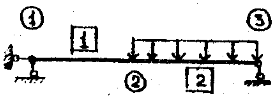


Рис. 2.37. Сукупність двох СЕ та трьох вузлів

Така ідеалізація вихідного пружного тіла з нескінченним числом ступенів вільності дає змогу перейти до розрахунку пружної системи з обмеженим числом ступенів вільності. Заміна вихідної конструкції сукупністю дискретних елементів грунтується на припущенні про рівність енергії конструкції та її дискретної розрахункової моделі. Для стержневих конструкцій /балок, арок, ферм/, які також складаються з окремих елементів, що дискретно сполучені між собою, МСЕ дає розв"язок з наперед заданою точністю всіх результатів обчислення. Для таких конструктивних елементів, як плити чи оболонки, результат виходить наближеним.

Таким чином, розрахункова модель для стержневих конструкцій являє собою обмежену сукупність вузлів, сполучених між собою кінцевими стержневими елементами.

## 2. Визначення основних залежностей МСЕ.

Спочатку МСЕ був відомий як метод жорсткостей. В основу алгоритмів розрахунку сучасних конструкцій МСЕ покладено метод переміщень, який виявився значно ефективнішим, ніж метод сил.

Щоб утворити єдину систему із СЕ, об"єднаних у вузлах розрахункової схеми, необхідно записати умови:

1/ рівноваги сил у вузлах;

2/ нерозривності переміщень у вузлах;

3/ залежності між переміщеннями і реакціями.

Умови нерозривності переміщень виконуються автоматично, оскільки переміщення вузлів розрахункової схеми є спільними для кількох СЕ, об"єднаних в одному вузлі.

При визначенні пружно-деформованного стану розрахункової стержневої моделі методом МСЕ /у формі методу переміщень/ відповідні рівняння складають, виходячи з рівноваги сил у вузлах розрахункової моделі.

Для будь-якого вузла $d$ рівняння рівноваги в матричній формі

$$\{F_d\} - \{\bar{R}_d\} = 0, \qquad /2.140/$$

де $F_d$ - вектор вузлових навантажень; $\bar{R}_d$ - вектор сумарних реакцій вузла $D_d$ на всі стержні, які прилягають до нього; $\{R\}$ - вектор реакцій від переміщень вузлів; $\{P\}$ - вектор реакцій від прогинаючого навантаження.

Лінійна залежність між переміщеннями вузлів і вузловими реакціями /фізичні рівняння/ у матричному вигляді така:

$$\{R\} = [k]\{\Delta\}, \qquad /2.141/$$

де $[k]$ – матриця жорсткості; $\{\Delta\}$ – вектор переміщень.

З урахуванням /2.141/ рівняння рівноваги набирають вигляду

$$\{F\} = [K]\{\Delta\} + \{P\}. \qquad /2.142/$$

3. Побудова функціоналу /запис виразу для потенціальної енергії елемента/.

Оскільки МСЕ є загальним методом дослідження задач техніки та фізики, матрицю жорсткості складають, абстрагуючись від розмірності елементів і фізичної природи задачі. При такому підході систему рівнянь, розв"язуваних МСЕ, дістають безпосередньо з умови мінімуму функціоналу повної потенціальної енергії, тобто варіаційного рівняння Лагранжа

$$\delta \Pi = \delta (U - A) = 0; \qquad /2.143/$$

де $U$ – робота внутрішніх сил /потенціальна енергія деформації/;
$A$ – робота зовнішніх сил /потенціал зовнішніх сил/.

Потенціальну енергію деформацій через внутрішні зусилля можна згідно з /2.138/ записати так:

$$U = \sum \int_0^l \frac{M^2 dx}{2EJ} + \sum \int_0^l \frac{N^2 dx}{2EF} + \sum \int_0^l 2 \frac{Q^2 dx}{2GF}.$$

Третій доданок у /2.138/ для стержневих систем являє собою величину порядку кількох процентів від першого і другого доданків. Тому в розрахунках третім доданком, як правило, нехтують.

Введемо такі позначення:

$W_y$ і $U_x$ – прогин та лінійне зміщення точок стержня;
$EJ$ і $EF$ – згинаюча та поздовжня жорсткості стержня;
$q_1$ і $q_2$ – прогинаючі рівномірно розподілені по балці навантаження вздовж осей $Ox$ та $Oy$.

Робота внутрішніх сил стержневого елемента

$$U = \frac{EJ}{2}\int_0^\ell \left(\frac{d^2W_y}{dx^2}\right)^2 dx + \frac{EF}{2}\int_0^\ell \left(\frac{dU_x}{dx}\right)^2 dx .$$

Робота відповідних зовнішніх сил

$$\frac{1}{2}\int_0^\ell q_1 U_x dx + \frac{1}{2}\int_0^\ell q_2 W_y dx .$$

Повна потенціальна енергія $i$ -го стержневого елемента

$$\Pi_i = \frac{EJ}{2}\int_0^\ell \left(\frac{d^2W_y}{dx^2}\right)^2 dx + \frac{EF}{2}\int_0^\ell \left(\frac{dU_x}{dx}\right)^2 dx - \frac{1}{2}\int_0^\ell q_1 U_x dx - \frac{1}{2}\int_0^\ell q_2 W_y dx . \qquad /2.I44/$$

## 4. Апроксимація переміщень внутрішніх точок елемента.

Розв"язати задачу у варіаційній постановці - означає знайти таку систему переміщень, яка мінімізує функціонал повноЇ потенціальноЇ енергіЇ системи. Вихідними даними для розв"язування є навантаження, геометричні та фізичні характеристики системи і граничні /межові/ умови. Як прийнято у варіаційних методах, шукані функціЇ переміщень наближено подають як набір компонентів

$$\Delta = \sum_1^n a_i u_i , \qquad /2.I45/$$

де $u_i$ - апроксимуючі /координатні/ функціЇ; $a_i$ - коефіцієнти, які в МСЕ називаються ступенями вільності /у фізичному розумінні $a_i$ - лінійні або кутові переміщення/.

Виразимо для плоскоЇ стержневоЇ системи /наприклад, рами, балки, ферми/ переміщення $W_y$ і $U_x$ внутрішніх точок елемента через переміщення $\{\Delta\}$ кінців СЕ.

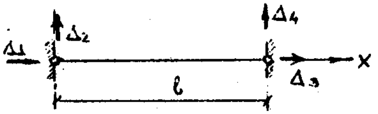
На практиці для цього використовуються поліноми, причому Їх вигляд і порядок залежать від розмірності СЕ /одно-, дво- і тривимірні/ та числа вузлів в елементі і в кожному конкретному випадку визначаються згідно з умовами розв"язуваноЇ задачі. Чим точніше вибраний поліном описує дійсні переміщення усередині елемента, тим точніший розв"язок задачі.

Для стержневих елементів, що перебувають в умовах розтягнення, стиснення чи вигину, функції $U_x$ і $W_y$ можна апроксимувати поліномами виду

$$U_x = \alpha_1 + \alpha_2 x ;$$
$$W_y = \alpha_3 + \alpha_4 x + \alpha_5 (x)^2 + \alpha_6 (x)^3 , \qquad /2.146/$$

де $x$ - координата точок скінченного елемента; $\alpha_i$ - коефіцієнти поліномів, які визначаються в процесі розв"язування задачі.

Апроксимація виразами типу /2.146/ виправдана тим, що переміщення точок вигнутого стержня описується функціями третього порядку /балочними функціями/. Щоб пов"язати коефіцієнти $\alpha_i$ полінома із значеннями ступенів вільності, записуються граничні умови на кінцях СЕ. Наприклад, для стержня зв"язок можна виразити так, як показано на рис. 2.38.



$$x = 0 ; \; \alpha_1 = \Delta_1$$
$$x = l ; \; \alpha_2 = 1/l \, (\Delta_3 - \Delta_1) \; ; \quad \text{и т. д.}$$

Рис. 2.38. Зв"язок коефіцієнтів $\alpha_i$ полінома із значеннями ступенів вільності

## Мінімізація потенціальної енергії системи. Побудова матриці жорсткості

Побудова матриці жорсткості, яка встановлює зв"язок між зусиллями і переміщеннями, базується на варіаційних принципах будівельної механіки. Наприклад, застосовується принцип Лагранжа. Якщо система в рівновазі, то з усіх можливих переміщень, які задовольняють задані граничні умови, справжніми будуть ті, при яких повна потенціальна енергія набуває стаціонарного значення.

Оскільки енергія всієї системи зображується сумою енергій окремих елементів, то принцип Лагранжа справджуватиметься і для окремого СЕ. У результаті введення апроксимації типу /2.146/ функціонал повної потенціальної енергії системи стає скінченновимірною функцією ступенів вільності / $i = 1,2,\ldots,n$/. Ступені вільності визначаються з умови мінімуму функціоналу, тобто з рівняння

$$\frac{\partial}{\partial a_i} \Pi(a) = \frac{\partial}{\partial a_i} U(a) - \frac{\partial}{\partial a_i} W(a) = 0 \; (i = 1,2,\ldots,n). \qquad /2.147/$$

Продиференціювавши функції декількох змінних, отримаємо систему алгебраїчних рівнянь:

$$\begin{cases} \frac{\partial}{\partial a_1}U(a_1)+\frac{\partial}{\partial a_1}U(a_2)+\ldots+\frac{\partial}{\partial a_1}U(a_n)=\frac{\partial}{\partial a_1}W(a_1); \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \frac{\partial}{\partial a_n}U(a_1)+\frac{\partial}{\partial a_n}U(a_2)+\ldots+\frac{\partial}{\partial a_n}U(a_n)=\frac{\partial}{\partial a_n}W(a_n). \end{cases} \qquad /2.148/$$

За аналогією з будівельною механікою стержневих систем ці рівняння називаються канонічними. Кожний їх член можна записати в вигляді добутку $k_{ij}\,a_i$, де $k_{ij}$ – коефіцієнти канонічних рівнянь, які утворюють матрицю жорсткості. Елементи правого стовпця рівнянь /2.148/ по суті є вузловими навантаженнями, до яких зводяться місцеві навантаження, розподілені по області системи. Цей стовпець /вектор/ позначимо $\{\bar{P}\}$. Таким чином, $\{\bar{P}_i\}=\frac{\partial}{\partial a_i}W(a_i)$.

Враховуючи введені позначення, /2.148/ можна записати в матричній формі:

$$\{\bar{P}\}=[k]\{a\}. \qquad /2.149/$$

Рівняння /2.149/ по суті й будуть розрахунковими рівняннями МСЕ. З фізичного боку їх можна тлумачити як рівняння рівноваги.

6. Розв"язування рівнянь згідно з МСЕ виконується за допомогою ЕОМ чисельними методами.

Науково-дослідний інститут автоматизованих систем планування і управління в будівництві Держбуду УССР /НДІ АСБ/ створив обчислювальний комплекс /ОК/ "Ліра", за допомогою якого проводяться міцнісні розрахунки щодо статичних і динамічних впливів будівельних конструкцій за допомогою ЄС ЕОМ. Обчислення виконуються МСК, який дає змогу поєднати розрахунки об"єктів будівельної механіки стержневих систем і об"єктів теорії пружності /плит, пластин, оболонок, масивів/. За допомогою ОК "Ліра" можна розраховувати складні стержневі системи, плити, балки, балки-стінки, оболонки, масивні тривимірні тіла, комбіновані системи, пластини на пружній основі та інші будівельні конструкції.

# ДОДАТОК

## Завдання для самостійної роботи

Пропонується ряд програм, які реалізують задачі лінійної алгебри та розв"язування диференціальних рівнянь. Деякі програми звертаються до програмного й математичного забезпечення ЄС ЕОМ – пакета наукових підпрограм, що має назву "Бібліотека інституту математики" /ПНП – БІМ/. Він містить понад 1000 підпрограм з чисельних методів аналізу й математичної статистики, записаних алгоритмічною мовою Фортран-IУ. Щоб закріпити теоретичний матеріал, пропонується виконати на ЄС ЕОМ варіанти завдань за названими далі темами /програми подаються з контрольними прикладами обчислень/.

1. Розв"язування системи лінійних алгебраїчних рівнянь /звертання до стандартної підпрограми *SIMQ*/.

2. Обчислення власних чисел і власних векторів матриці /звертання до підпрограми *EVVGM* /.

3. Знаходження коренів полінома /звертання до підпрограми *POLRT*/.

4. Розв"язування диференціального рівняння методом Адамса /"початковий" відрізок обчислюється методом Ейлера/.

5. Побудова епюр $M$ і $Q$ /моментів і поперечних сил/ в статично визначеній балці, навантаженій зосередженими силами і розподіленим навантаженням.

6. Обчислення переміщень в будівельних конструкціях.

7. Обчислення епюр $M, Q, N$ в статично визначених системах.

<u>Завдання 1</u>. Використовуючи програму виклику підпрограми *SIMQ* /табл. Д.1/ знайти розв"язки системи лінійних алгебраїчних рівнянь виду $Ax=B$:

$$\begin{vmatrix} 2 & 3 & 6 & 1 \\ 4 & 2 & 1 & 2 \\ 1 & 3 & 1 & 1 \\ 3 & 1 & 1 & 6 \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{vmatrix} = \begin{vmatrix} 2 \\ 3 \\ -1 \\ 4 \end{vmatrix} .$$

Розмірність вектора $x$ дорівнює 4.

<u>Розв"язання.</u>

Таблиця Д.1

```
      DIMENSION AX(4,4),DX(4),XX(4)
      READ(5,2)((AX(I,J),J=1,4),I=1,4)
      PRINT 2,((AX(I,J),J=1,4),I=1,4)
    2 FORMAT (4F5.2)
      READ(5,3) NX,DX
      PRINT 3,NX,DX
```

```
    3 FORMAT(1I2,4F6.2)
      CALL SIMQ(AX,DX,NX,KS)
      WRITE(6.4) DX,KS
    4 FORMAT(5(5XG12.4))
      STOP
      END
```

Результати обчислень:

```
2,00 3,00 6.00 1,00
4,00 2,00 1.00 2.00
1.00 3,00 1,00 1.00
3.00 1.00 1.00 6,00
4  2.00  3.00 -1.00  4.00
   0.9515          -.8767              0.4097             0.2687
```

Відповідь. $x_1 = 0,9515$; $x_2 = -0,8767$; $x_3 = 0,4097$; $x_4 = 0,2687$.

<u>Завдання 2</u>. Звертаючись до підпрограми *EVVGM*, /табл. Д.3/, знайти власні числа і власні вектори матриці

$$A = \begin{vmatrix} 2 & 3 & 5 & 7 \\ 7 & 6 & 4 & 9 \\ 1 & 4 & 7 & 10 \\ 62 & 5 & 8 & 9 \end{vmatrix}.$$

<u>Розв"язання</u>.

Таблиця Д.3

```
    DIMENSION A(4.4),VECR(4,4),VECI(4.4)
    DIMENSION EVR(4).EVI(4).FACT(4).SUBDIA(4).WORK(4).WORK 1(4),
   *WORK 2(4).IWORK(4).LOKAL(4).IER(4)
    READ(5.2)((A(I,J).J=1.4),I=1,4)
    PRINT 2.((A(I,J),J=1.4).I=1.4)
  2 FORMAT (4F6.2)
    READ(5,3) N
    PRINT 3.N
  3 FORMAT(1I2)
    CALL EVVGM(N,4,EVR,EVI,VECR,VECI,FACT,SUBDIA,WORK,
   *WORK 1,WORK 2,IWORK,LOCAL,IER,0)
    WRITE (6,4) IER,EVR,EVI
  4 FORMAT (4G10,4)
    WRITE (6,5) VECR,VECI
  5 FORMAT (4F10,4)
    STOP
    END
```

Результати обчислень:

```
  2,00  3,00  5,00  7,00
  7.00  6,00  4,00  9,00
  1.00  4,00  7,00 10.00
 62.00  5.00  8.00  9.00
 4
          2            2            2            2


 12.66       34.73       -.3458       2.269
```

```
.0           0.0           0.0           0.0
  0,2326      0,2479      0,3762     -0,8619
 -0.2630     -0,3695     -0,3572     -0,8165
 -0,0217      0,5642      0,5425     -0,6220
  0.0255     -0,9406      0.1261      0,3141
  0,0         0,0         0,0         0,0
  0,0         0,0         0,0         0,0
  0,0         0.0         0,0         0,0
  0.0         0.0         0.0         0.0
```

Відповідь. $\lambda_1 = 12,66$; $\lambda_2 = 34,73$; $\lambda_3 = -0,3458$; $\lambda_4 = 2,269$;

$$\vec{x}_{\lambda_1} = \begin{vmatrix} 0,2326 \\ -0,2630 \\ -0,0217 \\ 0,0255 \end{vmatrix}; \vec{x}_{\lambda_2} = \begin{vmatrix} 0,2479 \\ -0,3695 \\ 0,5642 \\ -0,9406 \end{vmatrix}; \vec{x}_{\lambda_3} = \begin{vmatrix} 0,3762 \\ -0,3572 \\ 0,5425 \\ 0,1261 \end{vmatrix}; \vec{x}_{\lambda_4} = \begin{vmatrix} -0,8619 \\ -0,8165 \\ -0,6220 \\ 0,3141 \end{vmatrix}.$$

Завдання 3. Використовуючи програму виклику підпрограми *POLRT* /табл. Д.2/, знайти корені полінома виду

$$ax^5 + bx^4 + cx^3 + dx^2 + ex^1 + fx^0 = 0;$$
$$2x^5 + 6x^4 + 4x^3 - x^2 + 3x^1 + 2 = 0.$$

Порядок полінома $M = 5$.

Розв"язання.

Таблиця Д.2

```
      DIMENSION XCOF(6),ROOTR(5),ROOTI(5)
      READ(5,3) M,XCOF
      PRINT 3,M,XCOF
    3 FORMAT(1I2,6F6.2)
      CALL POLRT (XCOF,COF,M,ROOTR,ROOTI,IER)
      WRITE(6,4) IER,ROOTR,ROOTI
    4 FORMAT(1I2/(3XG10.4))
      STOP
      END
```

Результати обчислень:

```
5  2.00  3.00 -1.00  4.00  6.00  2.00
0
   -,5172
   -,5172
   7,6093
   3,6093
   -2,736
   -,4491
   0,4491
   -,8108
   0,8108
   0,0
```

Відповідь.

$$x_1=-0,5172-i\cdot 0,4491;$$
$$x_2=-0,5172+i\cdot 0,4491;$$
$$x_3=0,6093-i\cdot 0,8108;$$
$$x_4=0,6093+i\cdot 0,8108;$$
$$x_5=-2,736.$$

Завдання 4. Розв"язати диференціальне рівняння $y'=xy(1-x^2)$ з початковими умовами $y(0)=4$ методом Адамса:

$$y_{n+1}=y_n+h\left(f_n+\frac{1}{2}\Delta f_n+\frac{5}{12}\Delta^2 f_n+\frac{3}{8}\Delta^3 f_n\right).$$

"Початковий" відрізок знайти методом Ейлера:

$$y_{n+1}=y_n+hf_n;\quad y'=f_n.$$

Крок $h=0,1,\ x_0=2,\ x_n=3,\ y_0=4.$

Розв"язання. Складаємо програму /табл. Д.4/:

```
        F(X,Y)=X*Y*(1.-X**2)
        READ(5,5)H,X0,XN,Y0
        PRINT 5,H,X0,XN,Y0
        Y1=Y0+H*F(X0,Y0)
        Y2=Y1+H*F(X0+H,Y1)
        Y3=Y2+H*F(X0+2.*H,Y2)
        PRINT 5,Y0,Y1,Y2,Y3
        F10=H*F(X0,Y0)
        F11=H*F(X0+H,Y1)
        F21=F11-F10
        F12=H*F(X0+2.*H,Y2)
        F13=H*F(X0+3.*H,Y3)
        F22=F12-F11
        F23=F13-F12
        F31=F22-F21
        F32=F23-F22
        F41=F32-F31
        X=X0+3.*H
  1     X=X+H
        Y=Y3+(F13+0.5*F23+5./12.*F32+3./8.*F41)*H
        PRINT 3,X,Y
        IF(ABS(X-XN)-H/2.)4,2,2
  2     F11=H*F(X,Y)
        F12=F11-F13
        F10=F12-F23
        F21=F10-F32
        Y3=Y
        F13=F11
        F23=F12
        F32=F10
        F41=F21
        GO TO1
      3 FORMAT(1X,2(F8.3,2X))
      5 FORMAT(1X,4(F8.3,1X))
      4 STOP
        END
```

Початковий відрізок знаходимо згідно з табл. Д.5.

Таблиця Д.5

```
 H - 0,100  X0-2.000  XN-3,000  Y0=4,000
 Y0 - 4,000  Y1-1.600  Y2-0.454  Y3-0.070 -> Початковий відрізок
 X4- 2,400  Y4= -0,012
 X5= 2,500  Y5=  0,028
     2,600      -0,054
     2,700       0,269
     2,800      -1,039
     2,900       4,782
     3.000     -24.393
```

Результати обчислень:

| $x_i$ | $y_i$ | $\Delta y_i$ | $q_i$ | $\Delta q_i$ | $\Delta^2 q_i$ | $\Delta^3 q_i$ |
|---|---|---|---|---|---|---|
| $x_0$ | $y_0$ | | $q_0$ | $\Delta q_0$ | $\Delta^2 q_0$ | $\Delta^3 q_0$ |
| $x_1$ | $y_1$ | | $q_1$ | $\Delta q_1$ | $\Delta^2 q_1$ | $\Delta^3 q_1$ |
| $x_2$ | $y_2$ | | $q_2$ | $\Delta q_2$ | $\Delta^2 q_2$ | |
| $x_3$ | $y_3$ | | $q_3$ | $\Delta q_3$ | | |
| | $y_4$ | | $q_4$ | | | |

<u>Завдання 5</u>. Побудувати епюри $M$ і $Q$ в статично визначеній балці /розрахункова схема балки і схема навантаження наведені в прикладі I.10/.

<u>Розв"язання</u>. Складаємо програму /табл. Д.6/.

Таблиця Д.6

```
      REAL MBX,L
      COMMON /PI/ KP,P(10).A(10)
      COMMON /QI/ KO.Q(5).B(5),C(5)
      READ (5,1) L,KP.KO
    1 FORMAT (F8.3,2I4)
      READ (5,8) P,A.Q.B,C
    8 FORMAT(10F8.2/10F8.3/5F8.3/5F8.3/5F8,3)
      WRITE(6,11)L,KP.KO
   11 FORMAT(1X.'L=',F8.3.' KP=',I4.' KO=',I4/1X,50('-'))
      WRITE(6,12)P
   12 FORMAT(1X,'MACCИB PI'/1X,10F8.2/1X,50('-'))
      WRITE(6,13)A,Q,B.C
   13 FORMAT(1X.10F8.3/1X,50('-')/1X,5F8.3/1X.50('-')/1X,5F8.3/
     *  1X,50('-')/1X.5F8.3/1X,50('-'))
      CALL BALKA(MBX,QBX,0..L.L)
      RA=-MBX/L
      RB=-QBX-RA
      KS=13
      DO 3 I=1,KS
      X=L*(I-1.)/(KS-1.)
      CALL BALKA(MBX,QBX,RA.L.X)
      WRITE(6,15) X.MBX,QBX
   15 FORMAT(3X.'X=',F8.3.3X,'MBX=',F10.3.3X,'QBX=',F10.3)
    3 CONTINUE
      STOP
      END
```

```
      SUBROUTINE BALKA(MBX,QBX,RA,L,X)
      REAL MBX,L
      COMMON /PI/ KP,P(10),A(10)
      COMMON /QI/ KQ,Q(5),B(5),C(5)
      MBX=RA*X
      QBX=RA
      DO 4 I=1,KP
      IF (X-A(I)) 4,4,2
    2 MBX=MBX-P(I)*(X-A(I))
      QBX=QBX-P(I)
    4 CONTINUE
      DO 6 I=1,KQ
      IF (B(I)-X) 5,6,6
    5 IF(X-(B(I)+C(I)))7,7,8
    7 MBX=MBX-Q(I)*(X-B(I))**2/2.
      QBX=QBX-Q(I)*(X-B(I))
      GOTO 6
    8 MBX=MBX-Q(I)*C(I)*(X-B(I)-C(I)/2.)
      QBX=QBX-Q(I)*C(I)
    6 CONTINUE
      RETURN
      END
```

**Результати обчислень:**

```
L=   6.000 KP=  10 KQ=    5
------------------------------------------------------------------------------
MACOHB PI
   2.00     4.00     8.00     9.00     7.00    10.00  18.  20.   7.  12.
------------------------------------------------------------------------------
  0.500    1.000    1.500    2.000    2.500    3.000  3.5  4.0  4.5  5.0
------------------------------------------------------------------------------
  2.000    4.000    5.000    4.000    3.000
-----------------------------------------------------------------
  0.100    0.600    2.000    4.000    3.000
-----------------------------------------------------------------
  2.000    1.000    3.000    1.500    3.000
-----------------------------------------------------------------
   X=   0.0       MBX=     0.0     QBX=    60.450
   X=   0.500     MBX=    30.065   QBX=    59.650
   X=   1.000     MBX=    58.320   QBX=    55.050
   X=   1.500     MBX=    83.095   QBX=    48.050
   X=   2.000     MBX=   102.690   QBX=    38.650
   X=   2.500     MBX=   116.800   QBX=    26.950
   X=   3.000     MBX=   126.150   QBX=    17.450
   X=   3.500     MBX=   128.875   QBX=     3.450
   X=   4.000     MBX=   120.600   QBX=   -18.550
   X=   4.500     MBX=    99.825   QBX=   -44.550
   X=   5.000     MBX=    72.550   QBX=   -57.550
   X=   5.500     MBX=    36.900   QBX=   -73.050
   X=   6.000     MBX=    -0.000   QBX=   -74.550
```

Шукані епюри показані на рис. Д.І.

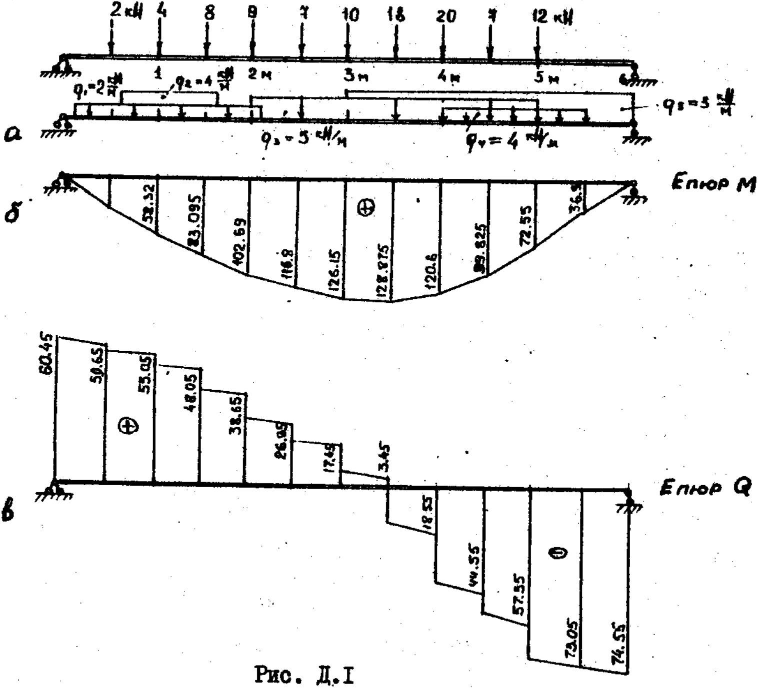


Рис. Д.І

Завдання 6. Обчислити горизонтальне та вертикальне переміщення точки $B$ рами та кут повороту рами в цій точці /рис. Д.2/.

Розв"язання. Запишемо векторну рівність

$$\vec{\Delta}_{iP} = \vec{m}_i^{\,T} * B_M * \vec{m}_P ,$$

де

$$\vec{m}_i = \begin{vmatrix} \vec{m}_{1i} \\ m_{2i} \\ \vdots \\ \vec{m}_{ni} \end{vmatrix} ; \quad \vec{m}_P = \begin{vmatrix} \vec{m}_{1P} \\ \vec{m}_{2P} \\ \vdots \\ \vec{m}_{nP} \end{vmatrix} ; \quad B_M = \begin{vmatrix} B_{1m} & & 0 \\ & B_{2m} & \\ 0 & & \ddots B_{nm} \end{vmatrix} ;$$

$\vec{m}_i$ - матриця одиничних епюр моментів; $\vec{m}_P$ - матриця грузових епюр моментів; $B_m$ - матриця піддатливості системи.

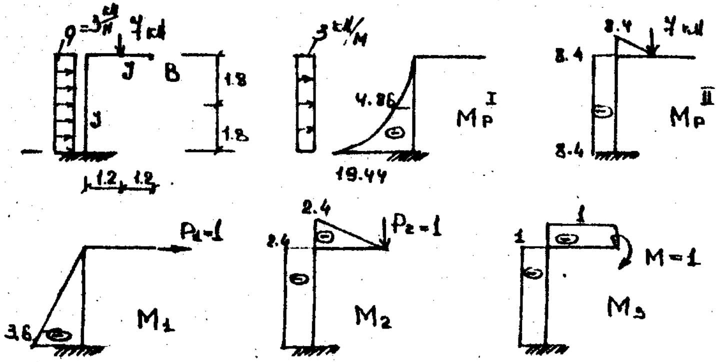


Рис. Д.2

Формуємо матрицю піддатливості:

$$B_{1m} = \frac{3,6}{6EJ}\begin{vmatrix}1 & & 0\\ & 4 & \\ 0 & & 1\end{vmatrix};\quad B_{2m} = B_{3m} = \frac{1,2}{6EJ}\begin{vmatrix}2 & 1\\ 1 & 2\end{vmatrix};$$

$$B_m = \begin{array}{|c|c|c|c|c|c|c|}\hline 0,6 & 0 & 0 & 0 & 0 & 0 & 0\\\hline 0 & 2,4 & 0 & 0 & 0 & 0 & 0\\\hline 0 & 0 & 0,6 & 0 & 0 & 0 & 0\\\hline 0 & 0 & 0 & 0,4 & 0,2 & 0 & 0\\\hline 0 & 0 & 0 & 0,2 & 0,4 & 0 & 0\\\hline 0 & 0 & 0 & 0 & 0 & 0,4 & 0,2\\\hline 0 & 0 & 0 & 0 & 0 & 0,2 & 0,4\\\hline\end{array}$$

$$\vec{m}_i = \begin{vmatrix}-3,6 & -2,4 & -1\\ -1,8 & -2,4 & -1\\ 0 & -2,4 & -1\\ 0 & -2,4 & -1\\ 0 & -1,2 & -1\\ 0 & -1,2 & -1\\ 0 & 0 & -1\end{vmatrix};\qquad \vec{m}_p = \begin{vmatrix}-19,44 & -8,4\\ -4,86 & -8,4\\ 0 & -8,4\\ 0 & -8,4\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{vmatrix}.$$

Переміщення точки $B$ обчислюємо за програмою /табл. Д.7/

Таблиця Д.7

```
      REAL MI(7,3),B(7,7),MP(7,2),R(3,2),X(3,7)
      READ(5,1) MI,B,MP
      PRINT 1,MI,B,MP
      CALL GTPRD(MI,B,X,7,3,7)
    1 FORMAT(7(3F8.3/),7(7F5.2/),7(2F8.3/))
      CALL GMPRD(X,MP,R,3,7,2)
      WRITE(6,2) R
    2 FORMAT(2F10.3)
      STOP
      END
```

Результати обчислень:

```
        -3,600   -2,400  -1,000
        -1,800   -2.400  -1,000
          0,0    -2,400  -1,000
          0,0    -2,400  -1,000
          0,0    -1,200  -1,000
          0,0    -1,200  -1,000
          0.0      0,0   -1.000
      0,60  0,0   0,0   0.0   0,0   0,0   0,0
      0,0   2,40  0.0   0,0   0,0   0,0   0,0
      0,0   0,0   0.60  0,0   0,0   0,0   0,0
      0,0   0,0   0,0   0,40  0,20  0,0   0,0
      0,0   0,0   0.0   0,20  0,40  0,0   0,0
      0,0   0,0   0.0   0,0   0,0   0,40  0,20
      0.0   0.0   0,0   0.0   0.0   0.20  0.40
     -19,440   -8.400
      -4,860   -8,400
        0,0    -8,400
        0,0    -8,400
        0,0     0,0
        0,0     0,0
        0.0     0.0

        106,730     59,914
         20,203     18,144
         12.096      5.040
```

Переміщення точки $$B = \frac{1}{EJ}\left|\begin{array}{cc} 106,73 & 59,914 \\ 20,203 & 18,144 \\ 12,096 & 5,04 \end{array}\right| \begin{array}{l} \leftarrow \Delta_B^{гор} \\ \leftarrow \Delta_B^{верт} \\ \leftarrow \varphi_B \end{array}$$

I-й варіант навантаження $Q = 3$ кН/м

2-й варіант навантаження $P = 7$ кН

Завдання 7. Знайти зусилля в стержнях ферми /рис. Д.3/методом перемноження матриць: $Q = L_Q \overline{P}$.

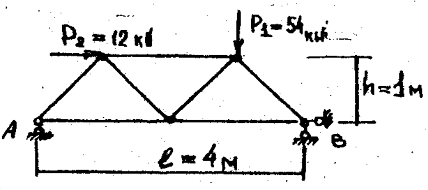
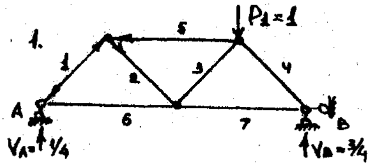


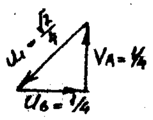
Рис. Д.3

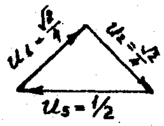
Етапи підготовки та формування матриці впливу $L_a$.

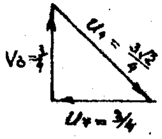


$\Sigma M_A=0; \qquad 1\cdot 3 - V_B\cdot 4=0; \rightarrow V_B - 3/4$

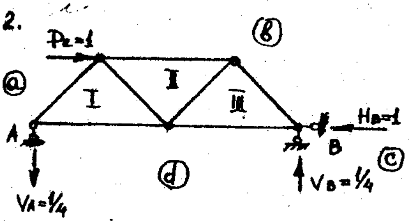
$\Sigma Y=0; \qquad Y_A = 1/4$

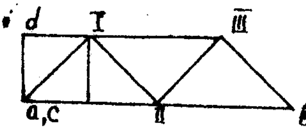






$$\begin{cases}\Sigma Y=0; \; -1+\frac{1}{4}\sqrt{2}\cdot\frac{\sqrt{2}}{2}+ \\ +U_5\cdot\frac{\sqrt{2}}{2}=0; \; U_3=\frac{\sqrt{2}}{4};\end{cases}$$





Матриця впливу поперечних сил

$$L_a = \begin{array}{c|c|c|} 1 & 0,35 & -0.35 \\ 2 & -0,35 & 0.35 \\ 3 & 0.35 & -2,10 \\ 4 & -0.35 & 0.35 \\ 5 & -0,50 & -0,50 \\ 6 & -0,25 & 0.25 \\ 7 & -0.25 & 0,25 \\ & N_1 & N_2 \end{array}$$

Матриці перемножуємо за програмою /табл. Д.8/:

Таблиця Д.8

```
      REAL A(7,2),B(2,1),R(7,1)
      READ(5,1) A
      PRINT 1.A
    1 FORMAT(7F10.3)
      READ(5,3)B
      PRINT 3.B
    3 FORMAT(2F10.3)
      CALL GMPRD(A,B,R,7,2,1)
      WRITE(6.2)R
    2 FORMAT(7F10.3)
      STOP
      END
```

Результати обчислень:

```
  0,350    -0.350     0,350    -0,350    -0,500    -0,250    -0,750
 -0,350     0,350    -2.100     0.350    -0,500     0.250     0.250
 54,000    12,000
 14.700   -14.700    -6.300   -14.700   -33.000   -10.500   -37.500
```

Відповідь. $N_1=14,7$, $N_2=-14,7$, $N_3=-6,3$, $N_4=-14,7$, $N_5=-33$, $N_6=-10,5$, $N_7=-37,5$.

# Зміст

Министерство высшего и среднего специального образования УССР
Учебно-методический кабинет по высшему образованию
Винницкий политехнический институт

Учебное издание

Шкодин Михаил Михайлович
Моргун Анатолий Иванович
Моргун Алла Серафимовна

## Программирование задач по строительству для ЭВМ

Учебное пособие

На украинском языке

Навчальне видання

Шкодин Михайло Михайлович
Моргун Анатолій Іванович
Моргун Алла Серафимівна

## Програмування задач з будівництва для ЕОМ

Навчальний посібник

Редактор О.П.Бондаренко
Коректори: С.М.Кушнір
Г.В.Мисливець
Т.С.Шленська
Н.Н.Савченко

Темплан 1990, поз. 264

Підп. до друку 23.11.90. . Формат $60\times84^{1}/_{16}$. Папір
друк. № 3 . Друк офсетний. Ум. др. арк. 8,37. Ум. фарбо-відб. 8,6
Обл.-вид. арк. 8,18 Тираж 1000.
Зам. № 01892р. Ціна 25к.

НМК ВО при Мінвузі УРСР ,252135,м.Київ,проспект Перемоги,10

РАПО «Укрвузполіграф».
252151, Київ, вул. Волинська, 60.