Міністерство освіти та науки України
Національний університет “Львівська політехніка”

СИСТЕМИ НЕЛІНІЙНИХ РІВНЯНЬ. МЕТОД НЬЮТОНА ТА ε-АЛГОРИТМ

Інструкція до лабораторної роботи № 3
з курсу “Чисельні методи”

для студентів спеціальності
122 “Комп’ютерні науки та інформаційні технології”
спеціалізації “Системна інженерія (інтернет речей)”

Затверджено
на засіданні кафедри
“Комп’ютеризовані
системи автоматики”
Протокол № 1 від 30.08.2017

Львів 2017

Системи нелінійних рівнянь. Метод Ньютона та ε-алгоритм: Інструкція до лабораторної роботи № 3 з курсу “Чисельні методи” для студентів спеціальності 122 “Комп’ютерні науки та інформаційні технології” спеціалізації “Системна інженерія (інтернет речей)”/Укл.: У.Ю. Дзелендзяк, А.Г. Павельчак– Львів: НУЛП, 2017. – 40 с.

Укладачі: У.Ю. Дзелендзяк, к.т.н., доцент
А.Г. Павельчак, к.т.н., доцент

Відповідальний за випуск:
А.Й. Наконечний, д.т.н., професор

Рецензент: З.Р. Мичуда, д.т.н., професор

Мета роботи: ознайомитися з найпоширенішим ітераційним методом розв'язування систем нелінійних рівнянь – методом Ньютона та екстраполяційним методом – ε -алгоритмом.

1. Системи нелінійних рівнянь

1.1. Постановка задачі.

Нехай для обчислення невідомих $x_1, x_2, \ldots, x_n$ необхідно розв'язати систему n нелінійних рівнянь

$$\left.\begin{array}{l} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{array}\right\}. \qquad (1.1)$$

У векторній формі система (1.1) записується так:

$$F(X) = 0, \qquad (1.2)$$

де $F = [f_1, f_2, \ldots, f_n]^T$, $X = [x_1, x_2, \ldots, x_n]^T$.

Ця задача є значно складнішою, аніж розглянута в лабораторній роботі № 3 задача пошуку коренів нелінійного рівняння з одним невідомим. Але й на практиці вона зустрічається значно частіше, тому що в реальних дослідженням інтерес представляють системи з багатьма параметрами (іноді їх число сягає сотень чи тисяч). Знаходження точного розв'язку системи (1.1), тобто вектора $\overline{X}$, практично неможливо. Лише в окремих випадках систему (1.1) можна розв'язати безпосередньо. Наприклад, для випадку двох рівнянь іноді вдається виразити одну невідому через іншу й, таким чином, звести задачу до пошуку кореня одного нелінійного рівняння. Тому розв'язок нелінійних систем шукають переважно лише ітераційними методами. Найпоширеніші серед них: метод простої ітерації, метод Зейделя та метод Ньютона.

Важливим аспектом при розв'язуванні систем нелінійних рівнянь є усвідомлення того, що система може й не мати розв'язку, а у випадку його існування – кількість розв'язків може бути довільною. У загальному випадку складно визначити чи має система розв'язки та скільки їх.

Пояснимо сказане на прикладі такої системи рівнянь

$$\left.\begin{aligned} 4x_1^{\,2} + x_2^{\,2} - 4 = 0 \\ x_1 - x_2^{\,2} + t = 0 \end{aligned}\right\} \qquad (1.3)$$

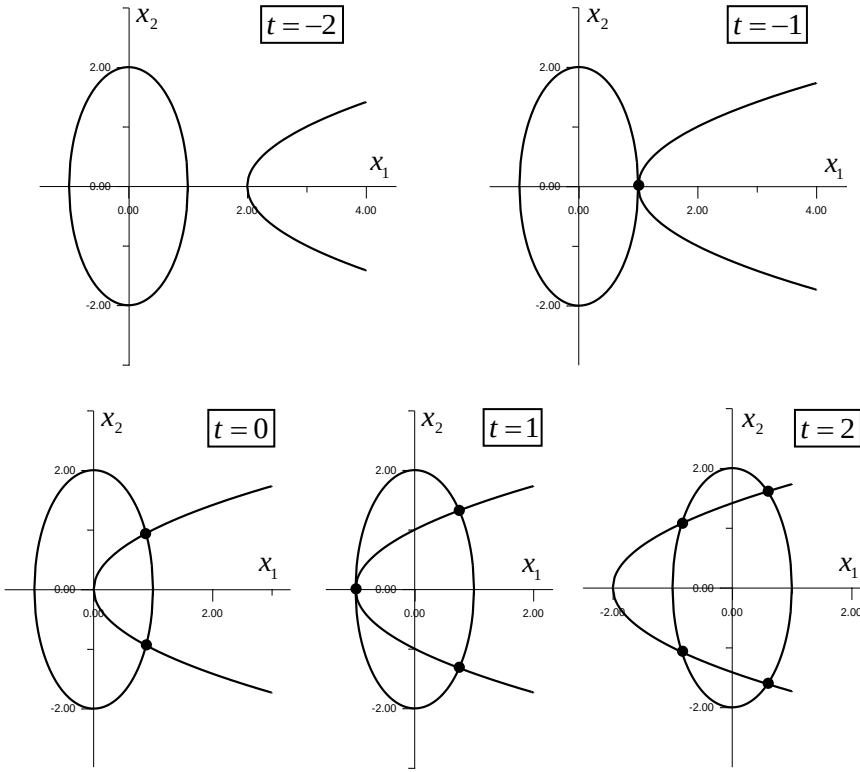


Рис.1. Сукупність розв'язків системи рівнянь (1.3)

Перше рівняння системи (1.3) задає на площині x_1Ox_2 еліпс, друге рівняння – параболу. Координати крапок перетину цих кривих є розв'язками системи. Якщо значення параметра t змінюється від -2 до 2, то можливі такі варіанти (рис. 1): $t=-2$ – система розв'язку немає; $t=-1$ – єдиний розв'язок; $t=0$ – два розв'язки; $t=1$ – три розв'язки; $t=2$ – чотири розв'язки.

На рис. 2 зображено трьохмірну ілюстрацію графічного методу пошуку розв'язків системи нелінійних рівнянь (1.3) при $t=0$. Таким чином, для цієї системи рівнянь будуються поверхні функцій f_1, f_2

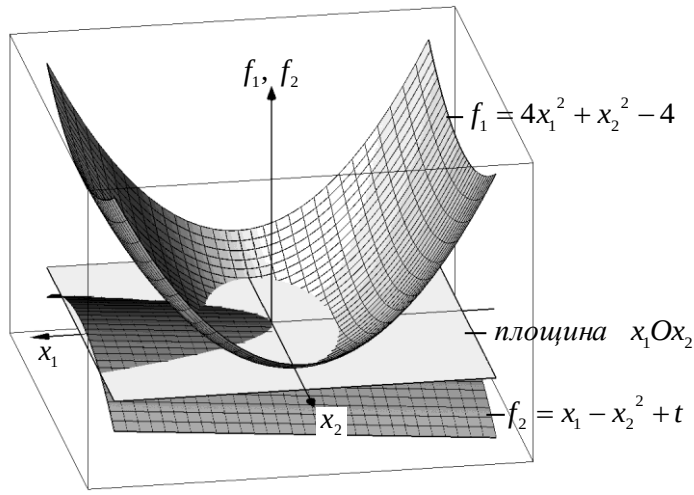


Рис.2. 3-D ілюстрація розв'язків системи рівнянь (1.3) при $t = 0$

при широкому діапазонові змін x_1, x_2 та шукаються крапки перетину цих поверхонь з площиною x_1Ox_2. Перетини цих функцій на площині x_1Ox_2 і дадуть шукані розв'язки системи нелінійних рівнянь.

1.2. Основні етапи розв'язування.

Як і у випадку рівняння з одним невідомим розв'язування нелінійної системи рівнянь здійснюється у три етапи:

а) локалізація коренів та вибір початкових наближень X_0;

б) ітераційне уточнення коренів;

в) перевірка умови збіжності ітераційного процесу.

На етапі локалізації для кожного із розв'язків $\overline{X}$ вказують множину, що містить лише цей розв'язок та розміщена в достатньо малій його околиці. Часто в якості такої множини вибирають паралелепіпед чи кулю в n-мірному просторі.

Іноді етап локалізації не створює труднощів: відповідні множини можуть бути задані, визначені з фізичних міркувань, зі змісту параметрів $x_1, x_2, \ldots, x_n$ чи бути відомими з досвіду розв'язування подібних задач. Однак найчастіше задача локалізації (особливо при великій розмірності n) представляє собою складну проблему, від успішного вирішення якої в основному і залежить

можливість пошуку розв'язку системи (1.1). Тому кваліфікація дослідника, розуміння поставленої задачі та досвід розв'язування подібних задач відіграють тут неабияку роль. У більшості випадків етап локалізації вважають завершеним при вдалому виборі початкового наближення $X^{(0)}$.

На другому етапі для пошуку розв'язку із заданою точністю ε використовують один з ітераційних методів розв'язування нелінійних систем.

Іноді при розв'язуванні нелінійних систем рівнянь вживають додаткових запобіжних заходів. По-перше, щоб мати гарантію, що на кожному кроці ітерацій відбувається наближення до шуканого розв'язку, а по-друге, щоб запобігти використанню великих кроків, які можуть призвести до аварії. Збіжність розв'язку визначається за допомогою значень нелінійних функцій на суміжних ітераціях:

$$\left|f_i\left(X^{(k+1)}\right)\right| < \left|f_i\left(X^{(k)}\right)\right|, \text{ для } i=\overline{1,n}. \qquad (1.3)$$

З метою запобігання великих кроків $p_i = x_i^{(k+1)} - x_i^{(k)}$, для $i=\overline{1,n}$ накладають обмеження

$$|p_i| \le \delta, \text{ для } i=\overline{1,n}, \qquad (1.4)$$

де δ – вибраний обмежувач. Якщо модель хороша, то апроксимація буде ефективною при великих значеннях p_i і тому можна вибрати великий обмежувач δ. Якщо модель погано обумовлена, то апроксимація буде прийнятною лише при малих значеннях p_i і тоді необхідно вибрати мале значення δ.

2. Метод Ньютона

Узагальнимо метод Ньютона, викладений в інструкції до лабораторної роботи № 3 для уточнення кореня одного нелінійного рівняння, на розв'язування системи нелінійних рівнянь (1.1).

Нехай наближені значення системи (1.1), отримані на попередній ітерації, дорівнюють відповідно $x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}$. Задача полягає в знаходженні приростів (поправок) до цих значень $\Delta x_1, \Delta x_2, \ldots, \Delta x_n$, завдяки яким наступне наближення до розв'язку системи (1.1) записується у вигляді

$$\begin{aligned} x_1^{(k)} &= x_1^{(k-1)} + \Delta x_1 \\ x_2^{(k)} &= x_2^{(k-1)} + \Delta x_2 \\ \dots \quad \dots \quad \dots \quad \dots \\ x_n^{(k)} &= x_n^{(k-1)} + \Delta x_n \end{aligned} . \qquad (2.1)$$

Здійснимо розклад лівих частин рівнянь (1.1) у ряд Тейлора, обмеживши лише лінійними членами відносно приростів (відкинувши члени, що містять другі та вищі похідні)

$$\left.\begin{aligned} f_1\left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right) &\approx f_1^{(k-1)} + \frac{\partial f_1}{\partial x_1}^{(k-1)} \Delta x_1 + \dots + \frac{\partial f_1}{\partial x_n}^{(k-1)} \Delta x_n \\ f_2\left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right) &\approx f_2^{(k-1)} + \frac{\partial f_2}{\partial x_1}^{(k-1)} \Delta x_1 + \dots + \frac{\partial f_2}{\partial x_n}^{(k-1)} \Delta x_n \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ f_n\left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right) &\approx f_n^{(k-1)} + \frac{\partial f_n}{\partial x_1}^{(k-1)} \Delta x_1 + \dots + \frac{\partial f_n}{\partial x_n}^{(k-1)} \Delta x_n \end{aligned}\right\} \qquad (2.2)$$

У правих частинах співвідношень (2.2) значення функцій $f_1, f_2, \dots, f_n$ та їх похідні обчислюються в крапці $X^{(k-1)} = \left[x_1^{(k-1)}, x_2^{(k-1)}, \dots, x_n^{(k-1)}\right]$.

Оскільки у відповідності з (1.1) ліві частини (2.2) повинні перетворюватися в нуль, прирівняємо до нуля і праві частини, тобто знайдемо нове наближення з умови рівності нулю розкладу функцій f_i. У результаті отримаємо таку систему лінійних алгебричних рівнянь відносно приростів

$$\left.\begin{aligned} f_1^{(k-1)} + \frac{\partial f_1}{\partial x_1}^{(k-1)} \cdot \Delta x_1 + \dots + \frac{\partial f_1}{\partial x_n}^{(k-1)} \cdot \Delta x_n = 0 \\ f_2^{(k-1)} + \frac{\partial f_2}{\partial x_1}^{(k-1)} \cdot \Delta x_1 + \dots + \frac{\partial f_2}{\partial x_n}^{(k-1)} \cdot \Delta x_n = 0 \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ f_n^{(k-1)} + \frac{\partial f_n}{\partial x_1}^{(k-1)} \cdot \Delta x_1 + \dots + \frac{\partial f_n}{\partial x_n}^{(k-1)} \cdot \Delta x_n = 0 \end{aligned}\right\} \qquad (2.3)$$

або в матричній формі

$$F^{(k-1)} + \frac{\partial F^{(k-1)}}{\partial X}\left(X^{(k)} - X^{(k-1)}\right) = 0, \qquad (2.4)$$

де $\Delta X = X^{(k)} - X^{(k-1)}$.

Виразимо з рівняння (2.4) $X^{(k)}$ та отримаємо ітераційну формулу методу Ньютона:

$$X^{(k)} = X^{(k-1)} - \left(J^{(k-1)}\right)^{-1} \cdot F^{(k-1)}, \tag{2.5}$$

де

$$J = \frac{\partial F}{\partial X} = \begin{bmatrix} \dfrac{\partial f_1(x_1, \ldots, x_n)}{\partial x_1} & \dfrac{\partial f_1(x_1, \ldots, x_n)}{\partial x_2} & \cdots & \dfrac{\partial f_1(x_1, \ldots, x_n)}{\partial x_n} \\ \dfrac{\partial f_2(x_1, \ldots, x_n)}{\partial x_1} & \dfrac{\partial f_2(x_1, \ldots, x_n)}{\partial x_2} & \cdots & \dfrac{\partial f_2(x_1, \ldots, x_n)}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial f_n(x_1, \ldots, x_n)}{\partial x_1} & \dfrac{\partial f_n(x_1, \ldots, x_n)}{\partial x_2} & \cdots & \dfrac{\partial f_n(x_1, \ldots, x_n)}{\partial x_n} \end{bmatrix} \tag{2.6}$$

матриця Якобі. Метод Ньютона передбачає обертання матриці Якобі.

Наведемо геометричну ілюстрацію методу Ньютона для розв'язування системи рівнянь на прикладі системи (1.3) при $t = 0$. Для цього запишемо рівняння дотичних площин до функцій у крапці $A = [a_1, a_2]$

$$z_i - f_i(a_1, a_2) = \sum_{j=1}^{2} J_{ij}(a_1, a_2) \cdot \left(x_j - a_j\right), \text{ для } i = 1,2. \tag{2.7}$$

Матриця Якобі (2.6) для системи (1.3) має такий вигляд

$$J = \begin{bmatrix} 8a_1 & 2a_2 \\ 1 & -2a_2 \end{bmatrix}. \tag{2.8}$$

Відповідно рівняння дотичних площин у розгорнутому вигляді будуть мати такий вигляд

$$z_1 = 8a_1(x_1 - a_1) + 2a_2(x_2 - a_2) + f_1(a_1, a_2)$$

$$z_2 = 1(x_1 - a_1) - 2a_2(x_2 - a_2) + f_2(a_1, a_2)$$

або

$$\begin{aligned} z_1 &= 8a_1(x_1 - a_1) + 2a_2(x_2 - a_2) + 4a_1^{\,2} + a_2^{\,2} - 4 \\ z_2 &= 1(x_1 - a_1) - 2a_2(x_2 - a_2) + a_1 - a_2^{\,2} \end{aligned} \tag{2.9}$$

На рис. 3 зображений випадок невдалого вибору початкового наближення $X^{(0)} = [0, 0]$, коли Метод Ньютона закінчується аварією. Дотична площина до функції f_1 не має крапок перетину з площиною x_1Ox_2.

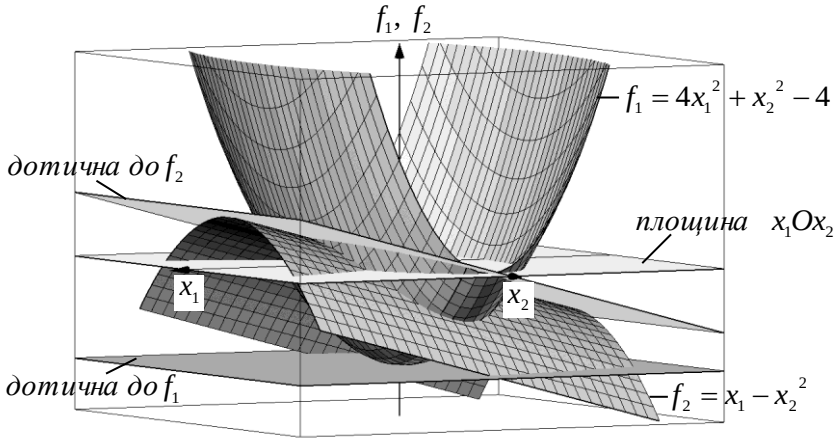


Рис.3. Наближення методу Ньютона при $X^{(0)} = [0, 0]$

Виберемо інше початкове наближення $X^{(0)} = [0{,}5;\ 0{,}5]$, зміщене до одного з розв'язків системи.

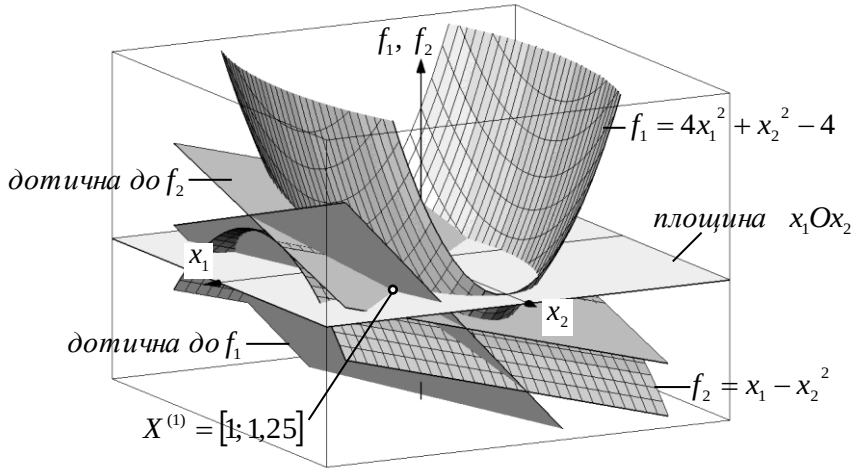


Рис.4. Наближення $X^{(1)}$ методу Ньютона при $X^{(0)} = [0{,}5;\ 0{,}5]$

Перше наближення $X^{(1)} = [1; 1{,}25]$ методу Ньютона (рис. 4) відповідає крапці перетину між собою усіх трьох площин: двох дотичних площин до функцій f_1, f_2 у крапці $X^{(0)} = [0{,}5; 0{,}5]$ та площини x_1Ox_2.

На рис. 5 проілюстровано наступну ітерацію методу Ньютона, на якій отримано наближення $X^{(2)} = [0{,}8889; 0{,}9806]$. Як бачимо, це наближення приблизилося до шуканого нами одного з розв'язків системи рівнянь.

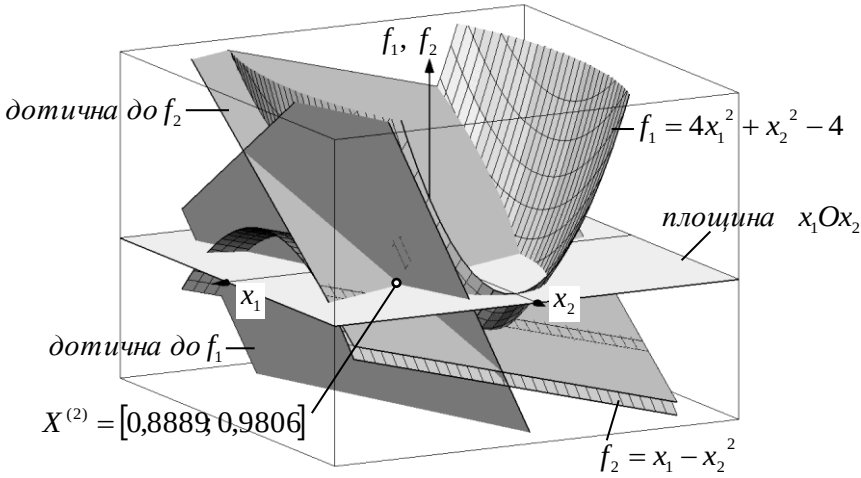


Рис.5. Наближення $X^{(2)}$ методу Ньютона при $X^{(1)} = [1; 1{,}25]$

Збіжність методу Ньютона забезпечується в околиці розв'язку $\overline{X}$ системи рівнянь (1.1) при умові що функції системи $f_i\left(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\right)$, для $i = \overline{1,n}$, є двічі неперервано диференційовані, а матриця Якобі цієї системи не вироджена і її детермінант не дорівнює нулю. Випадок, коли детермінант дорівнював нулю, був проілюстрований на рис. 3.

Ітераційний процес методу зупиняють при виконанні умови

$$\left|x_i^{(k)} - x_i^{(k-1)}\right| < \varepsilon, \text{ для } i = \overline{1,n}. \qquad (2.10)$$

Зазначимо, що при обертанні матриці Якобі необхідно використовувати прямі методи розв'язування лінійних алгебричних рівнянь, оскільки ця матриця на кожному кроці методу змінюється, і тому не можна гарантувати, що який-небудь ітераційний метод буде

завжди збігатися. Рекомендується використовувати метод Гауса з вибором головного елемента.

Загальний алгоритм методу Ньютона

Ініціалізація одиничної матриці:

для $i, j = \overline{1,n}$

якщо $i = j \ \rightarrow \{E_{ij} = 1\}$

інакше $\{E_{ij} = 0\}$

Ітераційний процес методу:

Обчислення вектора функцій f_i та матриці Якобі J_{ij}:

$$f_1 = f_1(x_1, \dots x_n); \ \dots \ f_n = f_n(x_1, \dots x_n)$$

$$J_{11} = J_{11}(x_1, \dots x_n); \ \dots \ \dots \ J_{nn} = J_{nn}(x_1, \dots x_n)$$

для $b = \overline{1,n}$

Обертання матриці Якобі:

Метод Гауса

Прямий хід:

для $i, j = \overline{1,n}$

$V_{ij} = J_{ij}$

$P_i = E_{ib}$

далі згідно алгоритму методу Гауса

$\vdots$

для $i = \overline{1,n}$

$INVERS_{ib} = X_i$

для $i = \overline{1,n}$

$$x_i = x_{OLD\ i} - \sum_{j=1}^{n} INVERS_{ij} \cdot f_j$$

Умова збіжності:

для $i = \overline{1,n}$

$$\left| \frac{x_i - x_{OLD\ i}}{x_i} \right| \cdot 100\% < \varepsilon ; \qquad x_{OLD\ i} = x_i$$

Опис алгоритму

1. На початку алгоритму задаємо початкове наближення $x_i = x_{OLD\,i} = x_i^{(0)}$, для $i = \overline{1,n}$, та відносну похибку ε у відсотках.
2. Встановлюємо значення для одиничної матриці E.
3. Обчислюємо значення елементів вектора функцій f_i та елементів матриці Якобі J_{ij}.
4. Здійснюємо обертання матриці Якобі J_{ij} методом Гауса (з вибором головного елемента по рядку, стовпцю чи по всій матриці). Алгоритм обертання матриці методом Гауса наведено в інструкції до лабораторної роботи № 2. Єдиною модифікацією є $V_{ij} = J_{ij}$ (замість базової матриці A_{ij} беремо J_{ij}).
5. Згідно ітераційної формули обчислюємо наступне наближення $x_i^{(k)}$, для $i = \overline{1,n}$.
6. Здійснюємо перевірку умови збіжності. Якщо вона не виконується, то процес уточнення повторюємо (п.3).
7. Для перевірки вірності роботи алгоритму підставляємо наші знайдені значення $x_1, x_2, \ldots, x_n$ в систему рівнянь $F(x) = 0$. Значення функцій f_i, для $i = \overline{1,n}$, мають бути близькими нулю, в залежності від вибраного значення ε.

3. Модифікації методу Ньютона

3.1. Спрощений метод Ньютона (метод паралельних площин)

Ідея спрощеного методу полягає у тому, що обчислення матриці Якобі J_{ij} та її наступне обертання здійснюємо лише один раз на першій ітерації, а далі використовуємо обернену матрицю Якобі з тими самими значеннями. В результаті отримаємо таку розрахункову формулу для спрощеного методу Ньютона

$$X^{(k)} = X^{(k-1)} - \left(J^{(0)}\right)^{-1} \cdot F^{(k-1)}. \qquad (3.1)$$

У геометричному змісті пошук розв'язку системи здійснюється за допомогою площин паралельних дотичним площинам до функцій системи у крапці початкового наближення $X^{(0)} = \left[x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right]$.

Загальний алгоритм спрощеного методу Ньютона

Ініціалізація одиничної матриці:

для $i, j = \overline{1,n}$

якщо $i = j \ \rightarrow \{E_{ij} = 1\}$

інакше $\{E_{ij} = 0\}$

Обчислення матриці Якобі J_{ij}:

$J_{11} = J_{11}(x_1, \dots x_n); \ \dots \ \dots \ J_{nn} = J_{nn}(x_1, \dots x_n)$

для $b = \overline{1,n}$

Обертання матриці Якобі:

Метод Гауса

Прямий хід:

для $i, j = \overline{1,n}$

$V_{ij} = J_{ij}$

$P_i = E_{ib}$

далі згідно алгоритму методу Гауса

⋮

для $i = \overline{1,n}$

$INVERS_{ib} = X_i$

Ітераційний процес методу:

Обчислення вектора функцій f_i

$f_1 = f_1(x_1, \dots x_n); \ \dots \ f_n = f_n(x_1, \dots x_n)$

для $i = \overline{1,n}$

$$x_i = x_{OLD\ i} - \sum_{j=1}^{n} INVERS_{ij} \cdot f_j$$

Умова збіжності:

для $i = \overline{1,n}$

$$\left| \frac{x_i - x_{OLD\ i}}{x_i} \right| \cdot 100\% < \varepsilon ; \qquad x_{OLD\ i} = x_i$$

Опис алгоритму

1. На початку алгоритму задаємо початкове наближення $x_i = x_{OLD\ i} = x_i^{(0)}$, для $i = \overline{1,n}$, та відносну похибку ε у відсотках.
2. Встановлюємо значення для одиничної матриці E.
3. Обчислюємо значення елементів матриці Якобі J_{ij}.
4. Здійснюємо обертання матриці Якобі J_{ij} методом Гауса (з вибором головного елемента по рядку, стовпцю чи по всій матриці). Алгоритм обертання матриці методом Гауса наведено в інструкції до лабораторної роботи № 2. Єдиною модифікацією є $V_{ij} = J_{ij}$ (замість базової матриці A_{ij} беремо J_{ij}).
5. Обчислюємо значення елементів вектора функцій f_i.
6. Згідно ітераційної формули обчислюємо наступне наближення $x_i^{(k)}$, для $i = \overline{1,n}$.
7. Здійснюємо перевірку умови збіжності. Якщо вона не виконується, то процес уточнення повторюємо (п.5).
8. Для перевірки вірності роботи алгоритму підставляємо наші знайдені значення $x_1, x_2, \ldots, x_n$ в систему рівнянь $F(x) = 0$. Значення функцій f_i, для $i = \overline{1,n}$, мають бути близькими нулю, в залежності від вибраного значення ε.

3.2. Метод Ньютона з кінцево-різницевою матрицею Якобі

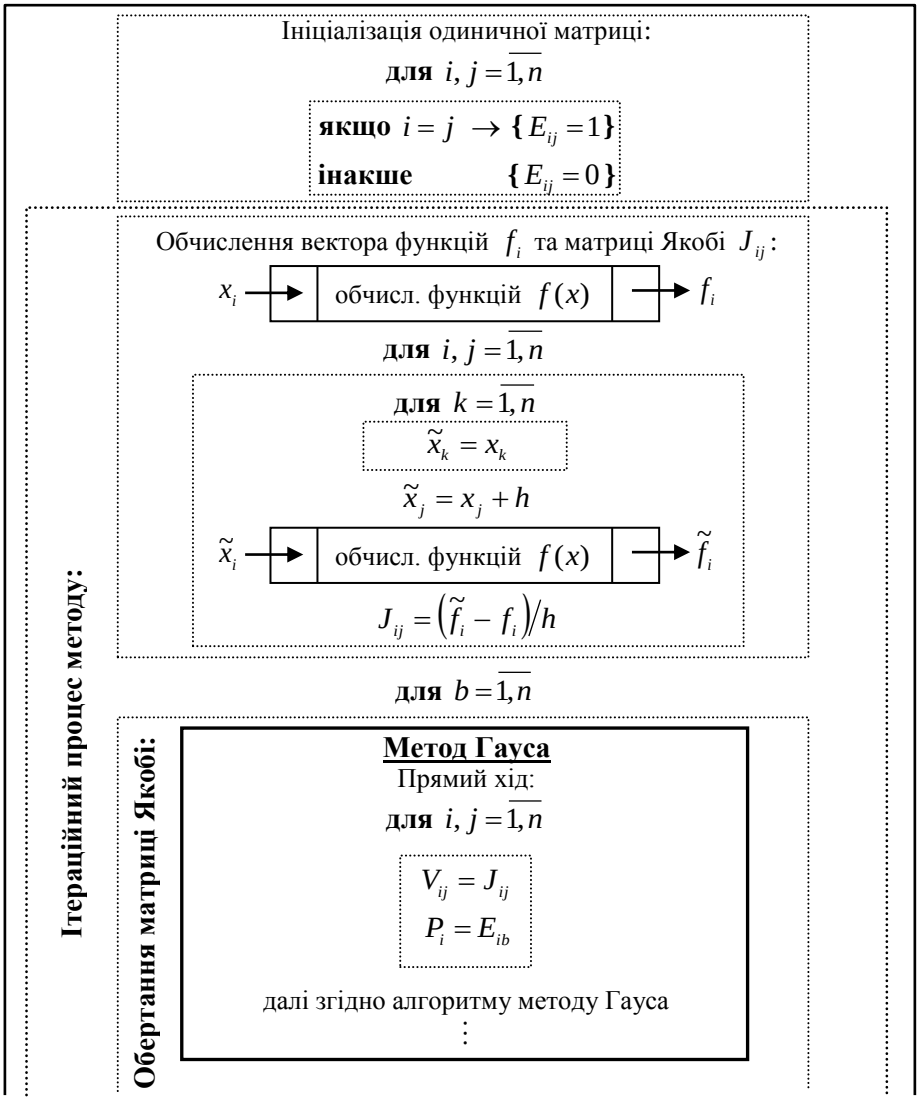
Часто на практиці обчислення часткових похідних матриці Якобі представляє собою доволі складну, а іноді й нездійсненну задачу. В такому випадку для наближеного обчислення часткових похідних можна спробувати використати формули числового диференціювання. Наприклад, застосувати таку кінцево-різницеву апроксимацію похідної у крапці $X^{(k)} = \left[x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\right]$:

$$J_{ij}(x_1^{(k)},\ldots,x_n^{(k)}) = \frac{\partial f_i}{\partial x_j}\left(x_1^{(k)},\ldots,x_n^{(k)}\right) \approx \frac{f_i(x_1^{(k)},\ldots,x_j^{(k)} + h_j^{(k)},\ldots,x_n^{(k)}) - f_i(x_1^{(k)},\ldots,x_n^{(k)})}{h_j^{(k)}}. \quad (3.2)$$

У найпростішому випадку цього методу кроки $h_j^{(k)}$, для $j = \overline{1,n}$, не залежать від k. Зазначимо, що вибір величини кроків представляє собою не просту задачу. З однієї сторони, вони повинні бути

достатньо малими, щоб матриця J_{ij} мала добре наближення, з іншої, вони не можуть бути дуже малими, так як вплив похибок обчислень функцій f_i на похибку формули (3.2) стає катастрофічним (виконується обчислення близьких наближених чисел).

Загальний алгоритм методу Ньютона
з кінцево-різницевою матрицею Якобі



для $i=\overline{1,n}$

$INVERS_{ib} = X_i$

для $i=\overline{1,n}$

$$x_i = x_{OLD\ i} - \sum_{j=1}^{n} INVERS_{ij} \cdot f_j$$

Умова збіжності:

для $i=\overline{1,n}$

$$\left|\frac{x_i - x_{OLD\ i}}{x_i}\right| \cdot 100\% < \varepsilon\ ; \quad x_{OLD\ i} = x_i$$

Опис алгоритму

1. На початку алгоритму задаємо початкове наближення $x_i = x_{OLD\ i} = x_i^{(0)}$, для $i=\overline{1,n}$, та відносну похибку ε у відсотках.
2. Встановлюємо значення для одиничної матриці E.
3. Обчислюємо значення елементів вектора функцій f_i та елементів матриці Якобі J_{ij} з кінцевих різниць. Обчислення функцій виконуються у зовнішній програмній процедурі, у яку як вхідні дані поступає вектор невідомих, а вихідними даними служить обчислений вектор функцій.
4. Здійснюємо обертання матриці Якобі J_{ij} методом Гауса (з вибором головного елемента по рядку, стовпцю чи по всій матриці). Алгоритм обертання матриці методом Гауса наведено в інструкції до лабораторної роботи № 2. Єдиною модифікацією є $V_{ij} = J_{ij}$ (замість базової матриці A_{ij} беремо J_{ij}).
5. Згідно ітераційної формули обчислюємо наступне наближення $x_i^{(k)}$, для $i=\overline{1,n}$.
6. Здійснюємо перевірку умови збіжності. Якщо вона не виконується, то процес уточнення повторюємо (п.3).
7. Для перевірки вірності роботи алгоритму підставляємо наші знайдені значення $x_1, x_2, \ldots, x_n$ в систему рівнянь $F(x)=0$. Значення функцій f_i, для $i=\overline{1,n}$, мають бути близькими нулю, в залежності від вибраного значення ε.

3.3. Метод січних

Метод січних грунтується на використанні формули (3.2) для обчислення матриці Якобі за кінцево-різницевою схемою, де

$$h_j^{(k)} = x_j^{(k-1)} - x_j^{(k)}, \text{ для } j = \overline{1,n}. \qquad (3.3)$$

Метод січних є двокроковим: для обчислення чергового обчислення $x_i^{(k)}$ використовуються два попередніх наближення $x_i^{(k-1)}$ та $x_i^{(k-2)}$. Перед початком обчислень необхідно задати два початкових наближення $x_i^{(0)}$ та $x_i^{(1)}$, для $i = \overline{1,n}$.

Наведемо геометричну ілюстрацію методу січних для розв'язування системи рівнянь на прикладі системи (1.3) при $t = 0$. Як початкові наближення візьмемо $X^{(0)} = [0{,}1;\ 0{,}1]$ та $X^{(1)} = [0{,}5;\ 0{,}5]$. Числові значення матриці Якобі та вектора кроку будуть такими

$$J^{(1)} = \begin{bmatrix} 2{,}4 & 0{,}6 \\ 1 & -0{,}6 \end{bmatrix}; \qquad h^{(1)} = \begin{bmatrix} -0.4 \\ -0.4 \end{bmatrix}. \qquad (3.4)$$

Як видно з рис. 6 січні площини точно перетинають графіки функцій у крапках $X^{(0)}$ та $X^{(1)}$. Наближення $X^{(2)}$ визначається як крапка перетину одразу трьох площин.

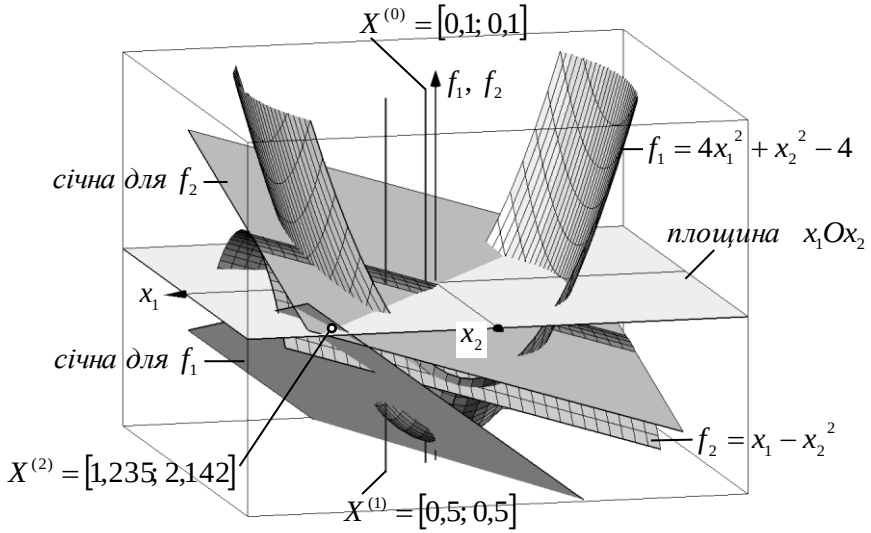


Рис.6. Наближення $X^{(2)}$ методу Січних

Загальний алгоритм методу січних

Ітераційний процес методу:

Ініціалізація одиничної матриці:

для $i, j = \overline{1,n}$

якщо $i = j \ \rightarrow \{E_{ij} = 1\}$

інакше $\{E_{ij} = 0\}$

Обчислення вектора кроку h_i:

для $i = \overline{1,n}$

$h_i = x_{OLDER\ i} - x_{OLD\ i}$

Обчислення вектора функцій f_i та матриці Якобі J_{ij}:

$x_i \rightarrow$ обчисл. функцій $f(x) \rightarrow f_i$

для $i, j = \overline{1,n}$

для $k = \overline{1,n}$

$\tilde{x}_k = x_k$

$\tilde{x}_j = x_j + h_j$

$\tilde{x}_i \rightarrow$ обчисл. функцій $f(x) \rightarrow \tilde{f}_i$

$J_{ij} = \left(\tilde{f}_i - f_i\right) / h_j$

для $b = \overline{1,n}$

Обертання матриці Якобі:

Метод Гауса

Прямий хід:

для $i, j = \overline{1,n}$

$V_{ij} = J_{ij}$

$P_i = E_{ib}$

далі згідно алгоритму методу Гауса

⋮

для $i=\overline{1,n}$

$$INVERS_{ib} = X_i$$

для $i=\overline{1,n}$

$$x_i = x_{OLD\ i} - \sum_{j=1}^{n} INVERS_{ij} \cdot f_j$$

Умова збіжності:

для $i=\overline{1,n}$

$$\left|\frac{x_i - x_{OLD\ i}}{x_i}\right| \cdot 100\% < \varepsilon\ ;\ x_{OLDER\ i} = x_{OLD\ i}\ ;\ x_{OLD\ i} = x_i$$

Опис алгоритму

1. На початку алгоритму задаємо початкове наближення $x_{OLDER\ i} = x_i^{(0)}$, $x_i = x_{OLD\ i} = x_i^{(1)}$, для $i=\overline{1,n}$, та відносну похибку ε у відсотках.
2. Встановлюємо значення для одиничної матриці E.
3. Обчислюємо значення елементів вектора кроку h_i.
4. Обчислюємо значення елементів вектора функцій f_i та елементів матриці Якобі J_{ij} з кінцевих різниць. Обчислення функцій виконуються в зовнішній програмній процедурі, у яку як вхідні дані поступає вектор невідомих, а вихідними даними служить обчислений вектор функцій.
5. Здійснюємо обертання матриці Якобі J_{ij} методом Гауса (з вибором головного елемента по рядку, стовпцю чи по всій матриці). Алгоритм обертання матриці методом Гауса наведено в інструкції до лабораторної роботи № 2. Єдиною модифікацією є $V_{ij} = J_{ij}$ (замість базової матриці A_{ij} беремо J_{ij}).
6. Згідно ітераційної формули обчислюємо наступне наближення $x_i^{(k)}$, для $i=\overline{1,n}$.
7. Здійснюємо перевірку умови збіжності. Якщо вона не виконується, то процес уточнення повторюємо (п.3).
8. Для перевірки вірності роботи алгоритму підставляємо наші знайдені значення $x_1, x_2, \ldots, x_n$ в систему рівнянь $F(x)=0$. Значення функцій f_i, для $i=\overline{1,n}$, мають бути близькими нулю, у залежності від вибраного значення ε.

4. Методи екстраполяції

Нехай (S_n) є скалярною послідовністю, що збігається до границі S. Метод екстраполяції полягає в перетворенні цієї послідовності в нову послідовність (T_n), шляхом перетворення $T:(S_n)\to(T_n)$. Вважають, що перетворення T прискорює збіжність послідовності (S_n), якщо і тільки якщо

$$\lim_{n\to\infty}\frac{T_n-S}{S_n-S}=0. \qquad (4.1)$$

Тоді ми можемо сказати, що послідовність (T_n) швидше збігається до S ніж (S_n).

Найперші методи екстраполяції використовували лінійні перетворення

$$T_n=\sum_{i=0}^{\infty}a_{ni}S_i, \qquad n=0,1,..., \qquad (4.2)$$

де a_{ni} – константи, незалежні від елементів послідовності (S_n). Таке лінійне перетворення переважно називають процесом сумування, а його властивості повністю визначені матрицею $A=\{a_{ni}\}$. Із практичних причин, тільки скінчене число коефіцієнтів a_{ni} відрізняються від нуля для кожного n. Серед таких методів – методи Ейлера, Чезара та Холдера. У випадку лінійних методів збіжність послідовності (T_n) до S для будь-якої збіжної послідовності (S_n) визначається теоремою сумування Toeplitz.

Приклади таких процесів:

$$T_n=\frac{1}{n+1}\sum_{i=0}^{n}S_i \quad \text{або} \qquad T_n=\frac{1}{k+1}\sum_{i=n}^{n+k}S_i \qquad (4.3)$$

У другому випадку, послідовність (T_n) також залежить від другого індексу k, і збіжність повинна бути досліджена або коли k зафіксовано, а n прямує до безмежності, або коли n зафіксовано, а k прямує до безмежності.

4.1. Скалярний ε-алгоритм

Алгоритм був отриманий Пітером Віном (Peter Wynn) у 1956 році. Цей алгоритм покращує збіжність послідовності

$$S=(s_0, s_1, s_2, ..., s_i) \qquad (4.4)$$

та складається з ініціалізації та етапу ітерацій:

Ініціалізація: **для** $j = 0, 1, 2, \ldots$

$$\varepsilon_{-1}^{(j)} = 0 \qquad \text{(штучно)}, \tag{4.5}$$

$$\varepsilon_{0}^{(j)} = s_j . \tag{4.6}$$

Ітерація: **для** $k, j = 0, 1, 2, \ldots$

$$\varepsilon_{k+1}^{(j)} = \varepsilon_{k-1}^{(j+1)} + \frac{1}{\varepsilon_{k}^{(j+1)} - \varepsilon_{k}^{(j)}} . \tag{4.7}$$

Ілюстрація роботи методу зображена на рис. 7.

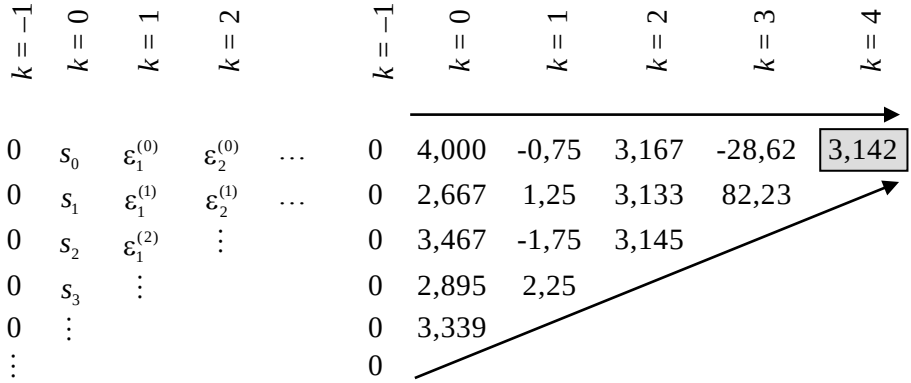


Рис. 7. Епсилон-таблиця та її числовий приклад

Як видно з рис. 7 скалярний ε-алгоритм утворює квадратну матрицю розмірності $(n+1)\times(n+1)$, де n – непарна кількість елементів послідовності (S_n). Реалізація обчислення цього прикладу алгоритмічною мовою C++ виглядає так:

```
const int n=5;
double e[n+1][n+1] = {{0, 4}, {0, 2.667}, {0, 3.467}, {0, 2.895}, {0, 3.339}, {0, 0}};
for (int k=1; k<=n-1; k++)
  for (int j=0; j<n-k; j++)
    e[j][k+1] = e[j+1][k-1] + 1 / (e[j+1][k] - e[j][k]);
```

4.2. Векторний ε-алгоритм

Перед тим як узагальнити скалярний епсилон алгоритм для векторного випадку, необхідно визначити інверсію вектора розмірності m. Для цього використовують процедуру обертання Самельсона

$$V_i^{-1} = \frac{V_i}{\sum_{j=1}^{m} V_j^{\,2}}, \text{ для } i = \overline{1,m}. \tag{4.8}$$

Після чого, векторний ε-алгоритм може бути записаний так:

Ініціалізація: **для** $j = 0, 1, 2, \ldots; \quad i = \overline{1, m}$

$$\varepsilon_{-1,i}^{(j)} = 0 \qquad \text{(штучно)}, \tag{4.9}$$

$$\varepsilon_{0,i}^{(j)} = s_{j,i}. \tag{4.10}$$

Ітерація: **для** $k,\ j = 0, 1, 2, \ldots; \quad i = \overline{1, m}$

$$\varepsilon_{k+1,i}^{(j)} = \varepsilon_{k-1,i}^{(j+1)} + \left[\varepsilon_{k,i}^{(j+1)} - \varepsilon_{k,i}^{(j)}\right]^{-1}. \tag{4.11}$$

Для діючих значень векторний ε-алгоритм був доведений Маклеодом (McLeod) [5], а для комплексного випадку – Грейвсом-Морісом (Graves-Morris) [6].

Приклад. Послідовність (S_n) обчислюється таким чином

$$s_{0,i} = b_i = \left[-0{,}1;\ 1{,}5\right]^T \tag{4.12}$$

(T позначає транспонування вектора) та є отримана рекурсивно

$$s_i^{(j+1)} = b_i + \sum_{d=1}^{2} G_{id} s_d^{(j)}, \quad j = 0, 1, \ldots, \tag{4.13}$$

де

$$G = \begin{bmatrix} 0{,}6 & 0{,}5 \\ -1 & 0{,}5 \end{bmatrix}.$$

Границею для (4.13) є вектор $X = \left[1; 1\right]$, який є розв'язком системи лінійних алгебричних рівнянь $A \cdot X = B$, де $A = E - G$, E – одинична матриця.

На рис. 8 наведено епсилон-таблицю для $k = 0, 2, 4$. Як бачимо, вже при $\varepsilon_{4,i}^{(j)}$, для $j = 0, 1, 2$, ми отримуємо нашу границю X, хоча остаточно обчислення завершуються при $k = 6$, де й отримуємо

шукану границю. При збільшенні кількості ітерацій j для (4.13) обчислення границі за ε-алгоритмом завершується аварією. Це свідчить про те, що необхідний правильний вибір кількості членів послідовності (S_n), однак на жаль не існує строгого критерію вибору їх кількості, тому тут можливий лише евристичний підхід.

У правій частині рис. 8 наведено графічну ілюстрацію розглянутого прикладу. Промені з крапкою на кінцях відображають координати векторів $\varepsilon_{k,i}^{(j)}$ на площині при різних j та k.

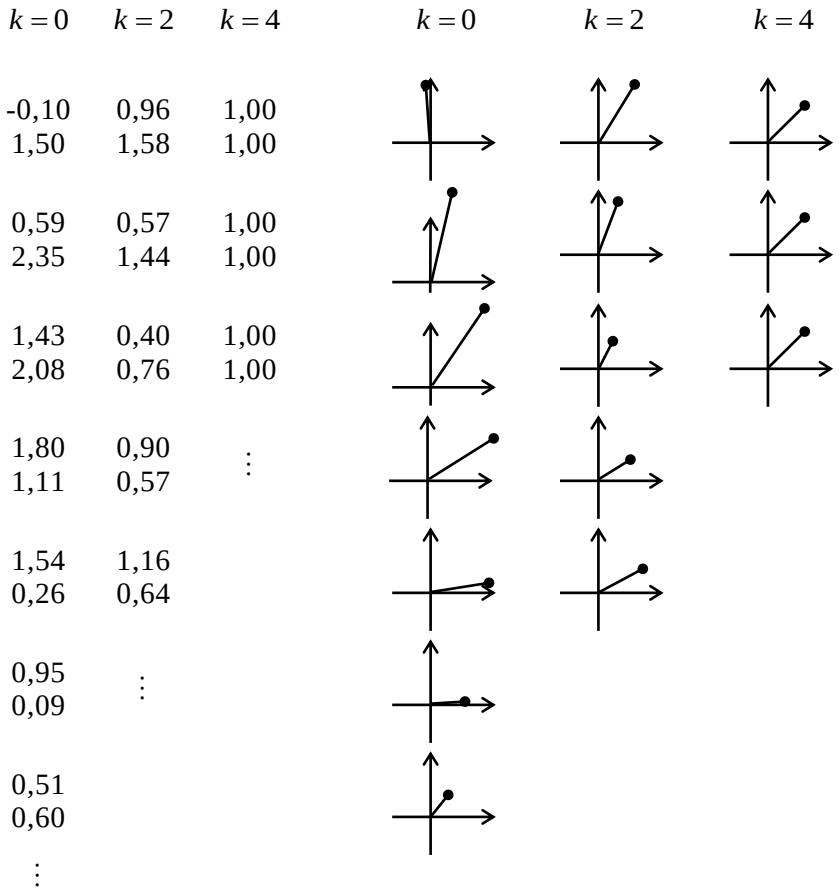


Рис. 8. Векторна епсилон-таблиця для розглянутого прикладу та її графічна ілюстрація

Реалізація обчислення цього прикладу алгоритмічною мовою С++ виглядає так:

```
const int n=7, m=2;
double G[m][m] = {{0.6, 0.5}, {-1, 0.5}};
double e[n+1][n+1][m], s[n][m] = {{-0.1, 1.5}}, Gs[m], V[m], w;
for (int j=0; j<n+1; j++)              //-- задання нулів для k=-1 е-алгоритму--
  for (int i=0; i<m; i++)
    e[j][0][i] = 0;
  for (int j=0; j<n-1; j++)            //----- обчислення Sn-послідовності ------
    for (int i=0; i<m; i++)
      {
        Gs[i] = 0;
        for (int d=0; d<2; d++)
          Gs[i] += G[i][d] * s[j][d];
        s[j+1][i] = s[0][i] + Gs[i];
      }                                //-- кінець обчислення Sn-послідовності --
  for (int j=0; j<n; j++)              //- присвоєння значень для k=0 е-алгоритму-
    for (int i=0; i<m; i++)
      e[j][1][i] = s[j][i];
  for (int k=1; k<=n-1; k++)           //-------- е-алгоритм ------------
      for (int j=0; j<n-k; j++)
        {
          for (int i=0; i<m; i++)      //----- обертанння Самельсона ------
            V[i] = e[j+1][k][i] - e[j][k][i];
          w = 0;
          for (int i=0; i<m; i++)
            w += V[i] * V[i];
          for (int i=0; i<m; i++)
            V[i] = V[i] / w;           //-- кінець обертанння Самельсона --
          for (int i=0; i<m; i++)
            e[j][k+1][i] = e[j+1][k-1][i] + V[i];
        }                              //------ кінець е-алгоритму -------
```

4.3. Застосування ε-алгоритму для лінійних систем

Розглянемо систему лінійних алгебричних рівнянь

$$A \cdot X = B, \tag{4.14}$$

де A – невироджена $m \times m$ матриця, B – вектор вільних членів, X – вектор невідомих.

Приведемо систему (4.14) до еквівалентного виду

$$X = X - \tau\left(A \cdot X - B\right) \tag{4.15}$$

або

$$X = \left(E - \tau \cdot A\right)X + \tau \cdot B, \tag{4.16}$$

який відповідає методу простої ітерації

$$X^{(k+1)} = V \cdot X^{(k)} + P, \tag{4.17}$$

де $V = E - \tau \cdot A$, $P = \tau \cdot B$ та параметр $\tau > 0$ вибирають так, щоб по можливості зробити мінімальною величину норми $\|V\|_2$

$$\|V\|_2 = \max_{1 \le j \le m} \sqrt{\lambda_j \left(V^T V\right)}, \tag{4.18}$$

де $\lambda_j \left(V^T V\right)$ – власні числа матриці $V^T V$. Нагадаємо, що число λ називають власним числом матриці A, якщо $\det\left(A - \lambda E\right) = 0$. Кожна матриця порядку m має рівно m власних чисел з врахуванням їх кратності.

Відомо, що умова збіжності методу простої ітерації $\|V\|_2 < 1$ буде виконана, якщо вибрати $\tau \in \left(0,\, 2 / \lambda_{\max}\right)$. Оптимальним є вибір

$$\tau = \frac{2}{\lambda_{\min} + \lambda_{\max}}. \tag{4.19}$$

Для покращення збіжності ітераційного процесу перетворення системи (4.14) до виду (4.17) можна здійснити також за допомогою спеціальної матриці, яка називається передобумовником. Наприклад, якщо матриця M близька до матриці A, то перетворена система у такий спосіб

$$M^{-1} \cdot A \cdot X = M^{-1} \cdot B \tag{4.20}$$

має такий ж розв'язок, що й система (4.14), але власні числа матриці $M^{-1} \cdot A$ будуть близькими до одиниці.

Тоді у формулі (4.17) вхідні матриця та вектор будуть такими: $V = E - M^{-1} \cdot A$, $P = M^{-1} \cdot B$.

В якості простого передобумовника можемо використати діагональ з коефіцієнтами вхідної матриці A

$$M = \mathrm{diag}\left[a_{11}, a_{22}, \ldots, a_{mm}\right]. \tag{4.21}$$

Застосування такого передобумовлення здійснюється достатньо просто, хоча з іншої сторони, використання більш складних процедур переважно дає кращі результати.

Приклад застосування передобумовника.

Розглянемо систему рівнянь з матрицею коефіцієнтів

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

Після застосування передобумовника $M = \mathrm{diag}[2, 4, 2]$ до матриці A, отримаємо перетворену матрицю

$$M^{-1} \cdot A = \begin{bmatrix} 0{,}5 & 0 & 0 \\ 0 & 0{,}25 & 0 \\ 0 & 0 & 0{,}5 \end{bmatrix} \cdot \begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0{,}25 & 0 \\ 0{,}25 & 1 & 0{,}25 \\ 0 & 0{,}5 & 1 \end{bmatrix}.$$

Тепер повернемося до ε-алгоритму. Починаючи з початкового вектора $s_{0,i}$, для $i = \overline{1,m}$, за допомогою ітераційної формули (4.17) будуємо послідовність (S_n)

$$s_{j+1,i} = \sum_{d=1}^{m} V_{id} \cdot s_{j,d} + P_i, \quad \text{для } j = 0, 1, \ldots; \quad i = \overline{1, m} \qquad (4.22)$$

Якщо лінійний процес (4.22) є збіжним, то з практичної сторони вважається за доцільніше застосовувати методи екстраполяції після визначеного числа p початкових ітерацій. З метою зменшення комп'ютерної пам'яті для запам'ятовування значень ітераційних кроків для великих систем рівнянь алгоритм знову повторюється після визначеного числа ітерацій.

Остаточно ε-алгоритм для розв'язування систем лінійних алгебричних рівнянь виглядає так:

1.Вибираємо $x_{0,i}$ та цілі числа p та q.

2. Початкові ітерації

$z_{0,i} = x_{0,i}, \quad i = \overline{1, m}$

$$z_{j+1,i} = \sum_{d=1}^{m} V_{id} \cdot z_{j,d} + P_i, \quad j = 0, \ldots, p-1; \quad i = \overline{1, m}$$

3. Етап екстраполяції

$s_{0,i} = z_{p,i}, \quad i = \overline{1, m};$

якщо $|s_{1,i} - s_{0,i}| < \varepsilon$, для $i = \overline{1, m}$, тоді процес зупиняємо;

інакше $s_{j+1,i} = \sum_{d=1}^{m} V_{id} \cdot z_{j,d} + P_i, \quad j = 0, \ldots, 2q-1, \quad i = \overline{1, m};$

обчислюємо наближення $x_{0,i} = \varepsilon_{2m}^{(0)}$ векторним ε-алгоритмом;

переходимо до п.2.

Для малих систем $q = m$ (розмірності системи), тоді $n = 2q + 1$.

4.4. Застосування ε-алгоритму для нелінійних систем

Нехай необхідно знайти розв'язок нелінійної системи (1.1) із заданою точністю ε. Перетворимо систему (1.1) до такого еквівалентного вигляду (до вигляду, зручного для методу простої ітерації)

$$\left.\begin{aligned} x_1 &= g_1(x_1, x_2, \dots, x_m) \\ x_2 &= g_2(x_1, x_2, \dots, x_m) \\ \dots &\quad \dots \quad \dots \quad \dots \quad \dots \\ x_m &= g_m(x_1, x_2, \dots, x_m) \end{aligned}\right\}. \tag{4.23}$$

У векторній формі система (4.23) записується так:

$$X = G(X), \tag{4.24}$$

де $G = [g_1, g_2, \dots, g_m]^T$, $X = [x_1, x_2, \dots, x_m]^T$.

Для методу простої ітерації формула (4.24) записується так

$$X^{(k+1)} = G(X^{(k)}). \tag{4.25}$$

Запис (4.25) означає, що чергове наближення $X^{(k+1)}$ обчислюється через попереднє наближення $X^{(k)}$ таким чином:

$$\left.\begin{aligned} x_1^{(k+1)} &= g_1(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}) \\ x_2^{(k+1)} &= g_2(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}) \\ \dots &\quad \dots \quad \dots \quad \dots \quad \dots \\ x_m^{(k+1)} &= g_m(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)}) \end{aligned}\right\}. \tag{4.26}$$

Збіжність методу простої ітерації. Якщо в околиці розв'язку $\overline{X} \in (X_0, A)$ функції $g_i(x_1^{(k)}, x_2^{(k)}, \dots, x_m^{(k)})$, для $i = \overline{1, m}$, диференційовані та виконується нерівність

$$\left\| \frac{\partial G(X)}{\partial X} \right\| < 1 \tag{4.27}$$

де

$$\left\| \frac{\partial G(X)}{\partial X} \right\| = \max_{1 \le i \le m} \sum_{j=1}^{m} \left| \frac{\partial g_i(x_1, x_2, \dots, x_m)}{\partial x_j} \right|,$$

то кажуть, що метод збігається.

Умова (4.27), грубо кажучи, означає, що в околиці розв'язку похідні $\partial g_i / \partial x_j$ для всіх i та j повинні бути «достатньо малими по модулю». Іншими словами, систему (1.1) необхідно перетворити до такого вигляду (4.23), щоб функції g_i слабо мінялися при зміні аргументів, тобто були «майже постійними». Якихось загальних рецептів, як це зробити, у загальному випадку немає.

Тепер нехай (S_n) є послідовністю векторів, отриманих від початкового наближення $s_{0,i}$, для $i = \overline{1,m}$, таким чином:

$$s_{j+1,i} = G(s_{j,i}), \quad \text{для } j = 0, 1, \ldots; \quad i = \overline{1,m}. \tag{4.28}$$

Як і у випадку систем лінійних алгебричних рівнянь, доцільно застосовувати методи екстраполяції після визначеного числа p початкових ітерацій та повторювати алгоритм після визначеного числа ітерацій.

ε -алгоритм для розв'язування систем нелінійних рівнянь:

1. Вибираємо $x_{0,i}$ та цілі числа p та q.
2. Початкові ітерації
   $z_{0,i} = x_{0,i}, \quad i = \overline{1,m}$
   $z_{j+1,i} = G(z_{j,i}), \quad j = 0, \ldots, p-1; \quad i = \overline{1,m}$
3. Етап екстраполяції
   $s_{0,i} = z_{p,i}, \quad i = \overline{1,m};$
   якщо $|s_{1,i} - s_{0,i}| < \varepsilon$, для $i = \overline{1,m}$, тоді процес зупиняємо;
   інакше $s_{j+1,i} = G(s_{j,i}), \quad j = 0, \ldots, 2q-1, \quad i = \overline{1,m};$
   обчислюємо наближення $x_{0,i} = \varepsilon_{2m}^{(0)}$ векторним ε -алгоритмом;
   переходимо до п.2.

Для малих систем $q = m$ (розмірності системи), тоді $n = 2q+1$.

Загальний алгоритм екстраполяційного методу ε -алгоритму для розв'язування систем нелінійних рівнянь

Встановлення нулів для $k = -1$ ε -3Dматриці:

для $j = \overline{0, n}$; $i = \overline{0, m-1}$

$$e_{j,0,i} = 0$$

Ітераційний процес методу:

$$cond = 0_{логічний}$$

початкові ітерації: **для** $j = \overline{1, p}$

$$x_0 = g_0(x_0, \ldots, x_{m-1}), \ \ldots, \ x_{m-1} = g_{m-1}(x_0, \ldots, x_{m-1})$$

Встановлення $s_0 = x$: **для** $i = \overline{0, m-1}$

$$e_{0,1,i} = x_i$$

Генерування S-послідовності: **для** $j = \overline{0, 2q-1}$

$$e_{j+1,1,0} = x_0 = g_0(x_0, \ldots, x_{m-1})$$

$$\vdots$$

$$e_{j+1,1,m-1} = x_{m-1} = g_{m-1}(x_0, \ldots, x_{m-1})$$

якщо $j = 0$

для $i = \overline{0, m-1}$

$$cond = (cond) \text{ АБО } \left(\left| \frac{e_{1,1,i} - e_{0,1,i}}{e_{1,1,i}} \right| \cdot 100\% > \varepsilon \right)$$

якщо $cond = 0_{логічний}$ → **{ вихід із циклу }**

Екстраполяція: **якщо** $cond = 1_{логічна}$

для $k = \overline{1, n-1}$; $j = \overline{0, n-k-1}$

для $i = \overline{0, m-1}$

$$V_i = e_{j+1,k,i} - e_{j,k,i}$$

$$sum = 0$$

для $i = \overline{0, m-1}$

$$sum = sum + V_i \cdot V_i$$

для $i = \overline{0, m-1}$

$$V_i = V_i / sum$$

для $i = \overline{0, m-1}$

$e_{j,\,k+1,\,i} = e_{j+1,\,k-1,\,i} + V_i$

для $i = \overline{0, m-1}$

$x_i = e_{0,\,n,\,i}$

Умова завершення ітераційного процесу: $cond = 0_{\text{логічний}}$

Опис алгоритму

1. На початку алгоритму задаємо значення цілих чисел m (розмірність системи рівнянь), q (можемо прийняти $q = m$), $n = 2q + 1$ (розмір (S_n) послідовності), p (кількість початкових ітерацій).
2. Задаємо початкове наближення $x_i^{(0)}$, для $i = \overline{1, m}$, та відносну похибку ε у відсотках.
3. Встановлюємо нулі у ε -матриці для випадку, коли $k = -1$.
4. Виконуємо p початкових ітерацій.
5. Встановлюємо початкові значення для (S_n) послідовності.
6. Генеруємо (S_n) послідовність. На етапі генерування послідовності здійснюємо перевірку збіжності методу. Використана змінна *cond* є логічного типу, відповідно, і операції, що проводяться з нею є логічними. Оператор $(\ )\text{АБО}(\ )$ – виконує логічну операцію «або».
7. На етапі екстраполювання обчислюємо інверсний вектор за допомогою процедури обертання Самельсона та шукаємо границю (S_n) послідовності, яка відповідає елементу матриці $e_{0,\,n,\,i}$. Присвоюємо знайдену границю вектору невідомих.
8. Перевіряємо умову завершення ітераційного процесу. Якщо вона не виконується, то процес уточнення повторюємо (п.4).
9. Для перевірки вірності роботи алгоритму підставляємо наші знайдені значення $x_1, x_2, \ldots, x_m$ в систему рівнянь $F(x) = 0$. Значення функцій f_i, для $i = \overline{1, m}$, мають бути близькими нулю, у залежності від вибраного значення ε.

***Зауваження**: алгоритм адаптований для алгоритмічної мови С++.*

Проілюструємо результати виконання роботи наведеного алгоритму на прикладі розв'язування системи нелінійних рівнянь (1.3). Спершу приведемо систему до вигляду (4.23) так, щоб задовольнялася умова збіжності методу простої ітерації (4.27)

$$\left.\begin{aligned} x_1 &= \frac{1}{x_1 + \dfrac{x_1^{\,2}}{4x_1}} \\ x_2 &= \sqrt{x_1} \end{aligned}\right\}. \qquad (4.29)$$

Виберемо значення: цілих чисел $m = 2$, $q = 2$, $n = 5$, $p = 2$, початкового наближення $x^{(0)} = [0{,}5;\ 0{,}5]$ та значення відносної похибки у відсотках $e = 10^{-6}$.

Згідно вхідних даних ми одержали такі результати.

Вектор невідомих:

$$\left.\begin{aligned} x_1 &= 0{,}882782218732049 \\ x_2 &= 0{,}939564909270269 \end{aligned}\right\}.$$

Вектор функцій системи рівнянь:

$$\left.\begin{aligned} f_1 &= 1{,}57 \cdot 10^{-9} \\ f_2 &= -1{,}04 \cdot 10^{-16} \end{aligned}\right\}.$$

Алгоритм виконав 15 ітераційних наближень та 2 стадії пошуку границі (S_n) послідовності. У той час, коли для методу простої ітерації необхідно було виконати 76 ітераційних наближень.

5. Порядок виконання роботи

1. Вдома вивчити теоретичні відомості, необхідні для виконання лабораторної роботи.
2. Згідно варіанту (порядкового номера в журналі викладача) завдання (таблиця 1 та 2), вдома написати програму для реалізації алгоритму вказаного методу, а в лабораторії вписати програмний код та налагодити цю програму.
3. Отримані на комп'ютері числові результати представити викладачу.
4. По результатах виконаної роботи оформити звіт та здати його.

Таблиця 1. Завдання до лабораторної роботи

***система нелінійних рівнянь вибирається з таблиці 2**

№ п/п	**Завдання (метод та початкові наближення коренів)** відносна похибка у відсотках $e = 10^{-5}$ %, для ε -алгоритму: $q = 2$, $p = 2$		
	Група 1	**Група 2**	**Група 3**
1	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт. матриці методом Гауса з вибор. гол. ел-тів по рядку)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)
2	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт. матриці методом Гауса з вибор. гол. ел-тів по рядку)
3	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт. матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)
4	ε -алгоритм	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт. матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
5	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)

6	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм
7	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)
8	ε -алгоритм	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
9	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)
10	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)	ε -алгоритм
11	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по по всій матриці)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)
12	ε -алгоритм	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по по всій матриці)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)

13	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по по всій матриці)
14	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм
15	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)
16	ε -алгоритм	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
17	Стандартний метод Ньютона (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)	ε -алгоритм	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)
18	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм
19	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)

20	ε -алгоритм	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
21	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по всій матриці)
22	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм
23	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)
24	ε -алгоритм	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
25	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)	ε -алгоритм	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)
26	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)	ε -алгоритм

27	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з виб. гол. ел-тів по стовпцю)
28	ε -алгоритм	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)	Метод січних (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)
29	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм	Стандартний метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по всій матриці)
30	Метод Ньютона з кінцево-різницевою матрицею Якобі (оберт.матриці методом Гауса з вибор. гол. ел-тів по рядку)	Спрощений метод Ньютона (оберт.матриці методом Гауса з вибор. гол. ел-тів по стовпцю)	ε -алгоритм

Таблиця 2. Системи нелінійних рівнянь $F(X)=0$

№1 $\begin{cases} e^{x_1}\cos x_2 - 1 - x_1 = 0 \\ e^{x_1}\sin x_2 + 1 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0} = -0{,}1$, $x_2^{\,0} = 0{,}1$	№2 $\begin{cases} \dfrac{\left(x_1^{\,2} - x_2^{\,2}\right)^2}{4} - x_1^{\,2}x_2^{\,2} + 0{,}5 - x_1 = 0 \\ x_1 x_2\left(x_1^{\,2} - x_2^{\,2}\right) + 0{,}5 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0} = 0$, $x_2^{\,0} = 0$
№3 $\begin{cases} x_1^{\,2} - x_2^{\,2} - 0{,}1 - x_1 = 0 \\ 2x_1x_2 + 0{,}1 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0} = 1$, $x_2^{\,0} = 1$	№4 $\begin{cases} 4x_1^{\,2} + x_2^{\,2} - 4 = 0 \\ x_1 - x_2^{\,2} + 1 = 0 \end{cases}$ поч. наближення $x_1^{\,0} = 2$, $x_2^{\,0} = 2$

№5

$$\begin{cases} \dfrac{x_1}{x_1^2 + x_2^2} + 0{,}4 - x_1 = 0 \\ -\dfrac{x_2}{x_1^2 + x_2^2} - 1{,}4 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0{,}5$, $x_2^0 = 0{,}5$

№6

$$\begin{cases} \dfrac{x_1}{x_1^2 + x_2^2} + 0{,}4 - x_1 = 0 \\ -\dfrac{x_2}{x_1^2 + x_2^2} + 1{,}4 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0{,}1$, $x_2^0 = 0{,}1$

№7

$$\begin{cases} x_1^2 - x_2^2 + 0{,}1 - x_1 = 0 \\ 2x_1x_2 + 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 1$, $x_2^0 = 1$

№8

$$\begin{cases} x_1^2 + 0{,}8x_2^2 + 0{,}1 - x_1 = 0 \\ 2x_1x_2 - 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0{,}5$, $x_2^0 = 0{,}5$

№9

$$\begin{cases} x_1^2 - x_2^2 + 0{,}1 - x_1 = 0 \\ 2x_1x_2 - 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 1$, $x_2^0 = 1$

№10

$$\begin{cases} x_1^2 x_2^2 - \dfrac{\left(x_1^2 - x_2^2\right)^2}{4} - 0{,}5 - x_1 = 0 \\ x_1x_2\left(x_2^2 - x_1^2\right) - 0{,}5 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 1$, $x_2^0 = 1$

№11

$$\begin{cases} x_1^2 + x_2^2 + 0{,}1 - x_1 = 0 \\ 2x_1x_2 + 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0$, $x_2^0 = 0$

№12

$$\begin{cases} x_1^2 + x_2^2 + 0{,}1 - x_1 = 0 \\ 2x_1x_2 - 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0$, $x_2^0 = 0$

№13

$$\begin{cases} 1 - e^{-x_1}\cos x_2 - x_1 = 0 \\ e^{-x_1}\sin x_2 + 1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0{,}1$, $x_2^0 = 0{,}1$

№14

$$\begin{cases} x_1^2 + x_2^2 - 0{,}1 - x_1 = 0 \\ 2x_1x_2 - 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^0 = 0{,}5$, $x_2^0 = 0{,}5$

№15

$$\begin{cases} 4x_1^{\,2} + x_2^{\,2} - 4 = 0 \\ x_1 - x_2^{\,2} + 2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 1{,}5$, $x_2^{\,0} = -1{,}5$

№16

$$\begin{cases} x_1^{\,2} - x_2^{\,2} - 0{,}1 - x_1 = 0 \\ 2x_1x_2 - 0{,}1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№17

$$\begin{cases} \dfrac{x_1}{x_1^{\,2} + x_2^{\,2}} - 0{,}4 - x_1 = 0 \\ 1{,}4 - \dfrac{x_2}{x_1^{\,2} + x_2^{\,2}} - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№18

$$\begin{cases} -x_1^{\,2} - 0{,}8x_2^{\,2} - 0{,}1 - x_1 = 0 \\ 0{,}1 - 2x_1x_2 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0$, $x_2^{\,0} = 0$

№19

$$\begin{cases} x_2^{\,2} - x_1^{\,2} - 0{,}1 - x_1 = 0 \\ 0{,}1 - 2x_1x_2 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№20

$$\begin{cases} 0{,}1 - x_1^{\,2} + x_2^{\,2} - x_1 = 0 \\ 0{,}1 - 2x_1x_2 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№21

$$\begin{cases} x_1^{\,2} + x_2^{\,2} + 0{,}1 + x_1 = 0 \\ 2x_1x_2 + 0{,}1 + x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0$, $x_2^{\,0} = 0$

№22

$$\begin{cases} 0{,}1 - x_1^{\,2} - x_2^{\,2} - x_1 = 0 \\ 2x_1x_2 + 0{,}1 + x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№23

$$\begin{cases} \dfrac{x_1}{x_1^{\,2} + x_2^{\,2}} - 0{,}4 - x_1 = 0 \\ \dfrac{1 - x_2}{x_1^{\,2} + x_2^{\,2}} + 1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№24

$$\begin{cases} e^{x_1}\cos x_2 - 1 - x_1 = 0 \\ e^{x_1}\sin x_2 - 1 - x_2 = 0 \end{cases}$$

поч. наближення $x_1^{\,0} = 0{,}5$, $x_2^{\,0} = 0{,}5$

№25	№26
$\begin{cases} \frac{\left(x_1^{\,2}-x_2^{\,2}\right)^2}{4} - x_1^{\,2}x_2^{\,2} + 0{,}5 - x_1 = 0 \\ x_1x_2\left(x_1^{\,2}-x_2^{\,2}\right) - 0{,}5 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0}=0$, $x_2^{\,0}=0$	$\begin{cases} 4x_1^{\,2} + x_2^{\,2} - 4 = 0 \\ x_1 - x_2^{\,2} = 0 \end{cases}$ поч. наближення $x_1^{\,0}=0{,}5$, $x_2^{\,0}=0{,}5$
№27	№28
$\begin{cases} 0{,}1 - x_1^{\,2} + x_2^{\,2} - x_1 = 0 \\ 0{,}1 - 2x_1x_2 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0}=0{,}5$, $x_2^{\,0}=0{,}5$	$\begin{cases} 1 - e^{-x_1}\cos x_2 - x_1 = 0 \\ e^{-x_1}\sin x_2 + 1 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0}=0{,}1$, $x_2^{\,0}=0{,}1$
№29	№30
$\begin{cases} x_1^{\,2} + 0{,}8x_2^{\,2} + 0{,}1 - x_1 = 0 \\ 2x_1x_2 + 0{,}1 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0}=1$, $x_2^{\,0}=1$	$\begin{cases} \frac{x_1}{x_1^{\,2}+x_2^{\,2}} + 0{,}4 - x_1 = 0 \\ \frac{1-x_2}{x_1^{\,2}+x_2^{\,2}} + 1 - x_2 = 0 \end{cases}$ поч. наближення $x_1^{\,0}=0{,}5$, $x_2^{\,0}=0{,}5$

6. Зміст звіту

1. Номер і назва лабораторної роботи, із зазначенням її виконавця.
2. Мета роботи.
3. Завдання до лабораторної роботи.
4. Короткі теоретичні відомості, що необхідні для виконання лабораторної роботи.
5. Блок-схема розробленої програми.
6. Список ідентифікаторів констант, змінних, функцій, методів, використаних у програмі, та їх пояснення.
7. Остаточна версія програми.
8. Результати виконання програми.
9. Висновки.

7. Контрольні запитання

1. У чому суть графічного методу пошуку розв'язків для системи з двох нелінійних рівнянь?
2. Основні етапи розв'язування систем нелінійних рівнянь.
3. На чому заснована ідея методу Ньютона для розв'язування систем нелінійних рівнянь?
4. Дайте геометричну інтерпретацію методу Ньютона для випадку двох нелінійних рівнянь.
5. У чому полягає відмінність спрощеного методу Ньютона від звичайного?
6. Дайте геометричну інтерпретацію спрощеного методу Ньютона для випадку двох нелінійних рівнянь.
7. Як здійснюється наближене обчислення матриці Якобі?
8. У чому полягає ідея методу Січних.
9. Для чого використовуються методи екстраполяції?
10. Наведіть формули ε-алгоритму та поясніть принцип його роботи.
11. У чому відмінність у застосуванні ε-алгоритму для скалярного та векторного випадків?
12. Як застосувати ε-алгоритм для розв'язування систем лінійних алгебричних рівнянь?
13. Як здійснюється приведення системи лінійних алгебричних рівнянь до вигляду зручного для методу простої ітерації?
14. Які існують підходи для забезпечення збіжності методу простої ітерації для розв'язування лінійних алгебричних рівнянь?
15. Як застосувати ε-алгоритм для розв'язування систем нелінійних алгебричних рівнянь?
16. Як здійснюється приведення системи нелінійних рівнянь до вигляду зручного для методу простої ітерації?
17. Який критерій збіжності методу простої ітерації?

8. Список літератури

1. Практикум з обчислювальної математики. Основні числові методи. Лекції: Навчальний посібник для вузів / І.А.Анджейчак, В.Є.Анохін, І.М.Бойко та ін. – Нац. ун-т "Львівська політехніка", Львів. – 2001. – 150 с.

2. Комп'ютерні методи прикладної математики: Навчальний посібник для студентів вищих технічних навчальних закладів / К.Х.Зеленський, В.М.Ігнатенко, О.П.Коц – К.: Академперіодика. – 2002. – 479 с.

3. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров: Учеб. пособие. – М.: Высш.шк. – 1994. – 544 с.

4. K. Jbiou, H. Sadok. Vector extrapolation methods. Applications and numerical comparison // Journal of computational and applied mathematics. – 2000. – V.122. – P. 149-165.

5. J. B. McLeod. A note on the ε-algorithm // Computing (Areh. Electron. Rechnen) – 1971. – V.7 – P. 17-24.

6. P.R. Graves-Morris. Vector valued rational interpolants I // Numer. Math. – 1983. – V.42. – P. 331-348.

7. P.R. Graves-Morris, D.E. Roberts, A. Salam. The epsilon algorithm and related topocs. // Journal of computational and applied mathematics. – 2000. – V.122. – P. 51-80.

ЗМІСТ

Навчальне видання

Системи нелінійних рівнянь. Метод Ньютона та ε-алгоритм: Інструкція до лабораторної роботи № 3 з курсу “Чисельні методи” для студентів спеціальності 122 “Комп’ютерні науки та інформаційні технології” спеціалізації “Системна інженерія (інтернет речей)”.

Укладачі: Дзелендзяк Уляна Юріївна
Павельчак Андрій Геннадійович