Fazylova L.S., Akynova N.T.

E.A. Buketov named Karaganda University, Kazakhstan

Programming iterative methods for solving systems of linear algebraic equations

The discipline "Numerical Methods" refers to the major disciplines in the preparation of students of the specialties "5B060200 - Informatics (natural science)", "5B070300 - Information systems", "5B07000-Mathematical and computer modeling". The important tasks of this course are: to teach students to create a mathematical model of problems in computational mathematics and to apply algorithms for methods of finding solutions to these problems. Students should acquire skills in the numerical implementation of the studied methods using various programming environments.

In this paper, we consider the numerical implementation of the simple iteration method and the Zeidel method for solving systems of linear algebraic equations (SLAE) in an Excel spreadsheet processor and in the Python programming language. Based on the results obtained, a comparative analysis of these methods is carried out.

Formulation of the problem. Solve SLAE by the simple iteration method and the Zeidel method with an accuracy of 0.001:

$$\begin{cases} x_1 = 0,23x_1 - 0,04x_2 + 0,21x_3 - 0,18x_4 + 1,21; \\ x_2 = 0,45x_1 - 0,23x_2 + 0,06x_3 - 0,88; \\ x_3 = 0,26x_1 + 0,34x_2 - 0,11x_3 + 0,62; \\ x_1 = 0,05x_1 - 0,26x_2 + 0,34x_3 - 0,12x_4 - 1,17; \end{cases}$$

Let's consider the implementation of these methods in an Excel spreadsheet processor. The condition for the convergence of the iterative process is satisfied for the given system [1]. Let's enter the input data into the table (Fig. 1).

	A	B	C	D	E	F	G	H
1		0,23	-0,04	0,21	-0,18			1,24
2	A=	0,45	-0,23	0,06	0		B=	-0,88
3		0,26	0,34	-0,11	0			0,62
4		0,05	-0,26	0,34	-0,12			-1,17

Figure 1. Matrix of the system and the vector of the right side

Using the simple iteration method, we get the following result (Fig. 2):

k	X1(k)	X2(k)	X3(k)	X4(k)
0	1,24	-0,88	0,62	-1,17
1	1,9012	-0,0824	0,575	-0,528
2	1,896362	0,028992	1,023046	-0,79466
3	2,032881	0,028078	1,010376	-0,63953
4	2,033733	0,088961	1,046954	-0,65539
5	2,04203	0,077536	1,063853	-0,65683
6	2,048204	0,084911	1,060266	-0,64753
7	2,046902	0,085778	1,064774	-0,65147
8	2,048224	0,085263	1,064234	-0,64976
9	2,048126	0,085944	1,064462	-0,64995

Figure 2. Result of the simple iteration method

Applying Zeidel's method, we get the following result (Fig. 3):

k	X1(k)	X2(k)	X3(k)	X4(k)
0	1,24	-0,88	0,62	-1,17
1	1,9012	0,21514	1,11926	-0,60993
2	2,013502	0,043749	1,035267	-0,65552
3	2,036755	0,088593	1,065799	-0,65016
4	2,045757	0,084162	1,063274	-0,65006
5	2,047456	0,085794	1,064549	-0,64998
6	2,048035	0,085756	1,064546	-0,64995
7	2,048164	0,085823	1,064602	-0,64995

Figure 3. Result of the Zeidel method

For this problem, the method of simple iterations found a solution to the SLAE in 9 iterations, and the Zeidel method in 7 iterations. Conducting a comparative analysis of the two methods, we conclude that the Zeidel method converges to the exact solution of the SLAE faster than the simple iteration method.

Here is the implementation of the considered methods in the Python programming language.

Simple iteration method, program code:

```
x0 = float(0.23); x1 = float( -0.04); x2 = float(0.21); x3 = float(-0.18);
x4 = float(0.45); x5 = float( -0.23);    x6 = float(0.06); x7 = float(0);
x8 = float(0.26); x9 = float(0.34); x10= float(-0.11);    x11= float(0);
x12= float(0.05); x13= float(-0.26); x14= float(0.34); x15= float(-0.12);
b0 = float(1.24); b1 = float(-0.88); b2 = float(0.62); b3 = float(-1.17);
print("Метод простых итераций")
a = 0
print("%2d" % a, b0, " ", b1, " ", b2, " ", b3)
v = b0
b = b1
n = b2
m = b3
s = x0 * v + x1 * b + x2 * n + x3 * m + b0
d = x4 * v + x5 * b + x6 * n + x7 * m + b1
f = x8 * v + x9 * b + x10* n +x11 * m + b2
g = x12* v + x13* b + x14* n +x15 * m + b3
for i in range(0, 9):
    a += 1
    s = x0 * v + x1 * b + x2 * n + x3 * m + b0
    d = x4 * v + x5 * b + x6 * n + x7 * m + b1
    f = x8 * v + x9 * b + x10* n +x11 * m + b2
    g = x12* v + x13* b + x14* n +x15 * m + b3
    v = s
    b = d
    n = f
    m = g
    print("%2d" % a, round(s,5), round(d,5), round(f,5), round(g,5))
```

The result of the implementation of the simple iteration method is shown in Figure 4.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Nazerke/Desktop/itt.py ==========
Метод простых итераций
 0 1.24    -0.88    0.62    -1.17
 1 1.9012 -0.0824 0.575 -0.528
 2 1.89636 0.02899 1.02305 -0.79466
 3 2.03288 0.02808 1.01038 -0.63953
 4 2.03373 0.08896 1.04695 -0.65539
 5 2.04203 0.07754 1.06385 -0.65683
 6 2.0482 0.08491 1.06027 -0.64753
 7 2.0469 0.08578 1.06477 -0.65147
 8 2.04822 0.08526 1.06423 -0.64976
 9 2.04813 0.08594 1.06446 -0.64995
>>>
```

Figure 4.

Zeidel's method, program code:

```
o0 = float(0.23); o1 = float(-0.04); o2 = float(0.21); o3 = float(-0.18);
o4 = float(0.45); o5 = float( -0.23);    o6 = float(0.06); o7 = float( 0);
o8 = float(0.26); o9 = float(0.34); o10= float(-0.11);    o11= float(0);
o12= float(0.05); o13= float(-0.26); o14= float(0.34); o15= float(-0.12);
p0 = float(1.24); p1 = float(-0.88); p2 = float(0.62); p3 = float(-1.17);
print("\nМетод Зейделя")
k = 0
print("%2d" % k, p0, " ", p1, " ", p2, " ", p3)
e = p0
r = p1
t = p2
y = p3
q = o0 * e + o1 * r + o2 * t + o3 * y + p0
w = o4 * q + o5 * r + o6 * t + o7 * y + p1
u = o8 * q + o9 * w +o10 * t +o11 * y + p2
i = o12* q + o13* w +o14 * u +o15 * y + p3
for i in range(0,7):
    k += 1
    q = o0 * e + o1 * r + o2 * t + o3 * y + p0
    w = o4 * q + o5 * r + o6 * t + o7 * y + p1
```

```
u = o8 * q + o9 * w +o10 * t +o11 * y + p2
i = o12* q + o13* w +o14 * u +o15 * y + p3
e = q
r = w
t = u
y = i
print("%2d" % k, round(q,5), round(w,5), round(u,5), round(i,5))
```

The result of the implementation of the Seidel method is shown in the figure 5.

```
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/Nazerke/Desktop/zeid.py ==========

Метод Зейделя
 0 1.24    -0.88    0.62    -1.17
 1 1.9012 0.21514 1.11926 -0.60993
 2 2.0135 0.04375 1.03527 -0.65552
 3 2.03675 0.08859 1.0658 -0.65016
 4 2.04576 0.08416 1.06327 -0.65006
 5 2.04746 0.08579 1.06455 -0.64998
 6 2.04803 0.08576 1.06455 -0.64995
 7 2.04816 0.08582 1.0646 -0.64995
>>>
```

Figure 5.

Comparing the results obtained in an Excel spreadsheet processor and in the Python programming language, we conclude that the results are the same.

The advantage of the Python programming language is that it is freely available and easily solves a computer problem, such as writing your solution. The code can be written once, and the program can be run on almost any computer without changing it.

References

1. Kireev V.I. Numerical methods in examples and problems: textbook. allowance/ V. I. Kireev, A. V. Panteleev. - Ed. 2nd, erased. - M.: Higher. shk., 2006. - 480 p.

2. http://pythontutor.ru/