

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

Механіко-технологічний факультет
Кафедра програмування та захисту інформації

МЕТОДИЧНІ ВКАЗІВКИ
до виконання самостійних робіт з навчальної дисципліни
“Алгоритми та методи обчислень”
для студентів денної форми навчання за спеціальностями
6.050102/123 “Комп’ютерна інженерія”, 6.170103 “Управління інформаційною
безпекою», 125 “Кібербезпека”

ЗАТВЕРДЖЕНО
на засіданні кафедри програмування
та захисту інформації,
протокол від 21 серпня 2016 року № 9

КРОПИВНИЦЬКИЙ
2016

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни
“Алгоритми та методи обчислень” для студентів денної форми навчання за
спеціальностями 6.050102/123 “Комп’ютерна інженерія”, 6.170103 “Управління
інформаційною безпекою”, ”125 “Кібербезпека” / уклад. В.І. Петренюк—
Кропивницький: КНТУ, 2016. — 16 с.

Укладач: Петренюк В.І.

Рецензенти: Смірнов О. А., д-р техн. наук, професор;

Якименко Н. М., канд. Фіз.-мат. наук.

*Схвалено на засіданні методичного семінару
кафедри програмування та захисту інформації
(протокол від 29 серпня 2016 року 1)*

©
©

В.І. Петренюк, укладання, 2016
Кіровоградський національний
технічний університет, 2016

Вступ до предмету **Алгоритми та методи обчислень** .

Що це за наука - *обчислювальна математика*? Це наука за допомогою якої вирішуються завдання фізики, яка проникла в хімію, біологію, економіку, медицину, соціологію та інші науки. Виявляється, що ця наука використовує як математику, так і інформатику, а іноді і знання розглянутої предметної області. Підходи і методи обчислювальної математики своєрідні і красиві, а сама ця наука давно вже стала самостійною дисципліною, досліджуваною у всіх університетах світу. Як дізнатися, наскільки адекватні результати дає комп'ютерна модель, яку ви побудували, або, як прийнято говорити у обчислювальних математиків, чи є збіжність рішення моделі до вирішення вихідної «безперервної» завдання? Крім ряду складних і красивих теорем, є кілька простих хитрощів, які дозволяють визначити адекватність вашої моделі.

1. *Точність обчислень*. В машинних обчисленнях завжди присутній машинна похибка, про що вже згадувалося. І перш ніж вірити результатам, отриманим на комп'ютері, спробуйте збільшити точність всіх розрахунків на пару розрядів. Якщо це змінить результат у попередніх знаках, значить, обчислення проводилися з недостатньою точністю і потрібно використовувати більш точні типи для подання чисел (наприклад, в мові Cі float замінити на double, а в мові Паскаль тип real замінити на extended).

2. *Обумовленість*. У обчислювальній практиці велике значення має чутливість рішення до малих змін вхідних даних. Задача називається погано обумовленою, якщо малі зміни вхідних даних призводять до помітних змін рішення. Виміряти цю обумовленість на практиці дуже просто. Потрібно просто спробувати трохи поміняти вхідні дані і подивитися, як змінюється результат. Власне, потрібно з'ясувати, з якою похибкою задані вхідні дані, і експериментально перевірити в яких межах змінюється результат при варіюванні вхідних даних в межах їх похибки.

3. *Залежність від алгоритму і моделі*. Є ще одне джерело неадекватності чисельних результатів, отриманих на комп'ютері. Це неточність обраної моделі та алгоритму обчислення. Тут все дещо складніше, ніж у попередніх пунктах. Але для багатьох замовників можна запропонувати надійне рішення - слід дати одну і ту ж задачу кількома групами прикладних математиків. Вони, швидше за все, побудують різні моделі і будуть користуватися різними алгоритмами. Якщо все видадуть один і той же результат, значить завдання хороша - вона стійка до вибору моделі і алгоритму, і не так важливо яким способом її вирішувати. Звичайно, бувають і такі випадки, коли всі роблять одну і ту ж помилку. У початківців обчислювальних математиків є кілька таких улюблених помилок.

Зрозуміло, якщо є можливість порівняти результати розрахунку з експериментальними, то такі порівняння є приводом для підтвердження вірності розрахунків. Що ж стосується дослідження чисельних методів на апроксимацію - то ця теорія досить добре розроблена, проте без обчислювальних помилок чисельних рішень не буває. Питання полягає в тому, значні вони чи незначні, що, втім, також можна оцінити.

Наведемо деякі приклади, що демонструють що «жарти з похибкою погані».

Розв'яжімо гіпотетичну задачу: древньогрецький філософ-мандрівник спостерігав за рухом Місяця на фоні зірок та тіні від Сонця. Це дало йому можливість зробити висновок про форму та розміри Землі і також розрахувати відстань до Місяця. Припустимо, що він отримав результат 400000км з можливим відхиленням від дійсного результату ± 100000 км. В той самий час філософ-лунатик на Місяці міркував про свій розмір, сили тяжіння, кількість води на Сонці та хутра на бізонах, прийшов до висновку що зріст людей на Землі складає $2\text{м} \pm 1\text{м}$. Питання задачі полягає у порівнянні якості проведених числових оцінок – який філософ визначив величину точніше.

Для розв'язання цієї задачі є необхідним визначитися як порівнювати похибки і коли це можливо. Тому ведемо поняття похибки.

Означення: Абсолютна похибка – модуль різниці між точним значенням числа та його наближенням.

Наприклад, для рівності $\pi=3,14$ модуль різниці між даним наближенням π та більш точним його значенням:

$$|3,1415926-3,14|=0,00159265$$

дає абсолютну похибку рівності $\pi=3,14$.

Означення: Відносна похибка – відношення абсолютної похибки до модуля точного значення числа. Найчастіше виражається в процентах.

На практиці майже завжди неможливо визначити точні значення похибок і тому використовують поняття граничної похибки, тобто такої похибки, яка завжди більша або рівна реальному відхиленню. Надалі без додаткових вказівок всі похибки вважатимемо граничними.

Введемо позначення:

A – точне значення числа;

a – наближення числа A ;

Δ – абсолютна похибка;

ε – відносна похибка.

Тоді у вигляді формул:

$$\Delta a = |A - a|, \quad \varepsilon a = \Delta a / |A|, \quad \text{або в процентах } \varepsilon a = \Delta a / |A| \cdot 100.$$

Тепер повернемося до питання про філософів, хто з них точніше визначив величину? Для цього знайдемо відносні похибки відстані до Місяця:

$$\varepsilon_M = 100\% \cdot 100000 \text{ км} / 400000 \text{ км} = 25\%,$$

та зросту людини:

$$\varepsilon_L = 100\% \cdot 1 \text{ м} / 2 \text{ м} = 50\%.$$

В результаті порівняння відносних похибок бачимо, що похибка відстані до Місяця в два рази менша, і тому вважатимемо земного філософа більш точним.

Приклади:

1) Задане наближене число $a=3,142$, та його абсолютна гранична похибка $\Delta a = \pm 0,001$. Знайти відносну граничну похибку.

Розв'язання:

Так як ми не знаємо точного значення числа, то $\varepsilon a = \Delta a / a$:

$$\varepsilon a = 0,001 / 3,142 = 0,000318 = 0,0318\%$$

2) Відоме число $a=12,4$ та його відносна похибка $\varepsilon a = 1\%$. Знайти абсолютну похибку Δa .

Розв'язання:

$$\Delta a = a \varepsilon a = 12,4 \cdot 0,01 = \pm 0,124$$

3) Є дві рівності: $2/3=0,66$ та $11/7=1,57$. Визначити яке з них точніше.

Розв'язання:

Щоб визначити відповідь, достатньо порівняти відносні похибки цих

Для першої рівності:

$$\varepsilon_1 = |2/3 - 0,66| / (2/3) = 0,01 = 1\%.$$

Для другої рівності:

$$\varepsilon_2 = |11/7 - 1,57| / (11/7) = 0,000909 = 0,0909\%.$$

$\varepsilon_1 > \varepsilon_2$, тобто похибка першої рівності більша, тому друга рівність точніше.

4) На мові паскаль для запису чисел `single` з плавучою комою використовується 4-байтний запис (32 біти). В результаті для запису мантиси відводиться вісім десяткових цифр. Максимальне значення мантиси сягає 9.9999999, а мінімальне 1.0000000 з абсолютними похибками не більші за ± 0.00000005 . Тепер легко визначити відносні похибки:

$$\varepsilon_1 = 100\% \cdot 0.00000005 / 9.9999999 = 0.0000005\%$$

$$\varepsilon_2 = 100\% \cdot 0.00000005 / 1.0000000 = 0.000005\%$$

Як класифікують похибки?

Електронна обчислювальна машина (ЕОМ) насамперед створена для ефективного проведення математичних розрахунків. Всі схеми та їх поєднання в єдину систему розраховані на ефективне, швидке та якісне проведення арифметичних дій із числами різного формату. Та для кожної цифрової системи обчислення існує обмеження пам'яті для зберігання числа, тобто будь яке ірраціональне число та переважна більшість раціональних чисел дозволеного діапазону записані не повністю, а з обмеженою кількістю цифр у числі. Такі числа називають наближеними, а відмінність записаного числа від реальної величини — похибкою. Похибки можуть бути незначними і не впливати на результат обчислень, так і можуть бути значно більшими за величину, яку вимірювали. При практичних обчисленнях знання про похибку результату обчислення є дуже важливим, і в деяких випадках можуть застерегти від неправильного результату, бо іноді незнання відповіді є менш шкідливим ніж вважати за відповідь неправильне значення.

Практичні обчислення в більшості випадках проводяться з наближеними числами. І це зрозуміло, бо калькулятори та комп'ютери для представлення чисел використовують обмежену кількість пам'яті. Виявляється, що не завжди можна так вільно оперувати наближеними числами як точними, можливі ситуації коли результат обчислення буде мати дуже великі відхилення від дійсних значень. Для того щоб передбачити такі ситуації та вміти поставити обчислення найвигіднішим, щодо похибок, шляхом і вводиться цей розділ.

Розв'яжімо гіпотетичну задачу.

Древньогрецький філософ-мандрівник спостерігав за рухом Місяця на фоні зірок та тіні від Сонця. Це дало йому можливість зробити висновок про форму та розміри Землі і також розрахувати відстань до Місяця. Припустимо, що він отримав результат 400000км з можливим відхиленням від дійсного результату ± 100000 км. В той самий час філософ-лунатик на Місяці міркував про свій розмір, сили тяжіння, кількість води на Сонці та хутра на бізонах, прийшов до висновку що зріст людей на Землі складає $2\text{м} \pm 1\text{м}$.

Задача полягає у порівнянні якості проведених числових оцінок – який філософ визначив величину точніше.

Для розв'язання цієї задачі є необхідним визначитися як порівнювати похибки і коли це можливо. Тому ведемо поняття похибки.

Означення: Абсолютна похибка – модуль різниці між точним значенням числа та його наближенням.

Наприклад, для рівності $\pi=3,14$ модуль різниці між даним наближенням π та більш точним його значенням:

$$|3,1415926-3,14|=0,0015926,5$$

дає абсолютну похибку рівності $\pi=3,14$.

Означення: Відносна похибка – відношення абсолютної похибки до модуля точного значення числа. Найчастіше виражається в процентах.

На практиці майже завжди неможливо визначити точні значення похибок і тому використовують поняття граничної похибки, тобто такої похибки, яка завжди більша або рівна реальному відхиленню. Надалі без додаткових вказівок всі похибки вважатимемо граничними.

Введемо позначення:

A – точне значення числа;

a – наближення числа A ;

Δ – абсолютна похибка;

ε – відносна похибка.

Тоді у вигляді формул:

$$\Delta a = |A - a|, \quad \varepsilon a = \Delta a |A|, \quad \text{або в процентах } \varepsilon a = \Delta a |A| \cdot 100.$$

Тепер повернемося до питання про філософів, хто з них точніше визначив величину? Для цього знайдемо відносні похибки відстані до Місяця:
 $\epsilon_M = 100\% \cdot 100000 \text{ км} / 400000 \text{ км} = 25\%$,

та зросту людини:

$$\epsilon_L = 100\% \cdot 1 \text{ м} / 2 \text{ м} = 50\%$$

В результаті порівняння відносних похибок бачимо, що похибка відстані до Місяця в два рази менша, і тому вважатимемо земного філософа більш точним.

Приклади:

1) Задане наближене число $a = 3,142$, та його абсолютна гранична похибка $\Delta a = \pm 0,001$. Знайти відносну граничну похибку.

Розв'язання:

Так як ми не знаємо точного значення числа, то $\epsilon_a = \Delta a / a$:

$$\epsilon_a = 0,001 / 3,142 = 0,000318 = 0,0318\%$$

2) Відоме число $a = 12,4$ та його відносна похибка $\epsilon_a = 1\%$. Знайти абсолютну похибку Δa .

Розв'язання:

$$\Delta a = a \epsilon_a = 12,4 \cdot 0,01 = \pm 0,124$$

3) Є дві рівності: $2/3 = 0,66$ та $11/7 = 1,57$. Визначити яке з них точніше.

Розв'язання:

Щоб визначити відповідь, достатньо порівняти відносні похибки цих

Для першої рівності:

$$\epsilon_1 = |2/3 - 0,66| / (2/3) = 0,01 = 1\%$$

Для другої рівності:

$$\epsilon_2 = |11/7 - 1,57| / (11/7) = 0,000909 = 0,0909\%$$

$\epsilon_1 > \epsilon_2$, тобто похибка першої рівності більша, тому друга рівність точніше.

4) На мові паскаль для запису чисел single з плаваючою комою використовується 4-байтний запис (32 біти). В результаті для запису мантиси відводиться вісім десяткових цифр. Максимальне значення мантиси сягає 9.9999999, а мінімальне 1.0000000 з абсолютними похибками не більші за ± 0.00000005 . Тепер легко визначити відносні похибки:

$$\epsilon_1 = 100\% \cdot 0.00000005 / 9.9999999 = 0.0000005\%$$

$$\epsilon_2 = 100\% \cdot 0.00000005 / 1.0000000 = 0.000005\%$$

Сумнівні та вірні цифри числа

При вимірюваннях реальних величин за допомогою приладів, отримаємо числа які є наближеними. Наприклад, шкільна лінійка має мінімальну поділку в 1мм. Це обмежує точність визначення розмірів об'єкту одним міліметром. Також, коли виміряємо об'єкт різними лінійками, майже завжди отримаємо різні результати, бо лінійки зроблені з різного матеріалу по різному оброблялися та зазнали різноманітних деформацій. Але кожна лінійка має своє максимальне відхилення від точного значення і загалом не повинне перевищувати зазначеної величини. Для приладів, як правило, максимальне відхилення не перевищує половини ціни поділки. Тому провівши виміри тією ж лінійкою, можемо отримати таке значення: $(12,4 \pm 0,05) \text{ см}$.

Зрозуміло, що точне значення повинне лежати між 12,35 та 12,45, і якою б лінійкою ми не вимірювали (що відповідає стандартам), завжди матимемо 12см., а от кількість міліметрів буде то 3, то 4. Тому, в даному випадку називатимемо цифри, що виражають кількість сантиметрів вірними, а цифри, що виражають кількість сантиметрів – сумнівними.

Вірні цифри наближеного числа – ті цифри, які співпадають в точному та наближеному числі.

Сумнівні цифри – ті цифри, що не співпадають з цифрами точного числа.

Також можна сказати і по іншому:

Вірними цифрами наближеного числа називають цифри, що не змінюють свого значення, якщо похибка не перевищуватиме максимально вказану (граничну). Всі інші цифри називатимемо сумнівними.

На практиці наближені числа записують так, щоб лише остання значуща цифра могла містити похибку, тобто була сумнівною. Наприклад, запис 1,012 означає, що гранична абсолютна похибка рівна $\pm 0,0005$, а запис 1,0000 визначає похибку $\pm 0,00005$. Тому в наближених числах при запису відкидати “зайві” нулі не варто.

Приклади:

1) Відоме число $a=5,2351$ та його абсолютна похибка 0,003. Округлити число залишивши лише вірні цифри.

Розв’язання:

Проведемо округлення поетапно. Так як абсолютна похибка впливає на третій знак після коми, то можна округлити число зразу до другого знаку та перевірити, чи всі цифри в округленому числі вірні:

$$a=5,2351 \approx 5,235 \approx 5,24.$$

Знайдемо похибку, що дає округлення:

$$|5,2351 - 5,24| = 0,0049$$

та додамо до неї “рідну” похибку:

$$0,0049 + 0,003 = 0,0079.$$

В результаті $5,24 - 0,0079 \leq a \leq 5,24 + 0,0079$; $5,2321 \leq a \leq 5,2479$; $5,23 \leq a \leq 5,25$.

Тобто при округленні до тих самих двох знаків після коми, остання цифра може змінити своє значення, тому число a потрібно округлити ще на один знак (при необхідності цю операцію повторюють не один раз):

$$a \approx 5,2.$$

Знайдемо похибку, що дає округлення:

$$|5,2351 - 5,2| = 0,0351$$

та додамо до неї “рідну” похибку:

$$0,0351 + 0,003 = 0,0381.$$

В результаті $5,2 - 0,0381 \leq a \leq 5,2 + 0,0381$; $5,1679 \leq a \leq 5,2381$; $5,2 \leq a \leq 5,2$.

В результаті число 5,2 має лише вірні цифри.

Додавання та віднімання наближених чисел

Звісно, при використанні наближених значень величин в розрахунках ми також отримаємо наближені значення і для нього також необхідно знати величну похибки. Як далі буде показано, після неправильно поставлених розрахунках, відносна похибка після обчислень може зростати в багато разів зводячи нанівець всі зусилля обчислень.

Нехай ми маємо два наближених числа a та b . Маємо також їх точні значення A, B , та абсолютні похибки $\Delta a, \Delta b$. Тоді:

$$A+B=a+\Delta a+b+\Delta b=(a+b)+(\Delta a+\Delta b).$$

Тобто при складанні двох наближених чисел щоб знайти абсолютну похибку результату, потрібно скласти абсолютні похибки доданків. При відніманні картина не змінюється – похибки теж додаються, бо взявши число $b=-c$ отримаємо ті самі викладки.

Оцінити ж відносну похибку після додавання чи віднімання можна буде за такими співвідношеннями:

$$\varepsilon_a \square b = \Delta a \square \Delta b a \square b, \quad \varepsilon_{a-b} = \Delta a \square \Delta b a-b$$

Приклад:

Оцінити відносну та абсолютну похибки після додавання та відні

8

$$\varepsilon_a \square b =$$

$$\Delta a \square \Delta b$$

$$a \square b$$

$$, \varepsilon_{a-b} =$$

$$\Delta a \square \Delta b$$

$$a-b$$

.

Приклад:

Оцінити відносну та абсолютну похибки після додавання та віднімання двох чисел: $4,12 \pm 0,0042$ та $3,98 \pm 0,0037$.

Розв'язання:

Додавання:

$$\Delta 1 = 0,0042 + 0,0037 = 0,0079;$$

$$\varepsilon 1 = (0,0042 + 0,0037) / (4,12 + 3,98) = 0,000975 = 0,0975\%$$

Віднімання:

$$\Delta 2 = 0,0042 + 0,0037 = 0,0079;$$

$$\varepsilon 2 = (0,0042 + 0,0037) / (4,12 - 3,98) = 0,056 = 5,6\%$$

Відносна похибка відрізняється більш як в 50 разів!

1.4. Множення та ділення наближених чисел

Повернемось до позначень попереднього пункту 1.3. та проведемо перетворення для дії множення:

$$A \cdot B = (a + \Delta a) \cdot (b + \Delta b) = ab + a\Delta b + b\Delta a + \Delta a\Delta b \approx ab + (a\Delta b + b\Delta a).$$

Тобто

$$\Delta a \cdot b = a\Delta b + b\Delta a.$$

Для відносної похибки:

$$\varepsilon_a \cdot b = a\Delta b \square b\Delta a \quad ab = \Delta bb \square \Delta aa = \varepsilon_a \square \varepsilon_b .$$

Тепер повторимо перетворення для ділення:

$$\Delta a/b = a \square \Delta ab \square \Delta b - ab = ab \square b\Delta a - ab \square a\Delta bb \square b \square \Delta b \square = b\Delta a \square a\Delta bb \square b \square \Delta b \square ,$$

$$\Delta a/b = b\Delta a \square a\Delta bb \square b \square \Delta b \square = a \cdot b\Delta a/a \square \Delta b/bb \square b \square \Delta b \square = ab \square \Delta b \cdot \square \Delta a/a \square \Delta b/b \square ,$$

$$\Delta a/ba/b \approx \Delta aa \square \Delta bb.$$

$$\text{Або } \varepsilon_{a/b} = \varepsilon_a + \varepsilon_b.$$

Як показано в попередніх викладках, що при множенні, що при

Як показано в попередніх викладках, що при множенні, що при діленні, для знаходження відносної похибки результату потрібно скласти відносні похибки аргументів дії.

Приклад:

Дані числа: $a = 2,122 \pm 0,013$, $b = 5,5 \pm 0,06$. Знайти відносну та абсолютну похибку результату обчислень за формулою: $c = ab - a/b$.

Розв'язання:

$$ab = 11,671 \quad ,$$

$$a/b = 0,3858 \quad ,$$

$$\varepsilon_{a/b} = \varepsilon_a + \varepsilon_b = \Delta a/a + \Delta b/b = 0,013/2,122 + 0,06/5,5 \approx 0,017 \quad ,$$

$$\Delta ab = ab \cdot \varepsilon_{ab} = 11,671 \cdot 0,017 = 0,198 \approx 0,2 \quad ,$$

$$\Delta a/b = a/b \cdot \varepsilon_{a/b} = 0,3858 \cdot 0,017 = 0,00656 \quad ,$$

$$\Delta ab - a/b = \Delta ab + \Delta a/b = 0,2 + 0,00656 \approx 0,2$$

$$\text{Відповідь: } ab - a/b = 11,3 \pm 0,2 \quad , \quad \varepsilon_{ab - a/b} = 0,2/11,3 \approx 0,0177 = 1,77\%.$$

Відшукання похибки результату обчислень за допомогою диференціального числення

Ідея використання диференціального числення для визначення похибок показана на малюнку 1. Лише покажемо як цей принцип використати для

функцій з декількома змінними.

Нехай є функція $y=f(x_1, x_2, \dots, x_n)$, тоді знаючи відхилення i -тої змінної (Δx_i), можна оцінити максимальне відхилення значення функції, що залежить саме від i -тої змінної:

$$\Delta y_i \approx f'_{x_i} \cdot \Delta x_i$$

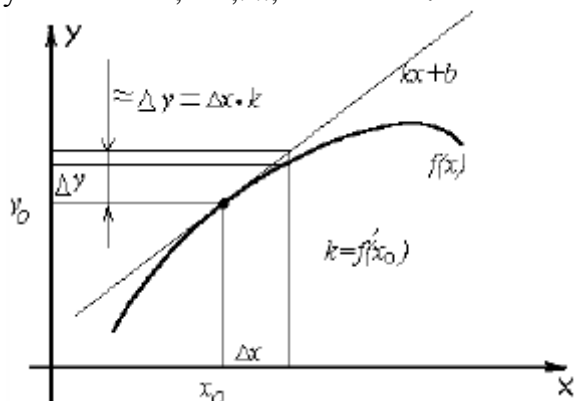


Рис. 1

Розв'язання рівнянь з однією невідомою

В школі та вищих навчальних закладах на заняттях з математики навчаються аналітично знаходити розв'язки рівнянь у вигляді однозначних виразів. На практиці ж рівняння, яке допускає аналітичне розв'язання зустрічається не часто, але розв'язувати такі рівняння все одно потрібно. Для розв'язання рівнянь розроблено багато чисельних методів.

Розв'язання рівнянь $f(x)=0$ починають з локалізації існуючих коренів, для цього людина повинна вказати комп'ютеру початковий інтервал, де будуть шукатися корені. Далі початковий проміжок розбивається на рівні частини (рис 4.1) на кінцях яких рахують значення $f(x)$. Для подальшої роботи відбирають проміжки, на кінцях яких значення $f(x)$ мають протилежні знаки. Звісно, якщо на етапі початкової локалізації коренів випадково потрапимо на один з коренів точно, то це лише позбавить нас від подальшого уточнення цього кореня.

Початкове розбиття інтервалу для пошуку коренів $\ln(x)-x+2=0$

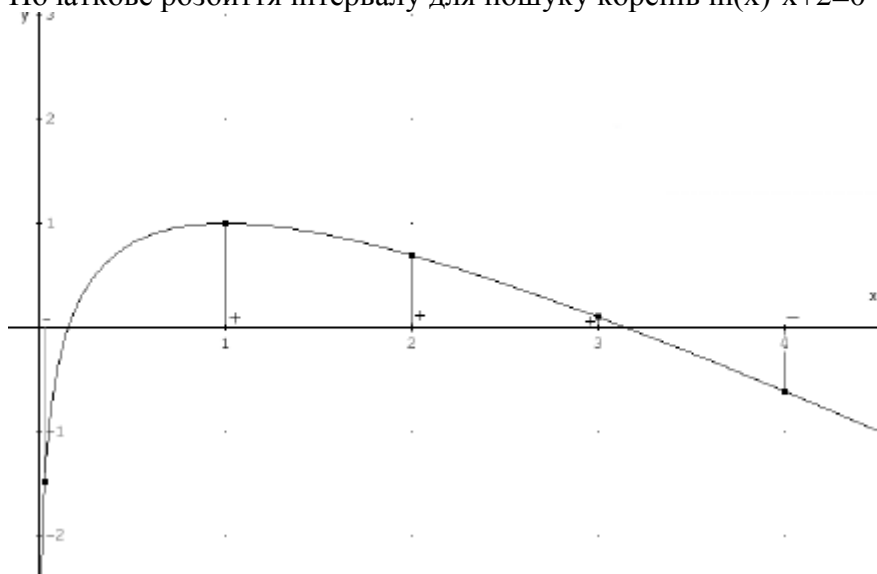


Рис. 4.1

На рисунку 4.1 показано початкове розбиття для виділення коренів функції $u(x)=\ln(x)-x+2$, і для подальшого уточнення коренів вибирають два інтервали $(0,1;1)$ та $(3;4)$. Іноді початкове розбиття з кроком в одиницю не дає змогу виділити інтервали з коренем, тоді проводять вибірку з меншим кроком, при

потребі крок зменшують ще декілька разів, перед тим як повідомити, що коренів не знайдено (на жаль такий процес початкового виділення коренів не доводить їх відсутність на пробному проміжку).

Такий процес виділення коренів необхідно виконувати для наступних алгоритмів послідовного звуження інтервалу, та бажано проводити і для інших методів уточнення коренів.

1. Методи послідовного звуження інтервалу

1.1. Метод простої дихотомії (половинного поділу) є найпростішим методом, який при наявності локалізованого кореня дає гарантовано збіжну послідовність наближення до кореню рівняння.

Процес відшукування коренів методом дихотомії покажімо далі. Нехай нам відомий інтервал де локалізовано один з коренів $[a_0; b_0]$, при цьому $f(a_0) < 0$ та $f(b_0) > 0$. Тепер поділимо інтервал навпіл $c_0 = (a_0 + b_0) / 2$ та визначимо знак $f(c_0)$. Тепер маємо три контрольних значення: $f(a_0) < 0$; $f(c_0) (> \text{ або } <) 0$; $f(b_0) > 0$ та два інтервали $[a_0; c_0]$, $[c_0; b_0]$. Один з цих інтервалів містить корінь рівняння. Звісно, той інтервал містить корінь, який на кінцях має різні знаки функції. Нехай $f(c_0) < 0$, тоді корінь буде на інтервалі $[c_0; b_0]$, або при $f(c_0) > 0$ — на інтервалі $[a_0; c_0]$. Новий інтервал позначимо $[a_1; b_1]$ і він є вдвічі меншим, тому можна сказати, що корінь знаємо тепер з вдвічі меншою абсолютною похибкою: $x = c_0 \pm |c_0 - b_0|$.

Процес можна повторювати, на кожному кроці зменшуючи абсолютну похибку вдвічі. Завдяки повній передбачуваності такого процесу можна визначити похибку визначення кореню при відомій довжині початкового інтервалу та відомій кількості операцій уточнення. Нехай початковий інтервал має довжину d , тоді за n операцій уточнення похибка зменшиться в 2^n раз: $\Delta x < d / 2^{n+1}$.

Спробуємо скласти алгоритм для пошуку коренів функції. Для успішної роботи функція повинна приймати вказівник на функцію, корінь якої ми повинні відшукати, початковий інтервал та бажану точність розрахунків:

```
float KorinDihotomiya(float (*f)(float x), float a,
float b, float delta)
{
    int i, n; //лічильник та кількість циклів уточнення
    float c; //середина пробного інтервалу
    n = log(fabs(b-a)/delta)/log(2);
    //тут визначається кількість ітерацій для
    //досягнення потрібної точності
    //  $\Delta x = d / 2^{n+1} \Rightarrow 1 = \log_2(d / \Delta x)$ 
    for(i=0; i<n; i++) //Повторюємо задану кількість
    уточнень
    {
        c = (a+b)*0.5; //Шукаємо середину інтервалу
        if( f(c) > 0.0 ) //перевірка знаку функції в
        середині інтервалу
        { //Звуження з права
            b = c;
        } else
        { //Звуження інтервалу зліва
            a = c;
        }
    }
    return (a+b)*0.5; //повернемо як корінь середину
    останнього інтервалу
}
```

Алгоритм простої дихотомії представлений мовою C++ можна переписати як і блок-схемами, так і іншими мовами програмування на вибір програміста.

Проаналізуємо роботу алгоритму на прикладі розв'язання нелінійного рівняння $\cos(x)-x=0$. Для початкового виділення кореня спробуємо значення $x=0$ та $x=1$: $\cos(0)-0>0$ та $\cos(1)-1<0$; тому $a=1$, $b=0$. Тепер можна перейти до звуження інтервалу. Серединою є $c=(a+b)/2=0.5$, значення функції є $\cos[0.5]-0.5\approx 0.3776>0$, що означає зміну $b=0.5$. Тепер маємо наступний інтервал $[0.5;1]$, середина якого є $c=(a+b)/2=(0.5+1)/2=0.75$. Значення функції в середині цього інтервалу є $\cos[0.75]-0.75\approx -0.0183<0$. Тепер маємо нове значення $a=0.75$ з інтервалом, де розташовано корінь $[0.5;0.75]$. Процес повторюється до звуження інтервалу, який забезпечить потрібну точність. У цьому випадку коренем є $x=0.625\pm 0.125$.

1.2. *Метод поділу на третини та золоті пропорції* алгоритмічно не відрізняється від половинного ділення, лише замість половинного ділення проміжку використовується ділення з іншими пропорціями при розрахунку c . Наприклад для методу третин $c=a+(b-a)/3$, а для золоті пропорції $c=a+\alpha(b-a)/\alpha$, де $\alpha=(1+\sqrt{5})/2\approx 1.618$ є коефіцієнтом золотого січення. Перевагами цих методик є ймовірність більш швидкого звуження інтервалу з коренем, але недолік пов'язаний з цією перевагою — також є ймовірність менш швидкого звуження інтервалу від методу половинного ділення. Також загалом не можна передбачити швидкість звуження інтервалу з коренем, тому потрібно слідкувати за шириною проміжка на кожному кроці звуження. Функція пошуку коренів цими методами матиме вигляд:

```
float KorFind(float (*f)(float x),float a, float b,
float delta)
{
    int n=0; //кількість циклів уточнення
    float c; //третина/частина пробного інтервалу
do//Повторюємо звуження інтервалу
    {
        c=a+(b-a)/3;//Шукаємо третину інтервалу
        //c=a+(b-a)*2.0/(1+sqrt(5)); //Для зол.проп.
40
        if( f(c)>0.0 )//перевірка знаку функції в
середині інтервалу
            { //Звуження зправа
                b=c;
            }else
            { //Звуження інтервалу зліва
                a=c;
            }
        n++; //рахуємо кількість уточнень
    }while(fabs(b-a)>delta);//кожного циклу ширину
        //інтервалу з коренем
        //порівнюємо з допустимою
        //похибкою
    return (a+b)*0.5; //повернемо як корінь середину
останнього інтервалу
}
```

На цьому лістингу програми показано, що перевірку довжини інтервалу, в якому лежить корінь, потрібно повторювати після кожного звуження.

2. Методи послідовного наближення

Пошук коренів є досить ресурсоємною задачею, тому дослідники

намагаються винайти засоби більш швидкого наближення до коренів і тут зменшення похибки вдвічі на кожному уточненні є поганим результатом. Особливо важливим є таке вдосконалення для ручних розрахунків та написання зверху продуктивних програмних продуктів.

2.1 *Метод хорд* уточнення коренів на інтервалі використовував ще Ньютон. Метод полягає на заміні функції на інтервалі пошуку кореня прямою, що дозволяє досить точно отримати наступне звуження інтервалу. Пояснення алгоритму уточнення кореню показано на рис. 4.2.

На рисунку показано фрагмент графіку $\cos(x)-x$ з початковим виділенням коренів $[0;0.8]$. Замість визначення середньої точки інтервалу для подальшого уточнення, чи поділу в наперед зазначеної пропорції, точки на графіку $f(0)=1$ та $f(0.8)=-0.103293$ з'єднуються прямою. Пряма перетинає вісь ОХ при $x=0.725105$, $f(0.725105)=0.023325>0$. Шлях вибору границь наступного інтервалу, де є корінь, є аналогічним методу дихотомії, і наступним інтервалом для подальшого пошуку кореня є $[0.725105;0.8]$. Довжина отриманого інтервалу порівняно з попереднім менша в $0.8/(0.8-0.725105)\approx 10$ раз, а метод половинного ділення зменшує проміжок лише у 2 рази. Звісно, приклад демонструє вдалу ситуацію, але й в загальному випадку метод хорд не гірший за метод половинного ділення.

Методи розв'язання систем нелінійних рівнянь

У завданнях проектування та дослідження поведінки реальних об'єктів, процесів і систем (ОПС) математичні моделі повинні відображати реальні фізичні нелінійні процеси. При цьому ці процеси залежать, як правило, від багатьох змінних.

В результаті математичні моделі реальних ОПС описуються системами нелінійних рівнянь.

Рішення систем нелінійних рівнянь

Дана система нелінійних рівнянь

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0, \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0, \end{cases} \quad (10.1)$$

або $f_i(x_1, x_2, x_3, \dots, x_n) = 0, i = \overline{1 \dots n}$.

Необхідно вирішити цю систему, тобто знайти вектор $\bar{X} = [x_1, x_2, x_3, \dots, x_n]$, що задовольняє системі (10.1) з точністю ε .

Вектор визначає точку в n -вимірному Евклідовому просторі, тобто $\bar{X} \in$ цьому простору і задовольняє всім рівнянням системи (10.1).

На відміну від систем лінійних рівнянь для систем нелінійних рівнянь невідомі прямі методи рішення. При вирішенні систем нелінійних рівнянь використовуються ітераційні методи.

Ефективність всіх ітераційних методів залежить від вибору початкового наближення

(початкової точки), тобто вектора $\bar{X}^0 = [x_1^0, x_2^0, \dots, x_n^0]$

Область, в якій початкове наближення \bar{X}^0 сходиться до шуканого рішення, називається областю збіжності G . Якщо початкове наближення лежить за межами G , то рішення системи отримати не вдається.

Вибір початкової точки \bar{X}^0 в чому визначається інтуїцією та досвідом фахівця.

Метод простих ітерацій

Для застосування цього методу вихідна система (10.1) повинна бути перетворена до виду

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, x_3, \dots, x_n), \\ x_2 = \varphi_2(x_1, x_2, x_3, \dots, x_n), \\ \dots\dots\dots \\ x_n = \varphi_n(x_1, x_2, x_3, \dots, x_n), \end{cases} \quad (10.2)$$

або $x_i = \varphi_i(x_1, x_2, x_3, \dots, x_n), i = \overline{1, n}$.

Далі, вибравши початкове наближення $\overline{X^0} = [x_1^0, x_2^0, \dots, x_n^0]$ і використовуючи систему (10.2), будемо ітераційний процес пошуку за схемою:

$$x_i^k = \varphi_i(x_1^{k-1}, x_2^{k-1}, x_3^{k-1}, \dots, x_n^{k-1}),$$

тобто на кожному k-му кроці пошуку вектор змінних \overline{X} знаходимо, використовуючи значення змінних, отриманих на кроці (k-1).

Ітераційний процес пошуку припиняється як тільки виконається умова

$$|x_j^k - x_j^{k-1}| \leq \varepsilon, j = \overline{1, n}. \quad (10.3)$$

При цьому умова (10.3) повинно виконуватися одночасно по всім змінним.

Метод простих ітерацій використовується для вирішення таких систем лінійних рівнянь, в яких виконується умова збіжності ітераційного процесу пошуку, а саме:

$$\sum_{i=1}^n \left| \frac{\delta \varphi_i}{\delta x_j} \right| < 1, j = \overline{1, n}. \quad (10.4)$$

тобто сума абсолютних величин приватних похідних всіх перетворених рівнянь системи (10.2) по j-ой змінної менше одиниці.

На малюнку 10.1 представлена схема алгоритму рішення систем нелінійних рівнянь методом простих ітерацій.

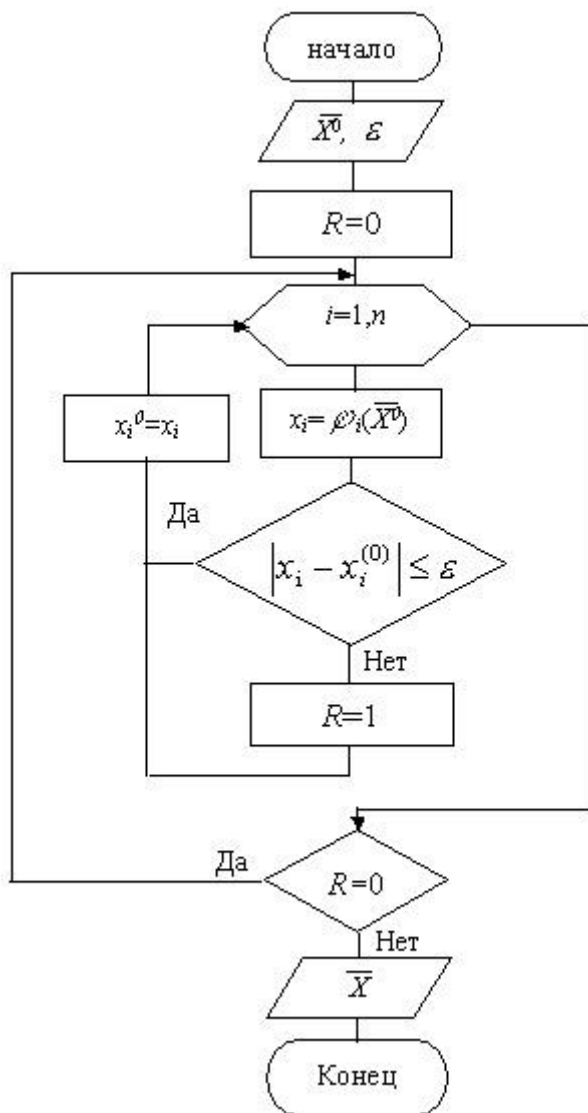


Рис. 10.1. Схема алгоритму методу простих ітерацій

Розглянемо приклад.

Дана система нелінійних рівнянь:

Необхідно визначити область збіжності системи, вибрати початкову точку і знайти одне з рішень системи.

1. Будуємо графіки рівнянь:

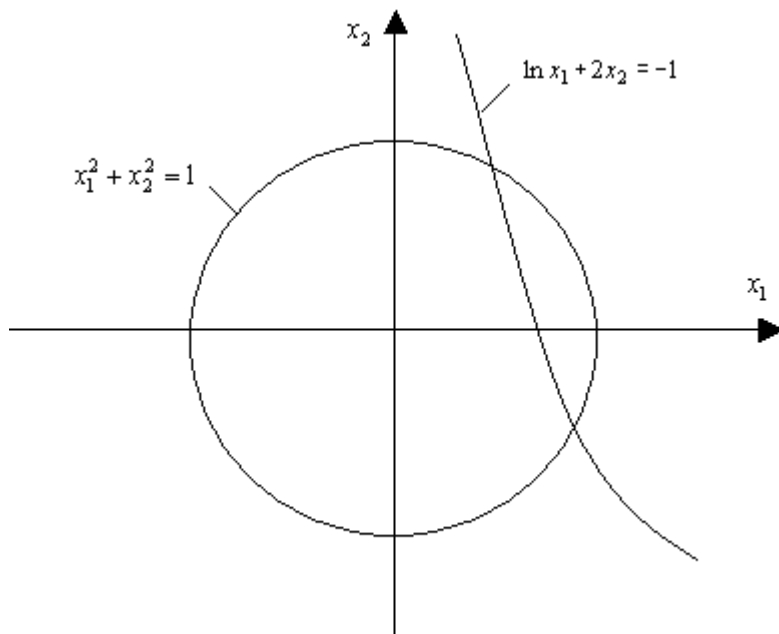


Рис. 2.

2. Перетворимо систему для вирішення методом ітерацій

$$\begin{cases} x_1 = \sqrt{1 - x_2^2} \rightarrow \varphi_1(x_1, x_2), \\ x_2 = -0,5 - 0,5 \ln x_1 \rightarrow \varphi_2(x_1, x_2). \end{cases}$$

Перевіряємо умову збіжності (10.4). Для заданої системи воно має вигляд:

$$\begin{cases} |\delta\varphi_1/\delta x_1| + |\delta\varphi_2/\delta x_1| < 1 \\ |\delta\varphi_1/\delta x_2| + |\delta\varphi_2/\delta x_2| < 1 \end{cases}$$

Знаходимо:

$$\begin{aligned} \delta\varphi_1/\delta x_1 &= 0; \\ \delta\varphi_1/\delta x_2 &= -x/\sqrt{1-x^2}; \\ \delta\varphi_2/\delta x_1 &= -1/2x_1; \\ \delta\varphi_2/\delta x_2 &= 0 \end{aligned}$$

В результаті умова (10.4) буде мати вигляд:

Визначаємо область збіжності G.

Межа області збіжності визначиться при вирішенні системи,

$$\begin{cases} 1/2x_1 = 1; \\ x_2/\sqrt{1-x_2^2} = 1. \end{cases}$$

Звідси $x_1 = 0,5$; $x_2 = \pm\sqrt{0,5}$

В результаті область збіжності визначиться при $|x_1| \geq 0,5$ і $-0,707 \leq x_2 \leq 0,707$.

На графіку рівнянь будуюмо область збіжності G:

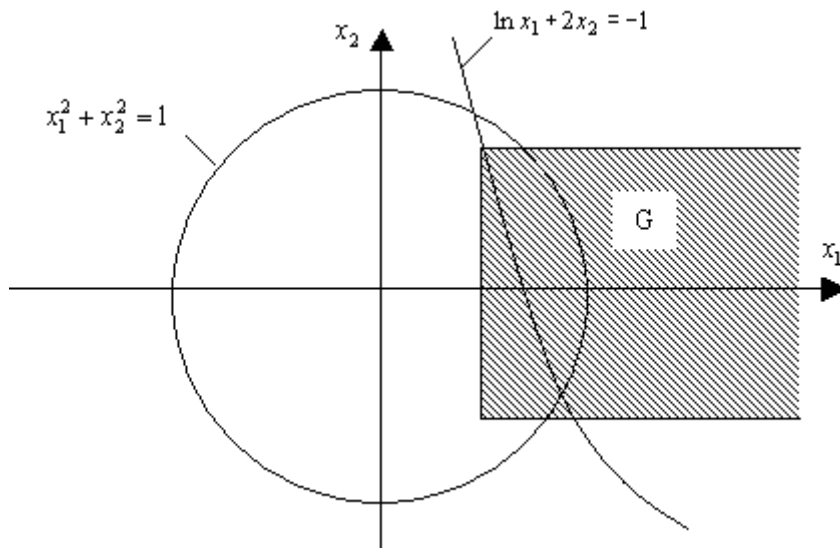


Рис. 3.

Вибираємо початкову точку $\overline{X^0} = [0.8; -0.6]$, що належить області збіжності G.

Використовуючи вибрану початкову точку $\overline{X^0} = [0.8; -0.6]$ вирішуємо задану систему нелінійних рівнянь.

Рекомендована література

1. Демидович В.П., Марон И.А. Основы вычислительной математики.- М.:Физматгиз, 1960
2. Бахвалов Н.С. Численные методы, т.1 – М.:Наука, 1978
3. Калиткин Н.Н. Численные методы. – М.:Наука, 1978
4. Положий Г.Н., Пахарева Н.А. и др. – Математический практикум. М.:Физматгиз, 1960
5. Хемминг Р. Численные методы. – М.:Наука, 1978
6. Березин И.С., Жидков И.П. Методы вычислений, т.т. 1,2 - М.:Наука, 1977
7. Мак-Крекен Д., Дорн У. Численные методы и программирование на Фортране, М. , 1977
8. Ю.М. Зорін. Конспект лекцій з дисципліни “Алгоритми та методи обчислень” – [http:// yzorin.ho.com.ua](http://yzorin.ho.com.ua).

Допоміжна

1. Волков Е.А. Численные методы. – М.:Наука, 1982
2. Кунцман Ж. Численные методы. – М.:Наука, 1979
3. Воеводин В.В. Вычислительные основы линейной алгебры. - М.:Наука, 1977

Інформаційні ресурси Бібліотеки, Інтернет, електронні книги.