

Нелінійні системи рівнянь. Метод Ньютона

Системи лінійних алгебраїчних рівнянь – це лише окремий випадок систем рівнянь. На практиці розв’язування задач моделювання приводить переважно до систем нелінійних рівнянь. Системи n нелінійних рівнянь з невідомими

$$x_1, x_2, \dots, x_n$$

у загальному випадку прийнято записувати в такий спосіб:

[illegible]

де $f_i(x_1, x_2, \dots, x_n) = 0$ - будь-які функції незалежних змінних (лінійні чи нелінійні) щодо невідомих.

Позначимо:

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \text{ - вектор стовпець розмірності } n \text{ з елементами } x_i \in R \text{ (} x \in R^n \text{)}$$

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_n(x_1, x_2, \dots, x_n) \end{pmatrix} \text{ - вектор функція розмірності } n \text{ елементами}$$

якої є функції $f_i(x) = f_i(x_1, x_2, \dots, x_n)$, $i = 1, n$,

$$\bar{0} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

то систему можна записати у векторному вигляді

$$F(\bar{x}) = 0$$

Розв'язати систему – означає знайти таке $x^* \in R^n$, для якого $F(x^*) = 0$, тобто

[illegible]

Таким чином, початкове завдання можна розглядати як задачу про нулі нелінійного відображення. У цій постановці вона є прямим узагальненням основного завдання побудови методів знаходження нулів одномірних нелінійних відображень. Фактично це те саме завдання, лише в просторах більшої розмірності. Тому можна як заново будувати методи її розв'язування на основі розроблених вище підходів, так і здійснювати формальне перенесення виведених для скалярного випадку розрахункових формул. У будь-якому разі слід подумати про правомірність тих або інших операцій над векторними змінними та векторними функціями а також про збіжність отриманих у такий спосіб ітераційних процесів.

Проте, не всі результати та не всі методи можна перенести з випадку $n=1$ на випадок $n \geq 2$. Наприклад, тут уже не працюватимуть методи подвійного поділу, оскільки безліч векторів не впорядкована. Водночас перехід від $n=1$ до $n \geq 2$ вносить до задачі знаходження нулів нелінійного відображення свою специфіку, облік якої приводить до появи нових методів і до різних модифікацій тих, що вже є. Зокрема, велика варіативність методів розв'язування нелінійних систем пов'язана з різноманітністю способів, якими можна розв'язувати лінійні завдання алгебри, що виникають при покроковій лінеаризації заданої нелінійної вектор-функції $F(x)$.

Більшість математичних моделей різних процесів і явищ записуються в загальному випадку у вигляді $F(x)=0$, тому дана задача має величезне практичне значення.

Система нелінійних рівнянь може не мати розв'язків, мати єдиний розв'язок, скінченну чи нескінченну кількість розв'язків. Питання щодо кількості розв'язків має ухвалюватися для кожної конкретної задачі окремо.

На відміну від систем лінійних рівнянь, для систем нелінійних рівнянь є невідомими прямі методи розв'язування. Лише в окремих випадках систему можна розв'язати безпосередньо. Наприклад, для системи з двох рівнянь іноді вдається подати одне невідоме через інше й у такий спосіб звести задачу до розв'язання одного нелінійного рівняння щодо одного невідомого. Тому ітераційні методи для нелінійних систем набувають особливої актуальності.

Розглянемо кілька найпростіших ітераційних методів розв’язування систем нелінійних рівнянь.

Метод Ньютона

Нехай (A_k) - деяка послідовність невідроджених дійсних матриць розміру $n \times n$. Тоді, очевидно, послідовність завдань

$$x = x - A_k F(x)$$

має ті самі розв'язки, що й початкове рівняння, і для знаходження цих розв'язків формально можна записати ітераційний процес

$$x^{(k+1)} = x^{(k)} - A_k F(x^{(k)}) \quad (3)$$

що має вигляд методу простих ітерацій.

Якщо A_k різні при різних k , то формула (3) визначає велику сім'ю ітераційних методів із матричними параметрами.

Нехай $A_k = [F'(x^{(k)})]^{-1}$, де

$$F'(x) = J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad - (4)$$

матриця Якобі вектор-функції $F(x)$. Підставимо A_k в (3) і отримаємо явну формулу методу Ньютона:

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}) \quad (5)$$

Формулу (5), що вимагає оберення матриць на кожній ітерації можна переписати в неявному вигляді

$$F'(x^{(k)})(x^{(k+1)} - x^{(k)}) = -F(x^{(k)}) \quad (6)$$

Використання (6) передбачає при кожному $k = 0, 1, 2, \dots$ розв'язування лінійної алгебраїчної системи

$$F'(x^{(k)}) p^{(k)} = -F(x^{(k)}) \quad (7)$$

відносно векторної поправки, а потім збільшення цієї поправки до поточного наближення для отримання

$$x^{(k+1)} = x^{(k)} + p^{(k)} \quad (8)$$

До розв'язування таких лінійних систем залучають найрізноманітніші методи – як прямі, так і ітераційні, залежно від розмірності n розв'язуваної задачі та специфіки матриць Якобі ($J(x^{(k)})$).

Новим, порівняно із скалярним випадком чинником, що ускладнює застосування методу Ньютона для розв'язування n – вимірних систем, є необхідність розв'язування n -вимірних лінійних завдань на кожній ітерації (обернення матриць в (5), або розв'язування систем лінійних алгебраїчних рівнянь в (6), обчислювальні витрати на які зростають із збільшенням n , узагалі кажучи непропорційно швидко.

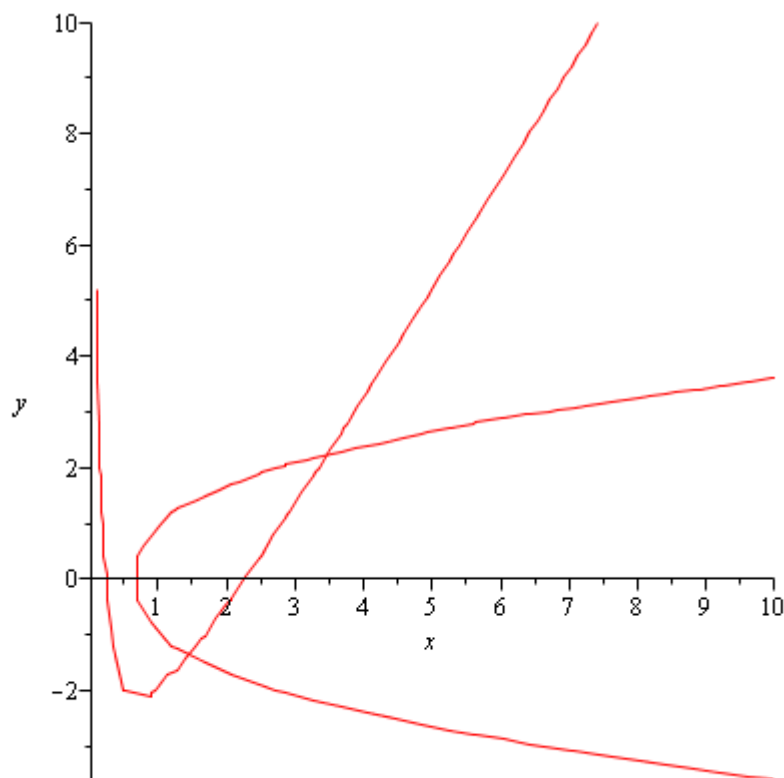
Приклад.

Розв'язати систему нелінійних рівнянь

$$\begin{cases} 2x^2 - xy - 5x + 1 = 0 \\ x + 3\lg x - y^2 = 0 \end{cases}$$

1. Щоб побачити скільки розв'язків має система використаємо пакет Maple для побудови графіків. Використаємо команду *with(plots)* (Рис.)

```
[> with(plots) :  
[> implicitplot({2*x^2 - x*y - 5*x + 1 = 0, x + 3*log10(x) - y^2 = 0}, x = 0.1 ..10, y = -10 ..10)
```



2. Позначимо

$$f_1(x, y) = 2x^2 - xy - 5x + 1$$

$$f_2(x, y) = x + 3\lg x - y^2$$

Введемо функції в робоче поле пакету Maple (Рис.)

```

> f1 := 2*x^2 - x*y - 5*x + 1
f1 := 2 x^2 - x y - 5 x + 1

> f2 := x + 3*log10(x) - y^2
f2 := x + 3 ln(x) / ln(10) - y^2

```

3. Обчислимо частинні похідні з використанням пакету Maple (Рис.):

```

> f1x := ∂/∂x f1
f1x := 4 x - y - 5

> f1y := ∂/∂y f1
f1y := -x

> f2x := ∂/∂x f2
f2x := 1 + 3 / (x ln(10))

> f2y := ∂/∂y f2
f2y := -2 y

```

4. Складемо матрицю Якобі

$$F'(x) = J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 4x - y & -x \\ 1 - \frac{3}{x \ln 10} & -2y \end{pmatrix}$$

5. Використовуємо рисунок для вибору початкових наближень:

$$x^{(0)} = 3, \quad y^{(0)} = 2$$

```

> x := 3
x := 3

> y := 2
y := 2

```

Обчислимо матрицю Якобі в точках початкового наближення, використовуючи пакет *with (linalg)* (Рис.)

```

> with(linalg):
> F := matrix([ [evalf(f1x), evalf(f1y)], [evalf(f2x), evalf(f2y)] ])
F := [ 5. -3.
       1.434294482 -4. ]

```

6. Знайдемо обернену матрицю Якобі

```

┌ > F1 := inverse(F)
└

```

$$F1 := \begin{bmatrix} 0.2548238708 & -0.1911179031 \\ 0.09137311795 & -0.3185298385 \end{bmatrix}$$

7. Знайдемо значення функцій $f_1(x, y)$ та $f_2(x, y)$ в точках початкового наближення та запишемо результат у вигляді вектора

```

┌ > f := matrix([[evalf(f1)], [evalf(f2)]])
└

```

$$f := \begin{bmatrix} -2. \\ 0.431363765 \end{bmatrix}$$

8. Помножимо обернену матрицю Якобі на вектор f . Результат запишемо у вигляді вектора M (Рис):

```

> M := evalm(F1&*f)

```

$$M := \begin{bmatrix} -0.5920890798 \\ -0.3201484663 \end{bmatrix}$$

9. Запишемо початкове наближення, як нульовий крок ітерації у вигляді вектора $X0$ (Рис)

```

┌ > X0 := matrix([[x], [y]])
└

```

$$X0 := \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

10. Обчислимо перший крок ітерації:

```

= > X1 := evalm(X0 - M)
=

```

$$X1 := \begin{bmatrix} 3.592089080 \\ 2.320148466 \end{bmatrix}$$

11. Виконаємо другий блок алгоритму, взявши за початкове наближення отримані результати першого кроку ітерації:

```

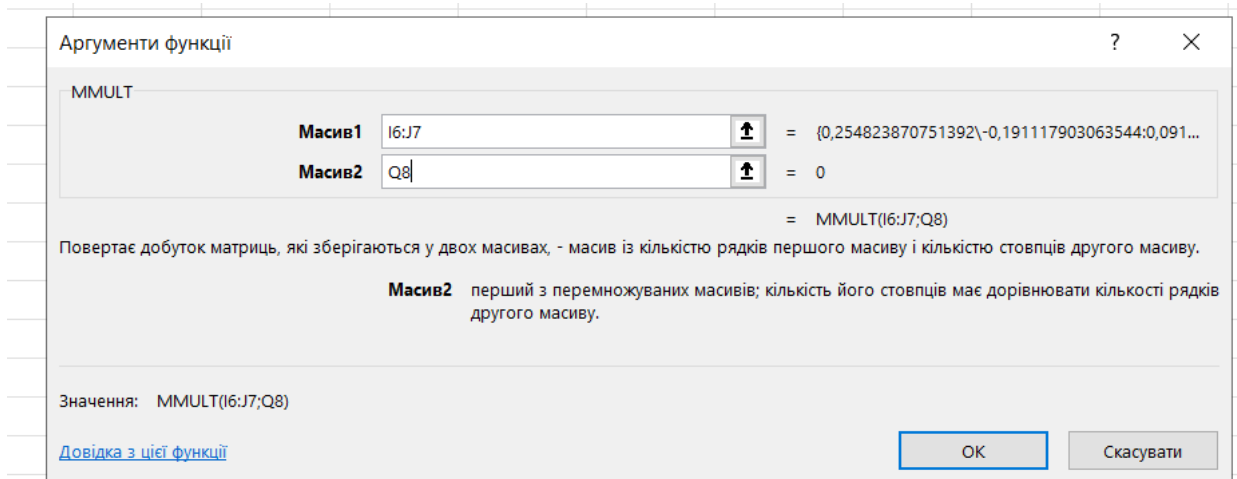
= > restart
= > f1 := 2*x^2 - x*y - 5*x + 1
=
> f2 := x + 3*log10(x) - y^2
=

```

$$f1 := 2x^2 - xy - 5x + 1$$

$$f2 := x + \frac{3 \ln(x)}{\ln(10)} - y^2$$

2. Об'єднаємо комірки A6:A7 і внесемо запис $k=0$, в комірки B6 та B7 введемо числа 3 та 2 відповідно, що відповідають початковому наближенню
3. Введемо формули в комірки :
 D6: $=2*B6^2-B6*B7-5*B6+1$
 D7: $= B6+3*LOG(B6)-B7^2$
 F6: $= 4*B6-B7-5$
 G6: $= -B6$
 F7: $= 1+3/(B6*LN(10))$
 G7: $= -2*B7$
4. Виділимо діапазон комірок I6:J7 і введемо формулу $=MINVERSE(F6:G7)$
5. Виділимо діапазон комірок L6:L7 і введемо формулу $=MMULT(I6:J7;D6:D7)$ заповнивши діалогове вікно



Щоб отримати результат натиснемо одночасно комбінацію клавіш Ctrl+Shift+Enter

Результати обчислень показано на рисунку

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	$f1(x,y)=2*x^2-x*y-5*x+1$				$f1'x=4*x-y-5$		$f1'y=-x$		$f2'x=1+3/(x*ln10)$			$f2'y=-2*y$	
2	$f2(x,y)=x+3*lgx-y^2$												
3													
4		X		F(x)		F'(x)			$F'(x)^{-1}$			$F'(x)^{-1}*F(X)$	
5													
6		3		-2		5	-3		0,254824	-0,19112		-0,59208908	
7	k=0	2		0,431364		1,434294	-4		0,091373	-0,31853		-0,32014847	
8													

6. Об'єднаємо комірки A10:A11 і внесемо запис $k=1$.
7. Внесемо формули в комірки B10 та B11
 B10: $= B6-L6$
 B11: $=B7-L7$

8. Введемо формули в комірки :

$$D10:=2*B10^2-B10*B11-5*B10+1$$

$$D11:=B10+3*LOG(B10)-B11^2$$

$$F10:=4*B10-B11-5$$

$$G10:= -B10$$

$$F11:= 1+3/(B10*LN(10))$$

$$G11:= -2*B11$$

9. Виділимо діапазон комірок I10:J11 і введемо формулу
=MINVERSE(F10:G11)

10. Виділимо діапазон комірок L6:L7 і введемо формулу
=MMULT(I6:J7;D6:D7) заповнивши діалогове вікно

Аргументи функції

?

×

MMULT

Масив1

I10:J11

↑

=

{0,1668523070107;-0,129161637436363;0,04899...

Масив2

D10:D11

↑

=

{0,511582545758248;-0,124958530099209}

=

{0,101498576345426;0,0567359306867897}

Повертає добуток матриць, які зберігаються у двох масивах, - масив із кількістю рядків першого масиву і кількістю стовпців другого масиву.

Масив2 перший з перемножуваних масивів; кількість його стовпців має дорівнювати кількості рядків другого масиву.

Значення: 0,101498576

Довідка з цієї функції

OK

Скасувати

Результати обчислень показано на рисунку

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	$f1(x,y)=2*x^2-x*y-5*x+1$				$f1'x=4*x-y-5$		$f1'y=-x$		$f2'x=1+3/(x*ln10)$			$f2'y=-2*y$	
2	$f2(x,y)=x+3*lgx-y^2$												
3													
4		X		F(x)		F'(x)			$F'(x)^{-1}$			$F'(x)^{-1}*F(X)$	
5													
6			3	-2		5	-3		0,254824	-0,19112		-0,59208908	
7	k=0	2		0,431364		1,434294	-4		0,091373	-0,31853		-0,32014847	
8													
9													
10		3,592089		0,511583		7,048208	-3,59209		0,166852	-0,12916		0,10149858	
11	k=1	2,320148		-0,12496		1,362709	-4,6403		0,048999	-0,25343		0,05673593	
12													

11.Об'єднаємо комірки A14:A15 і внесемо запис $k=2$.

12. Внесемо формули в комірки B14 та B15

$$B14:= B10-L10$$

$$B15:=B11-L11$$

Результати обчислень видно на рисунку

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	$f_1(x,y)=2*x^2-x*y-5*x+1$				$f_1'x=4*x-y-5$		$f_1'y=-x$			$f_2'x=1+3/(x*\ln 10)$		$f_2'y=-2*y$	
2	$f_2(x,y)=x+3*\lg x-y^2$												
3													
4		X		F(x)		F'(x)			$F'(x)^{-1}$			$F'(x)^{-1}*F(X)$	
5													
6		3		-2		5	-3		0,254824	-0,19112		-0,59208908	
7	k=0	2		0,431364		1,434294	-4		0,091373	-0,31853		-0,32014847	
8													
9													
10		3,592089		0,511583		7,048208	-3,59209		0,166852	-0,12916		0,10149858	
11	k=1	2,320148		-0,12496		1,362709	-4,6403		0,048999	-0,25343		0,05673593	
12													
13													
14		3,490591											
15	k=2	2,263413											
16													

Повторимо процес ще один раз для досягнення вищої точності.