

## ДОВГА АРИФМЕТИКА В EXCEL. ІІІ. РОЗВ'ЯЗУВАННЯ СЛАР ДЛЯ ТЕСТОВИХ ЗАДАЧ СКІНЧЕННО-ЕЛЕМЕНТНОГО МОДЕЛЮВАННЯ

В. Фурман, М. Хом'як, Я. Марко

*Львівський національний університет імені Івана Франка,  
вул. Грушевського, 4, 79005, Львів, Україна  
fourman@i.ua*

Клас раціональних чисел та алгоритми точних обчислень з використанням довгих арифметичних операцій адаптовано для розв'язування систем лінійних алгебричних рівнянь. Реалізовано простий алгоритм виключення Гауса, що ілюструє типові обчислювальні процедури й затрати комп'ютерних ресурсів, особливо, часу розрахунків. Одним з практичних застосувань цього алгоритму є розв'язування тестових задач і верифікація програмного забезпечення, наприклад, для моделювання механіки твердих тіл методом скінчених елементів. Розглянуто приклади як розріджених матриць, так і заповнених симетричних матриць з раціональними коефіцієнтами, таких як погано-визначені матриці Гілберта (випадок жорстких систем рівнянь розмірністю до 200). Числові експерименти підтвердили очікувану аналітичну точність і надійність запропонованого програмного коду.

*Ключові слова:* надбудова Excel, символічні обчислення, системи лінійних алгебричних рівнянь з раціональними коефіцієнтами, моделювання, МСЕ, тестування і верифікація програм.

Реалізація довгої арифметики в Excel включає операції над цілими та раціональними числами довільної розрядності [1]. Одним з практичних застосувань запропонованого програмного забезпечення (ПЗ) для точних обчислень є розв'язування систем лінійних алгебричних рівнянь (СЛАР). Як правило, прямі та ітераційні алгоритми [2, 3], працюють з подвійною точністю, а для калібрування цих методів і тестування нового ПЗ важливо мати не тільки теоретичні оцінки похибки, але й точні тестові розв'язки. Щоб апробувати можливості символічних обчислень в Excel з використанням класу `SRational`, ми реалізували алгоритм Гауса для розв'язування СЛАР. Метою є проаналізувати на прикладах можливості Excel для отримання точних розв'язків у прикладних задачах скінченно-елементного моделювання, де виникають такі системи.

Теоретично, якщо вхідні дані представлені як раціональні числа, то розв'язок системи отримаємо теж у класі раціональних чисел. Зрозуміло, що вже для СЛАР середніх розмірностей (порядку 100–1000 рівнянь) незалежно від розрядності початкових значень матриці та вектора правої частини, треба очікувати збільшення розрядності результату (знаменника і чисельника раціонального числа). Попередня оцінка (згідно з правилами арифметичних дій з двома дробами) передбачає подвоєння

розрядності на кожному етапі виключення однієї змінної – як у геометричній прогресії. Отже, не очевидно, чи взагалі в Excel вдасться отримати результат, і за прийнятний час, не вичерпавши ресурси програми й системи в цілому. Для цього потрібні числові експерименти і DEBUG-версія алгоритму Гауса, що дає змогу виводити стан матриці і вектора правої частини (ВПЧ) у клітинках електронних таблиць після деякого етапу виключення, щоб простежити накопичення розрядів. Також має бути можливість продовжити обчислення з цього етапу, не повторюючи вже виконану роботу.

**Алгоритм виключення Гауса.** Розглянемо СЛАР, записану в матричному вигляді  $AX = F$ ,

де  $A = [a_{ij}]$ ,  $X = [x_j]$ ,  $F = [f_i]$ ,  $i, j = 1, \dots, N$ . Щодо квадратної матриці  $A$  зробимо припущення про її невинудженість,  $\det A \neq 0$ , що гарантує існування та однозначність розв'язку. Найпростіший алгоритм виключення Гауса, без урахування симетрії та розрідженості матриці, а також перевірок на виродженість матриці та перестановок (за відсутності діагонального переважання) має вигляд [2, 4, 5]:

▪ *прямий хід*: для  $i = 1, 2, \dots, N-1$  виконати  $a_{kj} := a_{kj} - (a_{ki} / a_{ii})a_{ij}$ ,  $f_k := f_k - (a_{ki} / a_{ii})f_i$ ,  $\{j, k\} = i+1, \dots, N$ ;

▪ *обернений хід*: для  $i = N, N-1, \dots, 1$  виконати

$$x_i := \left( x_i - \sum_{j=i+1}^N a_{ij}x_j \right) / a_{ii} \quad (3)$$

Процес виключення полягає в приведенні матриці до верхнього трикутного вигляду (з нульовими елементами під головною діагоналлю), тобто в покроковому зануленні всіх елементів  $i$ -го стовпця нижче головної діагоналі, а також лінійній модифікації усіх елементів нижче і праворуч від діагонального елемента  $a_{ii}$ , причому  $a_{kj} = 0$ , якщо  $j \leq i$  і  $k > i$ . На місці логічно нульових елементів  $i$ -го стовпця на  $i$ -ому кроці виключення зручно зберігати результат ділення  $a_{ki} / a_{ii}$  – нормовані за діагональним елементом  $a_{ii}$  значення, які використовуються для модифікації решти елементів рядка. Тому після прямого ходу трикутна частина матриці нижче головної діагоналі міститиме проміжні результати, які надалі не використовуються. Отже, коефіцієнти початкової матриці не зберігаються, а на місці правої частини отримуємо вектор розв'язку.

Реалізація VBA-процедури для описаного методу Гауса представлено у Лістингу 1. Алгоритм використовує дії над об'єктами класу раціональних чисел `SRational`, а також здійснює логічні перевірки деяких нульових елементів. Якщо матриця СЛАР значно розріджена (містить багато нульових елементів, які не зазнають заповнення), то ці "перевірки на нуль" (`IsZero`) теоретично пришвидшують роботу алгоритму, оскільки тоді нема потреби викликати функції для декількох громіздких операцій, що включають множення і ділення. Навпаки, наприклад, для заповнених матриць, додаткові перевірки тільки сповільнять хід розв'язування. Зауважимо, що і без вказаних перевірок функції для арифметичних операцій з раціональними числами-об'єктами класу `SRational` виконуються коректно і ефективно (задіяно внутрішні перевірки для нульових значень).

```
Public Sub SRMSolveGauss(oMtr() As SRational, oRHS() As SRational)
' oMtr - square matrix [(0..n) x (0..n) ], oRHS - vector [(0..n) x (0..1)].
Dim i As Long, j As Long, k As Long, n As Long
n = UBound(oMtr(), 1) ' rows = cols
For i = 0 To n ' direct way:
  For k = i + 1 To n
    If Not oMtr(k, i).IsZero Then
      Set oMtr(k, i) = oMtr(k, i).Divide(oMtr(i, i), bSimplify:=True)
      For j = i + 1 To n
        If Not (oMtr(i, j).IsZero) Then Set oMtr(k, j) = _
          oMtr(k, j).X_YZ(oMtr(k, i), oMtr(i, j), bSimplify:=True)
      Next j
      If Not oRHS(i, 0).IsZero Then Set oRHS(k, 0) = _
        oRHS(k, 0).X_YZ(oMtr(k, i), oRHS(i, 0), bSimplify:=True)
    End If
  Next k
Next i
For i = n To 0 Step -1 ' inverse way:
  For j = i + 1 To n
    If Not oMtr(i, j).IsZero And Not oRHS(j, 0).IsZero Then _
      Set oRHS(i, 0) = oRHS(i, 0).X_YZ(oMtr(i, j), oRHS(j, 0), bSimplify:=True)
  Next j
  Set oRHS(i, 0) = oRHS(i, 0).Divide(oMtr(i, i), bSimplify:=True)
Next i
End Sub
```

Лістинг 1. Імплементація алгоритму Гауса для СЛАП з раціональними коефіцієнтами.

Розрахунки конкретних систем рівнянь дають змогу протестувати розроблені алгоритми і код програм, а також оцінити робочі параметри, як от загальний час виконання та взаємодію в середовищі Excel (зокрема, для матриць великих розмірностей ми рекомендуємо відключити опцію автоматичного перерахунку функцій робочого аркуша).

**Приклад 1. Симетричні матриці методу скінченних елементів (МСЕ).** Перш ніж розв'язувати конкретну систему рівнянь, коротко опишемо ситуацію, коли запропонований точний метод розв'язування має свої переваги і право на застосування.

Розглянемо абстрактну варіаційну задачу [6]

$$(Au, u) - 2(f, u) \rightarrow \min, \quad (4)$$

де  $A$  – лінійний додатно-визначений симетричний оператор,  $f$  – функція зовнішнього навантаження,  $u$  – шукана вектор-функція, що задовольняє головні граничні умови.

Застосування МСЕ дає вирішальну систему рівнянь:

$$[A_h]\{X\} = \{F_h\}, \quad (5)$$

де  $[A_h] = [N \times N]$  – матриця (дискретний аналог оператора  $A$ ), що є сумою матриць для кожного з елементів заданого розбиття  $h$ ,  $[A_h] = \sum [A_h^i]$ ,  $i = 1, 2, \dots, N$  – індекс елемента. Аналогічно для ВПЧ:  $\{F_h\} = \sum \{F_h^i\}$ . Вектор невідомих коефіцієнтів  $\{X\} = [C_1, C_2, \dots, C_N]^T$  визначає розв'язок задачі (4) у вигляді лінійної комбінації виду  $u = \sum C_i \varphi_i(x)$ , де  $\varphi_i(x)$  – вибрана система базисних функцій. Залежно від конкретного оператора  $A$ , для знаходження вкладів елемента  $[A_h^i]$  та  $\{F_h^i\}$  потрібно знаходити інтеграли, що містять добутки базисних функцій та їхніх похідних у різних комбінаціях. Якщо коефіцієнти оператора стали, то для поліноміальних базисних функцій такі інтеграли можна обчислити аналітично, наприклад, використовуючи онлайн-калькулятори для символьних обчислень [7]. У деяких випадках, наприклад, у тестових задачах, межі елементів можна вважати цілими чи, принаймні, раціональними числами, тобто відомими точно. Тоді й результат інтегрування буде раціональним числом, що ілюструє такий простий приклад:

$$\int_a^b x^2 dx = \frac{x^3}{3} \Big|_a^b = \frac{b^3 - a^3}{3} = \frac{2^3 - 1^3}{3} = \frac{7}{3} = 2\frac{1}{3}, \quad x \in [a, b] = [1; 2].$$

Нехай для програмного забезпечення, яке реалізує новий варіант МСЕ, потрібно створити тестові приклади для верифікації достовірності отримуваних результатів. Наш приклад стосується моделювання згину пластини постійної товщини, податливої до зсувних деформацій, зі сталими пружними властивостями, за умов плоскої деформації, що дає змогу розглядати задачу як одновимірну [8, 9]. Постановка тестової задачі також включає навантаження постійним тиском та умови шарнірного закріплення на краях. Необхідно знайти прогини пластини  $w(x)$  та кути повороту точок її серединної поверхні  $\gamma(x)$ . Нехай область пластини займає відрізок  $x \in [0, L] = [0, 10]$ , що розбито на  $L/h = N = 10$  елементів одиничної довжини. Кожну з цих функцій апроксимуємо лінійною комбінацією виду  $[w, \gamma] = \sum [C_i^w, C_i^\gamma] \varphi_i(x)$  з використанням відкритих бета-сплайнів як базисних функцій  $\varphi_i(x)$  [10]. Зокрема, квадратичні бета-сплайни матимуть компактний носій довжини 3, крім двох перших та останніх, що відмінні від нуля на відрізках довжиною 1 і 2 відповідно (рис. 1).

$$\varphi_1(x) = \begin{cases} \frac{x^2}{2}, & 0 < x \leq 1; \\ 0, & x < 0 \wedge x > 1. \end{cases} \quad \varphi_2(x) = \begin{cases} \frac{-x}{2}(3x-4), & 0 < x \leq 1; \\ \frac{1}{2}(x-2)^2, & 1 < x \leq 2; \\ 0, & x < 0 \wedge x > 2. \end{cases} \quad \varphi_3(x) = \begin{cases} \frac{x^2}{2}, & 0 < x \leq 1; \\ -x^2 + 3x - \frac{3}{2}, & 1 < x \leq 2; \\ \frac{1}{2}(x-3)^2, & 2 < x \leq 3; \\ 0, & x < 0 \wedge x > 3. \end{cases}$$

$$\varphi_{2+i}(x) = \varphi_3(x-i+1), \quad i = 1, \dots, 8; \quad \varphi_{11}(x) = \varphi_2(-x+8), \quad \varphi_{12}(x) = \varphi_1(-x+9). \quad (6)$$

Тому кількість невідомих коефіцієнтів  $[C_i^w, C_i^y]$ ,  $i = 1, 2, \dots, 12$  дорівнюватиме  $2(N + 2) = 24$ .

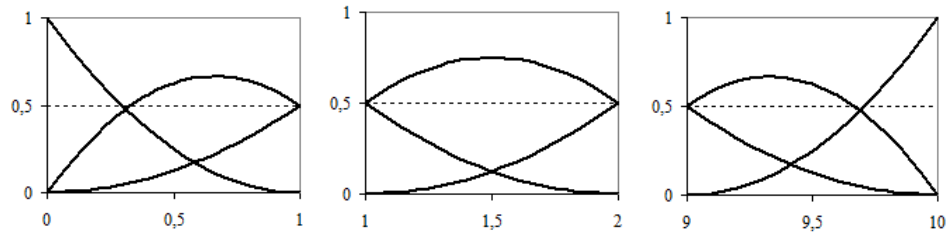


Рис. 1. В-сплайни степеня 2 для відрізка  $[0, 10]$  з кроком розбиття  $h = 1$ .

Матриці і ВПЧ для кожного з  $N = 10$  елементів-відрізків матимуть значення, які зручно подати в нормованому вигляді зі спільним знаменником 120:

$$A_1 = \frac{1}{120} \begin{bmatrix} 184 & -60 & -106 & 50 & -38 & 10 \\ -60 & 160 & -50 & -120 & -10 & -40 \\ -106 & -50 & 154 & 15 & 12 & 35 \\ 50 & -120 & 15 & 120 & -5 & 0 \\ -38 & -10 & 12 & -5 & 46 & 15 \\ 10 & -40 & 35 & 0 & 15 & 40 \end{bmatrix}, F_1 = \frac{1}{120} \begin{bmatrix} 0 \\ 40 \\ 0 \\ 60 \\ 0 \\ 20 \end{bmatrix},$$

$$A_i = \frac{1}{120} \begin{bmatrix} 46 & -15 & -7 & 10 & -19 & 5 \\ -15 & 40 & -40 & -20 & -5 & -20 \\ -7 & -40 & 94 & 0 & -7 & 40 \\ 10 & -20 & 0 & 40 & -10 & -20 \\ -19 & -5 & -7 & -10 & 46 & 15 \\ 5 & -20 & 40 & -20 & 15 & 40 \end{bmatrix}, F_i = \frac{1}{120} \begin{bmatrix} 0 \\ 20 \\ 0 \\ 40 \\ 0 \\ 20 \end{bmatrix}, i = 2, \dots, 9,$$

$$A_{10} = \frac{1}{120} \begin{bmatrix} 46 & -15 & 12 & 5 & -38 & 10 \\ -15 & 40 & -35 & 0 & -10 & -40 \\ 12 & -35 & 154 & -15 & -106 & 50 \\ 5 & 0 & -15 & 120 & -50 & -120 \\ -38 & -10 & -106 & -50 & 184 & 60 \\ 10 & -40 & 50 & -120 & 60 & 160 \end{bmatrix}, F_N = \frac{1}{120} \begin{bmatrix} 0 \\ 20 \\ 0 \\ 60 \\ 0 \\ 40 \end{bmatrix}.$$

Згідно зі стандартною процедурою МСЕ, вклади цих матриць потрібно розіслати в глобальну  $24 \times 24$ -матрицю, щоразу збільшуючи стартові індекси рядків та стовпців на 2 і насумовуючи. Аналогічно для ВПЧ. Отриману СЛАР та її точний розв'язок у класі раціональних чисел показано на рис. 2. Другий та останній рядки й стовпці містять нулі,

крім одиниці на головній діагоналі, що зумовлено виконанням головних граничних умов  $w(0) = w(L) = 0$ , які зумовлюють значення коефіцієнтів  $C_1^w = C_{12}^w = 0$ .

Для перевірки правильності розв'язування можна використати розроблені функції матричного множення (матриці  $[A_h]$  на вектор-розв'язок  $\{X\}$ ) та почленного віднімання від ВПЧ  $\{F_h\}$ , одночасно тестуючи їх. Очікувано отримуємо нуль-вектор:

$$[A_h]\{X\} - \{F_h\} = 0. \tag{7}$$

Зауважимо, що формат виводу включає виділення цілої частини та розрядку пробілами між знаками арифметичних операцій, що зручно для візуального аналізу.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	
1	23/15		-53/60	5/12	-19/60	1/12																			0	
2		1																								0
3	-53/60		5/3		1/24	3/8	-19/120	1/24																		0
4	5/12			4/3	-3/8	-1/6	-1/24	-1/6																		2/3
5	-19/60		1/24	-3/8	31/20		-7/60	5/12	-19/120	1/24																0
6	1/12		3/8	-1/6		1	-5/12	-1/3	-1/24	-1/6																1
7		-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24															0
8		1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6															1
9				-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24													0
10				1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6													1
11						-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24											0
12						1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6											1
13							-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24										0
14							1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6										1
15									-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24								0
16									1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6								1
17										-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24							0
18										1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6							1
19											-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24						0
20											1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6						1
21												-19/120	-1/24	-7/60	-5/12	31/20		-7/60	5/12	-19/120	1/24					0
22												1/24	-1/6	5/12	-1/3		1	-5/12	-1/3	-1/24	-1/6					2/3
23																		-19/60	-1/12	-53/60	-5/12	23/15				0
24																								1		0

Z	AE
.41 - 2 / 3	-41,67
0	0,00
.41 - 35214458498671 / 57521042850406	-41,61
23 + 91043779314001 / 172563128551218	23,53
.37 - 6684674109767 / 172563128551218	-37,04
67 + 33157178506673 / 172563128551218	67,19
.28 - 164681767885771 / 172563128551218	-28,95
103 + 75531761231483 / 172563128551218	103,44
.18 - 21458599441931 / 57521042850406	-18,37
129 + 46274718704707 / 172563128551218	129,27
.6 - 50203972052891 / 172563128551218	-6,29
142 + 117947042642399 / 172563128551218	142,68
6 + 50203972052891 / 172563128551218	6,29
142 + 117947042642399 / 172563128551218	142,68
18 + 21458599441931 / 57521042850406	18,37
129 + 46274718704707 / 172563128551218	129,27
28 + 164681767885771 / 172563128551218	28,95
103 + 75531761231483 / 172563128551218	103,44
37 + 6684674109767 / 172563128551218	37,04
67 + 33157178506673 / 172563128551218	67,19
41 + 35214458498671 / 57521042850406	41,61
23 + 91043779314001 / 172563128551218	23,53
41 + 2 / 3	41,67
0	0,00

Рис. 2. Скріншоти (Screenshots) матриці і ВПЧ (зверху) та вектора-розв'язку в раціональних та десяткових дробах (знизу). Порожні клітинки еквівалентні нульовим значенням.

**Приклад 2. Системи з "жорсткими" матрицями.** Розглянемо матрицю Гілберта  $[H] = [H_{ij}]_{i,j=1}^N$ , де  $H_{ij} = 1/(i+j-1)$ . Це приклад симетричної погано обумовленої матриці з раціональними коефіцієнтами, що часто використовується для тестування чутливості алгоритмів розв'язування СЛАР до похибок заокруглення [2, 11]. Запропонований алгоритм не має похибок заокруглення, "платою" за що є стрімке збільшення розрядності чисельника і знаменника коефіцієнтів матриці в процесі виключення Гауса (2) і (3). Як вектор навантаження (ВПЧ), ми брали  $[F] = [F_i]_{i=1}^N$ , де  $F_i = 1/i$  або  $F_i = \sum_{j=1, \dots, N} H_{ij}$ , тобто, коефіцієнти ВПЧ дорівнюють сумі рядків матриці.

Для цього випадку розроблено спеціальну тестову функцію, що динамічно формує матрицю Гілберта і ВПЧ для заданого  $N$ , без збереження коефіцієнтів у комірках таблиць. Для перевірки розв'язків використовували матричне множення та порівняння із заданим ВПЧ, згідно з рівнянням (7). Результати числових експериментів для  $N \in \{10; 25; 50; 100; 200\}$  включають деякі загальні показники, що характеризують складність системи, зокрема, фактичний час рахунку та максимальну кількість цифр чисельника або знаменника (Табл. 1). Накопичення розрядів в раціональних коефіцієнтах по ходу виконання методу Гауса відбувається досить швидко і для системи з  $N = 200$  склала 240 десяткових цифр у знаменнику, що пояснює недосяжність точних розв'язків внаслідок заокруглення в разі використання подвійної точності.

Таблиця 1.

Характеристики тестових систем рівнянь з матрицями Гілберта.

Порядок матриці, $N$		10	25	50	100	200
Сумарна кількість десяткових цифр чисельника і знаменника, крім нульових елементів	початкова матриця	254	1829	7454	35004	155004
	матриця після прямого ходу	423	5680	43491	343084	2726882
Максимальна кількість цифр (після прямого ходу)	чисельника	4	10	21	41	87
	знаменника	11	29	59	120	240
Загальний час рахунку, с		0,05	1,42	23,73	1337,01	37335,86

Ми зупинилися на  $N = 200$ , оскільки затрати часу ставали неприйнятно великими, хоча в усіх випадках запропонована реалізація алгоритму Гауса знаходила точний розв'язок. Оскільки матриці Гілберта є заповненими, то можна припустити, що для розріджених матриць МСЕ за аналогічний час можна було б отримати розв'язок для дещо більших розмірностей, орієнтовно  $N \sim 500$ , що достатньо для простих тестових задач МСЕ.

Треба відмітити, що час рахунку суттєво зменшується, якщо використовувати потужнішу техніку. В нашому розпорядженні були 2 типові комп'ютери з 64-розрядною системою Windows: 1) 2-процесорний Intel(R) Celeron(R) N2840 із частотою 2,16 ГГц і

2 Гб ОП; 2) 4-процесорний Intel(R) Core(TM) i3 із частотою 3,20 ГГц і 8 Гб ОП. Для цих комп'ютерів (перший – нижчого рівня, другий – потужніший) загальний час рахунку зменшився в 3 рази, хоча часові профілі, показані для  $N = 200$  на рис. 3, мають подібну форму. Закономірність витрат часу на кожне рівняння полягає в тому, що час швидко наростає для першої чверті з  $N$  рівнянь, довго залишається на високому рівні (наступна половина з  $N$  рівнянь) і, нарешті, процес стрімко закінчується для останньої чверті. Це підтверджує, що для розв'язання громіздких задач критично необхідні потужні комп'ютерні ресурси, зокрема кількість процесорів, їхня швидкодія та об'єм оперативної пам'яті. Разом з тим, підвищення ефективності програмної реалізації запропонованих алгоритмів точних обчислень з використанням раціональних чисел у таблицях Excel може ще зменшити час рахунку та збільшити досяжний порядок  $N$ , а, отже, збільшити комфортність таких обчислень.

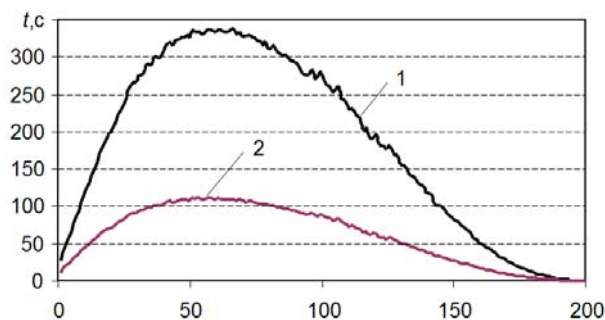


Рис. 3. Залежність часу, затраченого на один цикл прямого ходу методу Гауса, від номера рівняння для  $N = 200$ . Графіки відповідають двом протестованим комп'ютерам: 1 – "повільний", 2 – "швидкий" (пояснення в тексті).

Наприкінці ще зауважимо, що з використанням раціональних чисел можна запропонувати й інші алгоритми для електронних таблиць, наприклад, ітераційні алгоритми для СЛАР або символічне інтегрування означених інтегралів.

#### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Фурман В. Довга арифметика в Excel. II. Раціональні числа / В. Фурман, М. Хом'як, Я. Марко // Електроніка та інформаційні технології. – 2019. – Вип. 11. – С. 99–108. – Режим доступу: <http://elit.lnu.edu.ua/issue.php?lang=&number=11>.
2. Харів А. Паралельні методи розв'язання СЛАР з використанням архітектури CUDA / А. Харів, І. Хвищун [Електронний ресурс] // Електроніка та інформаційні технології. – 2015, 5. – С. 185–191. – Режим доступу: [http://elit.lnu.edu.ua/pdf/5\\_20.pdf](http://elit.lnu.edu.ua/pdf/5_20.pdf).
3. He H. Solution of linear systems by GMRES method on global computing platform / He H., Bergere G., Wang Z., Petiton S. // Journal of Algorithms & Computational Technology. – 2007, Vol. 1, No. 4. – P. 449–465. – Mode of access: <https://journals.sagepub.com/toc/acta/1/4>.
4. Гантмахер Ф. Р. Теория матриц: -4-е изд. – М. Наука. Физматлит, 1988. – 552 с.



5. *Cormen T.H.* Introduction to Algorithms, 3rd Ed. / T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein. – MIT Press, 2009. – 1312 p.
6. *Савула Я. Г.* Числовий аналіз задач математичної фізики варіаційними методами / Я. Г. Савула. – Львів: видавничий центр ЛНУ імені Івана Франка, 2004. – 221 с.
7. Online Integral Calculator [Electronic resource]. – Mode of access: <https://www.wolframalpha.com/calculators/integral-calculator/>
8. *Марчук М. В.* Змішана схема методу скінченних елементів для розрахунку шаруватих композитних оболонок і пластин / М.В Марчук., М.М. Хом'як. – Львів: Національна академія наук України. Інститут прикладних проблем механіки і математики імені Я.С. Підстригача, 2003. – 216 с.
9. *Хом'як М.* Тестування схем МСЕ з використанням B-сплайнів для теорії пластин Кірхгофа-Лява та С.П. Тимошенка / Хом'як Микола, Марко Ярослав // Сучасні проблеми механіки та математики: збірник наукових праць у 3-х т. / за заг. ред. А.М. Самойленка та Р.М. Кушніра [Електронний ресурс] // Інститут прикладних проблем механіки і математики ім. Я.С. Підстригача НАН України. – 2018. – Т. 1. – С. 251–252. – Режим доступу: [www.iapmm.lviv.ua/mpmm2018/Volume\\_1.pdf](http://www.iapmm.lviv.ua/mpmm2018/Volume_1.pdf).
10. *Aggarwal B.* B-spline finite elements for plane elasticity problems. – Texas A&M University, 2006. – 80 p. – Mode of access: <https://core.ac.uk/download/pdf/4271776.pdf>
11. *Nishi T.* On the generation of very ill-conditioned integer matrices [Electronic resource] / Nishi T., Rump S.M., Oishi S. // NonLinear Theory and its applications, IEICE. – 2011, Vol. 2, No 2. –P. 226–245.– Mode of access: [https://www.jstage.jst.go.jp/article/nolta/2/2/2\\_2\\_226/\\_article](https://www.jstage.jst.go.jp/article/nolta/2/2/2_2_226/_article).

### LONG ARITHMETIC IN EXCEL. III. SOLUTION OF SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS FOR TEST PROBLEMS OF THE FINITE ELEMENT MODELLING

V. Fourman, M. Khomyak, Ya. Marko

*Ivan Franko National University of Lviv,  
Hrushevskogo Str. 4, UA-79005 Lviv, Ukraine,  
[fourman@i.ua](mailto:fourman@i.ua)*

The class of rational numbers and full precision algorithms are adopted to solve systems of the linear algebraic equations. The simple method of Gaussian elimination is implemented which shows typical computational procedures and costs of computer resources, especially the time of calculus.

One practical application of this algorithm is test tasks solving and software verifying. For example, the finite element method is often used in the solid mechanics modelling. Brief theoretical basis of the variational problem of bending of a complaint to shear plate under constant pressure is given. The perspective scheme of the FEM using B-splines is proposed.

For example, one-dimensional finite elements using quadratic opened B-splines as basis functions are considered. Typical matrices contain the product of the basis functions and their first derivatives in various combinations under the integral. In the case of a single length of these elements, the matrices are obtained analytically in the form of rational numbers. The assembly procedure for ten elements with two degrees of freedom on each is illustrated. The sparse global

matrix and right hand side size of the resulting system with rational coefficients are constructed and its exact solution is obtained.

Also examples of filled symmetric matrices are considered. A case of rigid systems with ill-conditioned Gilbert's matrices dimensionality up to  $N=200$  is examined. The maximum number of digits of a denominator is 240 for  $N=200$ , what explains the inaccessibility of exact solutions when using double precision.

Numeric simulation confirmed the expected analytical precision and the robustness of the proposed program code. In particular, the time cost and increasing of rational coefficients size during the direct way of the Gauss method are traced. We compared a "slow" and "fast" computers with 2 processors and 2 MB of RAM vs. 4 processors and 8 MB of RAM respectively. The calculation time was reduced by three times. This indicates that significant computer resources are critically required. On the other hand, long arithmetic using binary algorithms can be implemented more efficiently.

*Key words:* EXCEL Add-In, symbolic computations, systems of linear algebraic equations with rational coefficients, modelling, FEM, software testing and verifying.

*Стаття: надійшла до редакції 15.03.2019,  
доопрацьована 11.09.2019  
прийнята до друку 12.09.2019.*