

Часть № 2 «Решение СЛАУ в Python и Maple»

1. Сформировать квадратную матрицу и вектор свободных членов в Maple, используя команду **seq** и заданные функции. Записать матрицу в один файл, а вектор свободных членов в другой файл.
2. Используя Maple, прочитать данные из файлов и решить СЛАУ. Выполнить проверку. Записать решение в файл xMaple.txt.
3. Используя Python, прочитать данные из файлов и решить СЛАУ. Записать решение в файл xPython.txt. Прочитать данные из файлов xMaple.txt, xPython.txt и выполнить сравнение результатов решений, полученных с помощью Maple и Python.
4. К каждой строчке кода написать комментарий. Оформить раздел отчета.

Полезные ссылки:

1. fopen, <https://www.maplesoft.com/support/help/maple/view.aspx?path=fopen>
2. writedata, <https://www.maplesoft.com/support/help/maple/view.aspx?path=writedata>
3. fclose, <https://www.maplesoft.com/support/help/maple/view.aspx?path=fclose>
4. convert,
<https://www.maplesoft.com/support/help/maple/view.aspx?path=DifferentialGeometry%2FConvert>
5. numpy.fromfile, https://pyprog.pro/array_creation/fromfile.html
6. numpy.reshape, https://pyprog.pro/array_manipulation/reshape.html
7. ndarray.tofile, https://pyprog.pro/io_functions/ndarray_tofile.html

Фрагмент программы в Maple

```
restart;  
f1 := (x, y) -> (-1)^x*cos(y*x);  
f2 := x -> (-1)^x*x^2;  
n := 10:  
  
with(LinearAlgebra);  
a := evalf([seq([seq(f1(x, y), x = 1 .. n)], y = 1 .. n)]);  
A := convert(a, Matrix);  
# Число обусловленности матрицы характеризует, насколько велика  
# погрешность решения.  
# Если число обусловленности между 1 и 100, то матрица  
# хорошо обусловлена.  
Norm(A)/Norm(A^-1);  
. . .
```

Программа в Python:

```
import numpy as np

a = np.fromfile('d:\\A.txt', sep = '\t')
n=int((len(a))*(1/2))
a=np.reshape(a, (n, n))
print(a)
print(type(a))

b = np.fromfile('d:\\b.txt', sep = '\n')
print(b)
print(type(b))

x=np.linalg.solve(a, b)
print(x)

x.tofile('d:\\xPython.txt', sep='\n', format='%f')

xp = np.fromfile('d:\\xPython.txt', sep = '\n')
xm = np.fromfile('d:\\xMaple.txt', sep = '\n')

er =xp-xm
er.tofile('d:\\er.txt', sep='\n', format='%f')

print(er)
print(sum(er))
```

Варианты

№	Функции для формирования последовательностей
1	$\cos(x^2 - 2*y*x + y^2), \sin(x)$
2	$\cos(x^3 + y^3 - 3*x*y), \cos(x)$
3	$\cos(x^4 + y^4 - 4*x*y), \sin(x)$
4	$\cos(-125*x^3*y^3 + x^5 + y^5 - 5*x^2*y^2), \cos(x)$
5	$\sin(-12*x^5*y^5 + x^6 + y^6), x*\cos(x)$
6	$\cos(-5*x^5*y^5 + x^3 + y^3), x*\sin(x)$
7	$\cos(-353*x^5*y^5 + x^7 + y^7), x*\sin(x)$
8	$\cos(x^8 - 1021*x^4*y^4 + y^8), x*\sin(x)$
9	$\sin(-13*x^4*y^4 + y^3 + x^2), x*\cos(x)$
10	$\sin(x^3 - 128*x*y + y^2), x*\cos(x + 2)$
11	$\sin(x^4 + y^3 - 256*x*y), x*\cos(x^2)$
12	$\cos(x^4 + y^3 - 256*x*y), x*\cos(2*x)$
13	$\cos(x^7 + y^7 - 890*x*y), x*\cos(x^3)$
14	$\sin(x^8 + y^7 - 546*x*y), x*\cos(x^4)$
15	$\sin(x^6 + y^5 - 8*x^2*y), x*\sin(x^2)$
16	$\sin(14*x^4 - 4*x^2*y + y^2), x*\sin(x + 2)$
17	$\cos(x^9 - 137*x^2*y^2 + y^4), x*\sin(120*x)$
18	$\sin(-x^2*y^2 + x + y), x^2*\cos(x)$
19	$\cos(x^4 - x^2*y^2 + y^2), x*\sin(10*x)$
20	$\cos(-x^3*y^3 + x^4 + y^4), x*\cos(10 + x)$
21	$\sin(x^4 - x^2*y^2 + y^4), x*\cos(x - 10)$
22	$\cos(y^3 - 13*x*y + x), x*\sin(x^2 - 10)$
23	$\cos(y^3 + x^2 - 26*x*y), x*\sin(x^2 - 10*x)$
24	$\cos(x^4 + y^3 - 173*x*y), x*\sin(5*x)$
25	$(-1)^x*\cos(x^2 - 2*x*y + y^2), x*\sin(5*x)$
26	$(-1)^y*\sin(x^2 - 5*x*y + y^2), x*\sin(5*x^2)$
27	$(-1)^{(x+y)}*\sin(x^3 - 3*x^2*y + y^3), x*\sin(x/2)$
28	$(-1)^{(x^2)}*\sin(-3*x^2*y + 12*x^2 + 16*y^2), x*\cos(x)$
29	$(-1)^{(y^2)}*\cos(\sin(x^2 - 18*x*y + y^2)), x^2$
30	$(-1)^{(y^2)}*\sin(\sin(x^2 - 180*x*y + y^2)), x^2$