

# **Отчёт по лабораторной работе №2**

по дисциплине «Вычислительная математика»

Студент гр. 3530202/90002

Потапова А.М.

Преподаватель

Леонтьева Т.В.

2021г

## Вариант №13

### Постановка задачи:

1. Составить процедуру вычисления матрицы  $A$  вида:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_1 & \dots & x_N \\ \dots & \dots & \dots & \dots \\ x_1^{N-1} & x_2^{N-1} & \dots & x_N^{N-1} \end{pmatrix}$$

2. Используя ее, решить несколько раз линейную систему  $Az = b$ , где  $N = 5, 7, 9$ ;  
 $b_k = 2^k + \cos(k)$ ,  $k = 1, \dots, N$ ;  
 $x_k = k$ ,  $k = 1, \dots, N-1$   
 $x_N = 1.1, 1.01, 1.001, 1.0001$ .  
Системы решать, используя программы **DECOMP** и **SOLVE**. Сделать необходимые выводы.

### Ход работы

1. В первую очередь была написана процедура вычисления матрицы  $A$ ;
2. Далее была написана функция, вычисляющая  $b_k$ ;
3. Стоит отметить, что в языке FORTRAN, с библиотек которого и написаны функции `smath`, двумерные массивы часто представляются по столбцам в виде одномерного массива, а индекс пересчитывается по формуле  $A_{ij} = A[i * \text{ndim} + j]$  (1), где  $\text{ndim}$  – размерность массива. Поэтому, специально для программ **DECOMP** и **SOLVE** написана функция `fortranStyle`, которая из двух двумерных массивов  $A$  и  $B$  размерности  $n$  делает одномерный массив  $a$  размерности  $n * n$  по формуле (1).
4. Затем стало возможным получить решение линейной системы  $Az = b$  при помощи программ **DECOMP** и **SOLVE** с учетом различных данных, которые были предоставлены в задании.

## Результаты:

$N = 5$

Консоль:

```
N = 5, Bk = 2^k + cos(k), k = 1 ... 5
=====

Bk:
b[1] = 2.5403
b[2] = 3.58385
b[3] = 7.01001
b[4] = 15.3464
b[5] = 32.2837

Matrix A (xn = 1.1):
  1 |   1 |   1 |   1 |   1 |
  1 |   2 |   3 |   4 |   1.1 |
  1 |   4 |   9 |  16 |   1.21 |
  1 |   8 |  27 |  64 |   1.331 |
  1 |  16 |  81 | 256 |   1.4641 |

Matrix A (xn = 1.01):
  1 |   1 |   1 |   1 |   1 |
  1 |   2 |   3 |   4 |   1.01 |
  1 |   4 |   9 |  16 |   1.0201 |
  1 |   8 |  27 |  64 |   1.0303 |
  1 |  16 |  81 | 256 |   1.0406 |

Matrix A (xn = 1.001):
  1 |   1 |   1 |   1 |   1 |
  1 |   2 |   3 |   4 |   1.001 |
  1 |   4 |   9 |  16 |   1.002 |
  1 |   8 |  27 |  64 |   1.003 |
  1 |  16 |  81 | 256 |   1.004 |

Matrix A (xn = 1.0001):
  1 |   1 |   1 |   1 |   1 |
  1 |   2 |   3 |   4 |   1.0001 |
  1 |   4 |   9 |  16 |   1.0002 |
  1 |   8 |  27 |  64 |   1.0003 |
  1 |  16 |  81 | 256 |   1.0004 |
```

Result.txt:

```
N = 5
=====

Cond: 54131.7
Solutions (xn = 1.1):
x[0] = 12.4967
x[1] = 1.74069
x[2] = 0.371622
x[3] = -0.080522
x[4] = -11.9882

Cond: 428093
Solutions (xn = 1.01):
x[0] = 101.671
x[1] = 1.44044
x[2] = 0.442377
x[3] = -0.0908062
x[4] = -100.923

Cond: 4.18068e+06
Solutions (xn = 1.001):
x[0] = 993.415
x[1] = 1.41339
x[2] = 0.449102
x[3] = -0.0918007
x[4] = -992.645

Cond: 4.17077e+07
Solutions (xn = 1.0001):
x[0] = 9910.85
x[1] = 1.41072
x[2] = 0.449771
x[3] = -0.0918998
x[4] = -9910.08
```

$N = 7$

Консоль

```
N = 7, Bk = 2^k + cos(k), k = 1 ... 7
=====

Bk:
b[1] = 2.5403
b[2] = 3.5839
b[3] = 7.01
b[4] = 15.346
b[5] = 32.284
b[6] = 64.96
b[7] = 128.75

Matrix B (xn = 1.1):
1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 1.1 |
1 | 4 | 9 | 16 | 25 | 36 | 1.21 |
1 | 8 | 27 | 64 | 125 | 216 | 1.331 |
1 | 16 | 81 | 256 | 625 | 1296 | 1.4641 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 1.61051 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 1.771561 |

Matrix B (xn = 1.01):
1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 1.01 |
1 | 4 | 9 | 16 | 25 | 36 | 1.0201 |
1 | 8 | 27 | 64 | 125 | 216 | 1.030301 |
1 | 16 | 81 | 256 | 625 | 1296 | 1.040604 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 1.05101 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 1.06152 |

Matrix B (xn = 1.001):
1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 1.001 |
1 | 4 | 9 | 16 | 25 | 36 | 1.002001 |
1 | 8 | 27 | 64 | 125 | 216 | 1.003003 |
1 | 16 | 81 | 256 | 625 | 1296 | 1.004006 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 1.00501 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 1.006015 |

Matrix B (xn = 1.0001):
1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 1.0001 |
1 | 4 | 9 | 16 | 25 | 36 | 1.0002 |
1 | 8 | 27 | 64 | 125 | 216 | 1.0003 |
1 | 16 | 81 | 256 | 625 | 1296 | 1.0004 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 1.0005 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 1.0006 |
```

Result.txt:

```
N = 7
=====

Cond: 1.81567e+07
Solutions (xn = 1.1):
x[0] = 5.12857
x[1] = -1.97223
x[2] = 3.52866
x[3] = -1.91276
x[4] = 0.59341
x[5] = -0.0804839
x[6] = -2.74487

Cond: 1.3017e+08
Solutions (xn = 1.01):
x[0] = 24.6377
x[1] = -2.08171
x[2] = 3.58026
x[3] = -1.93526
x[4] = 0.599679
x[5] = -0.0812818
x[6] = -22.1791

Cond: 1.26934e+09
Solutions (xn = 1.001):
x[0] = 219.729
x[1] = -2.09157
x[2] = 3.58517
x[3] = -1.93743
x[4] = 0.60029
x[5] = -0.08136
x[6] = -217.263

Cond: 1.26646e+10
Solutions (xn = 1.0001):
x[0] = 2170.64
x[1] = -2.09255
x[2] = 3.58565
x[3] = -1.93765
x[4] = 0.600351
x[5] = -0.0813678
x[6] = -2168.17
```

N = 9

Консоль

Result.txt:

```
N = 9, Bk = 2^k + cos(k), k = 1 ... 9
=====
Bk:
b[1] = 2.540302
b[2] = 3.583853
b[3] = 7.010008
b[4] = 15.34636
b[5] = 32.28366
b[6] = 64.96017
b[7] = 128.7539
b[8] = 255.8545
b[9] = 511.0889

Matrix C (xn = 1.1):
1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1.1 |
1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 1.21 |
1 | 8 | 27 | 64 | 125 | 216 | 343 | 512 | 1.331 |
1 | 16 | 81 | 256 | 625 | 1296 | 2401 | 4096 | 1.4641 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 16807 | 32768 | 1.61051 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 117649 | 262144 | 1.771561 |
1 | 128 | 2187 | 16384 | 78125 | 279936 | 823543 | 2097152 | 1.9487171 |
1 | 256 | 6561 | 65536 | 390625 | 1679616 | 5764801 | 16777216 | 2.14358881 |

Matrix C (xn = 1.01):
1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1.01 |
1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 1.0201 |
1 | 8 | 27 | 64 | 125 | 216 | 343 | 512 | 1.030301 |
1 | 16 | 81 | 256 | 625 | 1296 | 2401 | 4096 | 1.04060401 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 16807 | 32768 | 1.05101005 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 117649 | 262144 | 1.06152015 |
1 | 128 | 2187 | 16384 | 78125 | 279936 | 823543 | 2097152 | 1.07213535 |
1 | 256 | 6561 | 65536 | 390625 | 1679616 | 5764801 | 16777216 | 1.08285671 |

Matrix C (xn = 1.001):
1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1.001 |
1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 1.002001 |
1 | 8 | 27 | 64 | 125 | 216 | 343 | 512 | 1.003003 |
1 | 16 | 81 | 256 | 625 | 1296 | 2401 | 4096 | 1.004006 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 16807 | 32768 | 1.00501001 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 117649 | 262144 | 1.00601502 |
1 | 128 | 2187 | 16384 | 78125 | 279936 | 823543 | 2097152 | 1.00702104 |
1 | 256 | 6561 | 65536 | 390625 | 1679616 | 5764801 | 16777216 | 1.00802806 |

Matrix C (xn = 1.0001):
1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1.0001 |
1 | 4 | 9 | 16 | 25 | 36 | 49 | 64 | 1.00020001 |
1 | 8 | 27 | 64 | 125 | 216 | 343 | 512 | 1.00030003 |
1 | 16 | 81 | 256 | 625 | 1296 | 2401 | 4096 | 1.00040006 |
1 | 32 | 243 | 1024 | 3125 | 7776 | 16807 | 32768 | 1.00050001 |
1 | 64 | 729 | 4096 | 15625 | 46656 | 117649 | 262144 | 1.000600015 |
1 | 128 | 2187 | 16384 | 78125 | 279936 | 823543 | 2097152 | 1.000700021 |
1 | 256 | 6561 | 65536 | 390625 | 1679616 | 5764801 | 16777216 | 1.000800028 |

N = 9
=====
Cond: 1.4419e+10
Solutions (xn = 1.1):
x[0] = -3.27322
x[1] = -8.04376
x[2] = 11.4984
x[3] = -9.8966
x[4] = 5.99565
x[5] = -2.4051
x[6] = 0.572881
x[7] = -0.0613482
x[8] = 8.1534

Cond: 6.65749e+10
Solutions (xn = 1.01):
x[0] = -59.4435
x[1] = -7.60247
x[2] = 11.1864
x[3] = -9.66987
x[4] = 5.86931
x[5] = -2.35686
x[6] = 0.561755
x[7] = -0.0601836
x[8] = 64.0557

Cond: 7.04707e+11
Solutions (xn = 1.001):
x[0] = -621.146
x[1] = -7.56272
x[2] = 11.1568
x[3] = -9.64795
x[4] = 5.85699
x[5] = -2.35213
x[6] = 0.560661
x[7] = -0.0600688
x[8] = 625.735

Cond: 6.86085e+12
Solutions (xn = 1.0001):
x[0] = -6238.17
x[1] = -7.55878
x[2] = 11.1538
x[3] = -9.64576
x[4] = 5.85576
x[5] = -2.35166
x[6] = 0.560552
x[7] = -0.0600574
x[8] = 6242.76
```

## **Вывод:**

Рассматривая уже первый пример, где  $N = 5$ , можно отметить, что чем ближе  $x_N$  к 1, тем больше первая строка похожа на вторую, вторая на третью и так далее. Помимо этого, если обратить внимание на последующие примеры, где размерность матрицы  $N$  увеличивается ( $N = 7, N = 9$ ), мы возводим единицу с каким-либо значением в далеком разряде в степень, и в итоге это значение отодвигается еще на большее количество разрядов, т. е. мы наблюдаем линейную зависимость строк. Таким образом, мы не можем найти переменную, так как у нас получаются фактически одинаковые уравнения, поэтому у нас появляется бесконечное множество решений. Это так же можно наблюдать на примере роста числа обусловленности матрицы.

## Текст программы:

```
#include <math.h>
#include <iomanip>
#include <iostream>
#include <fstream>
#include "Forsythe.h"

std::ofstream out("result.txt");

void printBk(double **B, int n)
{
    for (int i = 0; i < n; i++)
    {
        B[i][0] = pow(2, i + 1) + cos(i + 1);
    }

    std::cout << "=====\n";
    std::cout << "N = " << n << ", Bk = 2^k + cos(k), k = 1 ... " << n << "\n";
    std::cout << "=====\n\n";
    std::cout << "Bk:\n";
    for (int i = 0; i < n; i++)
    {
        std::cout << "b[" << i + 1 << "] = " << B[i][0] << "\n";
    }
    std::cout << "\n\n";
}

void fortranStyle(int n, double **A, double **B, double *newA, double *newB)
{
    int k = 0;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++, k++)
        {
            newA[k] = A[i][j];
        }
    }

    for (int i = 0; i < n; i++)
    {
        newB[i] = B[i][0];
    }
}

void solution(double **A, double **B, int n, double xn)
{
    double newA[n * n];
    double newB[n];

    fortranStyle(n, A, B, newA, newB);

    double cond1 = 0;
    int *ipvt = new int[n];

    for (int i = 0; i < n; i++)
    {
        ipvt[i] = 0;
    }
    Decomp(n, newA, &cond1, ipvt);
    Solve(n, newA, newB, ipvt);

    out << "Solutions (xn = " << xn << "):\n";

    for (int i = 0; i < n; i++)
    {
        out << "x[" << i << "] = " << newB[i] << "\n";
    }
    out << "\n";
    return;
}
```

```

double** build(double **A, int n, double xn)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            A[i][j] = j + 1;

            if (j == n - 1)
            {
                A[i][j] = xn;
            }

            double power = A[i][j];

            if (i == 0)
            {
                A[i][j] = 1;
            }
            else if (i == 1)
            {
            }
            else
            {
                for (int involution = 1; involution < i; involution++)
                {
                    A[i][j] = A[i][j] * power;
                }
            }

            if (n == 5)
            {
                std::cout << std::setprecision(5) << std::setw(7) << A[i][j] << " | ";
            }
            else if (n == 7)
            {
                std::cout << std::setprecision(7) << std::setw(9) << A[i][j] << " | ";
            }
            else if (n == 9)
            {
                std::cout << std::setprecision(9) << std::setw(11) << A[i][j] << " | ";
            }
        }
        std::cout << "\n";
    }
    return A;
}

int main()
{
    int N[3] = {5, 7, 9};

    double Xn[4] = {1.1, 1.01, 1.001, 1.0001};

    double **A = new double*[5];
    double **B = new double*[7];
    double **C = new double*[9];

    double **S1 = new double*[5];
    double **S2 = new double*[7];
    double **S3 = new double*[9];

    for (int i = 0; i < 3; i++)
    {
        out << "=====\n";
        out << "N = " << N[i] << "\n";
        out << "=====\n\n";

        switch (N[i])
        {
            case 5:
                for (int i = 0 ; i < 5; i++)
                {
                    A[i] = new double [5];
                }
                for (int i = 0 ; i < 5; i++)
                {
                    S1[i] = new double[1];
                }
            }
        }
    }
}

```



```

        printBk(S1, N[i]);

        for (int j = 0; j < 4; j++)
        {
            std::cout << "Matrix A (xn = " << Xn[j] << "):\n";
            A = build(A, N[i], Xn[j]);
            solution(A, S1, N[i], Xn[j]);
            std::cout << "\n";
        }

        break;

    case 7:

        for (int i = 0 ; i < 7; i++)
        {
            B[i] = new double [7];
        }

        for (int i = 0 ; i < 7; i++)
        {
            S2[i] = new double[1];
        }

        printBk(S2, N[i]);

        for (int j = 0; j < 4; j++)
        {
            std::cout << "Matrix B (xn = " << Xn[j] << "):\n";
            B = build(B, N[i], Xn[j]);
            solution(B, S2, N[i], Xn[j]);
            std::cout << "\n";
        }

        break;

    case 9:

        for (int i = 0 ; i < 9; i++)
        {
            C[i] = new double [9];
        }

        for (int i = 0 ; i < 9; i++)
        {
            S3[i] = new double[1];
        }

        printBk(S3, N[i]);

        for (int j = 0; j < 4; j++)
        {
            std::cout << "Matrix C (xn = " << Xn[j] << "):\n";
            C = build(C, N[i], Xn[j]);
            solution(C, S3, N[i], Xn[j]);
            std::cout << "\n";
        }

        break;

    default:
        break;
    }
    std::cout << "\n\n";
}

return 0;
}

```