

## ВЕКТОРЫ И МАТРИЦЫ

### Определение вектора

С упорядоченной последовательностью действительных чисел  $a_1, a_2, a_3, \dots, a_{n-1}, a_n$  можно связать понятие связанного вектора в  $n$ -мерном пространстве и обозначить как:

$$\vec{a} = \bar{a} = [a] = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}$$

или понятие точки  $A(a_1, a_2, a_3, \dots, a_n)$ . Числа  $a_1, a_2, a_3, \dots, a_n$  называются координатами точки  $A$  или элементами вектора  $\vec{a}$ , а количество элементов в векторе называется размерностью этого вектора. Положение элемента  $a_i$  определяется индексом  $i$ , где  $i = 1, 2, \dots, n$ . Компоненты вектора записываются в виде столбца.

### Типы и характеристики векторов

*Нулевой вектор* – вектор, все компоненты которого равны нулю и обозначается как:

$$\vec{0} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

*Единичный вектор* – вектор, длина которого равна единице:

$$\vec{a} = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}; \quad \ell = \sqrt{0.6^2 + 0.8^2} = 1$$

*Транспонированный вектор* – вектор, компоненты которого располагаются в виде строки:

$$\vec{a}^T = [a_1 \ a_2 \ \dots \ a_{n-1} \ a_n] \quad \text{Пример.} \quad \vec{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \vec{a}^T = [2 \ 3]$$

Два вектора с одинаковой размерностью равны тогда, когда равны их соответствующие компоненты.

$$\vec{a} = \vec{b} \quad a_1 = b_1, a_2 = b_2, \dots, a_n = b_n, \quad \text{где } i = 1, 2, 3, \dots, n$$

### Определение матрицы

Совокупность чисел расположенных в прямоугольной таблице, состоящей из  $n$  строк и  $m$  столбцов, называется матрицей и обозначается как:

$$\vec{A} = [A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & \cdot & a_{nm} \end{bmatrix}$$

Положение элемента  $a_{ij}$  в матрице определяется двумя индексами ( $i$  и  $j$ ), где  $i$  определяет номер строки ( $1 \leq i \leq n$ ), а  $j$  – номер столбца ( $1 \leq j \leq m$ ).

### Типы матриц

*Вектор строка* – матрица, состоящая из одной строки  $n=1$

*Вектор столбец* – матрица, состоящая из одного столбца  $m=1$

*Квадратная матрица* – матрица, у которой  $n$  равно  $m$

*Верхне треугольная* – матрица, у которой  $a_{ij}=0$  при  $i > j$

*Нижне треугольная* – матрица, у которой  $a_{ij}=0$  при  $i < j$

*Диагональная* – матрица, у которой  $a_{ij} = 0$  при  $i \neq j$

*Единичная* – матрица, у которой  $a_{ij} = \begin{cases} 1 & \text{при } i = j \\ 0 & \text{при } i \neq j \end{cases}$

*Равенство матриц*  $\vec{A} = \vec{B}$  т.е.  $a_{ij} = b_{ij}$ , где  $i = 1, 2, 3, \dots, n$   $j = 1, 2, 3, \dots, m$

### Характеристики и операции

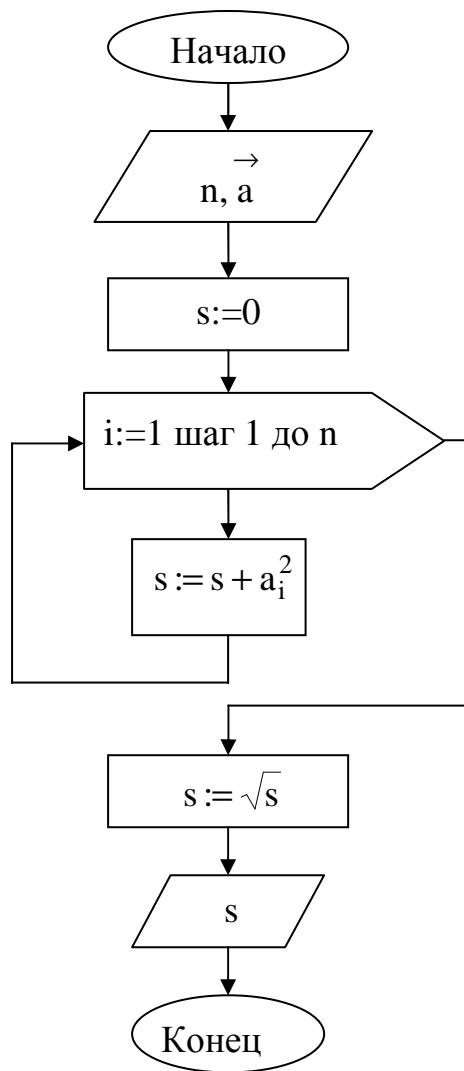
*Норма (длина) вектора*

$$\|\vec{a}\| = \sqrt{\sum_{i=1}^n a_i^2}$$

**Пример.**  $\vec{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$   $\|\vec{a}\| = \sqrt{2^2 + 3^2} = 3.61$

*Алгоритм и программа для вычисления нормы вектора*

Блок-схема



Исполняемая процедура

```

Option base 1
Sub NormaVectora()
Dim n As Integer, a() As Single
Dim s As Single, i As Integer
n = Cells(2, 1)
Redim a(n)
For i= 1 to n
a(i) = Cells(i + 3, 1)
Next i
s = 0
For i = 1 to n
s = s + a(i) * a(i)
Next i
s = Sqr(s)
Cells(2, 3) = s
End Sub
  
```

Вызываемая функция

```

Function NVec(Nrow%, a!()) As Single
Dim s As Single
Dim i As Integer
s = 0
For i = 1 to Nrow
s = s + a(i) * a(i)
Next i
NVec = sqr(s)
  
```

Исполняемая процедура

```

Option base 1
Sub NormaVectora()
Dim n As Integer, a() As Single
Dim s As Single, i As Integer
n = Cells(2, 1)
Redim a(n)
For i= 1 to n
a(i) = Cells(i + 3, 1)
  
```

End Function

Next i

Cells(2, 3) = NVec(n,a)

End Sub

*Норма матрицы (Эвклидова).*

$$\|A\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{i,j}^2}$$

*Программа для вычисления нормы матрицы*

### Исполняемая процедура

```
Option base 1
Sub NormaMatrix()
  Dim n%, m%, a!()
  Dim i%, j%, s !
  n=Cells(2, 1): m= Cells(2, 2)
  Redim a(n,m)
  For i = 1 to n
    For j = 1 to m
      a(i, j) = Cells(3 + i, j)
    Next j
  Next i
  s = 0
  For i = 1 to n
    For j = 1 to m
      s = s + a(i,j)^2
    Next j
  Next i
  Cells(2, 3) = Sqr(s)
End Sub
```

### Вызываемая функция

```
Function NormM(Nrow%, Ncol%, A!())
  Dim i%, j%, s!
  s = 0
  For i = 1 to Nrow
    For j = 1 to Ncol
      s = s + A(i, j)^2
    Next j
  Next i
  NormM = sqr(s)
End Function
```

### Исполняемая процедура

```
Option base 1
Sub NormaMatrix()
  Dim n%, m%, a!()
  Dim i%, j%, s !
  n=Cells(2, 1): m= Cells(2, 2)
  Redim a(n,m)
  For i = 1 to n
    For j = 1 to m
      a(i, j) = Cells(3 + i, j)
    Next j
  Next i
  Cells(2, 3) = NormM(n,m,a)
End Sub
```

**Сложение и вычитание векторов.** Складывать или вычитать можно только вектора с одинаковой размерностью. Результатом операции сложения (вычитания)

является вектор, компоненты которого равны сумме (разности) соответствующих компонент двух других векторов.

$$\vec{c} = \vec{a} \pm \vec{b} \quad c_i = a_i \pm b_i, i = 1, 2, 3, \dots, n \quad \text{Пример.} \quad \vec{a} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \quad \vec{b} = \begin{bmatrix} 2 \\ 7 \end{bmatrix} \quad \vec{c} = \begin{bmatrix} 5 \\ 12 \end{bmatrix}$$

### свойство операции сложения (вычитания):

КОММУТАТИВНОСТЬ

$$\vec{a} + \vec{b} = \vec{b} + \vec{a}$$

ассоциативность

$$\vec{a} + (\vec{b} + \vec{c}) = (\vec{a} + \vec{b}) + \vec{c}$$

### Программа для вычисления суммы векторов

Исполняемая процедура (main)

Вызываемая процедура

Option base 1

Sub AddV(Nrow%, a!()), b!(), r())

Sub AddVector()

Dim i As Integer

Dim n%, a!(), b!(), c!()

For i = 1 to Nrow

Dim i As Integer

r(i) = a(i) + b(i)

n=Cells(2,1)

next i

Redim a(n), b(n), c(n)

End Sub

For i = 1 to n

a(i)=Cells(3 + i, 1)

b(i)=Cells(3 + i, 2)

Вызов процедуры

Next i

For i= 1 to n

в исполняемой процедуре

c(i) = a(i) + b(i)

NormaMatrix заменяем подчеркнутые строки на строку:

Next I

For i= 1 to n

Cells(3 + i, 3)=c(i)

Call AddV(n,a,b,c)

Next I

End Sub

**Сложение и вычитание матриц.** Складывать или вычитать можно только матрицы с одинаковой размерностью. Результатом операции сложения (вычитания) является матрица, элементы которой равны сумме (разности) соответствующих элементов двух других матриц.

$$\vec{C} = \vec{A} \pm \vec{B}; \quad c_{ij} = a_{ij} \pm b_{ij}, \quad i = 1, 2, 3, \dots, n; \quad j = 1, 2, 3, \dots, m$$

### свойство операции сложения (вычитания):

КОММУТАТИВНОСТЬ

$$\overset{=}{A} + \overset{=}{B} = \overset{=}{B} + \overset{=}{A};$$

ассоциативность

$$\overset{=}{A} + (\overset{=}{B} + \overset{=}{C}) = (\overset{=}{A} + \overset{=}{B}) + \overset{=}{C}$$

*Программа для вычисления суммы матриц*

Вызываемая процедура

```
Sub AddM(Nrow As Integer, Ncol As Integer, _
        A() As Single, B() As Single, C() As Single)
Dim i As Integer
Dim j As Integer
For i = 1 to Nrow
  For j = 1 to Ncol
    C(i, j) = A(i, j) + B(i, j)
  Next j
Next i
End Sub
```

**Умножение вектора на константу.** Результатом умножения является вектор, компоненты которого равны произведению соответствующих компонент исходного вектора на константу.

$$\vec{c} = \lambda \cdot \vec{a} \quad c_i = \lambda \cdot a_i, \quad i = 1, 2, 3, \dots, n$$

**Пример.**  $\lambda = 2.3 \quad \vec{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \vec{c} = 2.3 \cdot \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4.6 \\ 6.9 \end{bmatrix}$

**Свойства операции:**

$$\lambda * (\vec{a} + \vec{b}) = \lambda * \vec{a} + \lambda * \vec{b}$$

$$(\lambda + \beta) * \vec{a} = \lambda * \vec{a} + \beta * \vec{a}$$

$$\lambda * (\beta * \vec{a}) = (\lambda * \beta) * \vec{a}$$

$$0 * \vec{a} = \vec{0}$$

$$1 * \vec{a} = \vec{a}$$

$$(-1) * \vec{a} = -\vec{a}$$

**Умножение матрицы на константу.** Результатом умножения является матрица, элементы которой равны произведению соответствующих элементов исходной матрицы на константу.

$$\bar{C} = \lambda \cdot \bar{A}; \quad c_{i,j} = \lambda \cdot a_{i,j}, \quad i = 1, 2, 3, \dots, n; \quad j = 1, 2, 3, \dots, m$$

**Свойства операции:**

$$\lambda * (\bar{A} + \bar{B}) = \lambda * \bar{A} + \lambda * \bar{B}$$

$$(\lambda + \beta) * \bar{A} = \lambda * \bar{A} + \beta * \bar{A}$$

$$\lambda * (\beta * \bar{A}) = (\lambda * \beta) * \bar{A}$$

$$0 * \bar{A} = \vec{0}$$

$$1 * \bar{A} = \bar{A}$$

$$(-1) * \bar{A} = -\bar{A}$$

**Транспонирование вектора** – это изменение представления вектора. Представление в виде столбца на строку.

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix} \quad \vec{a}^T = [a_1 \ a_2 \ \dots \ a_{n-1} \ a_n] \quad \text{Пример.} \quad \vec{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \vec{a}^T = [2 \ 3]$$

**Транспонирование матрицы** – это замена строк матрицы соответствующими столбцами т.е.  $a'_{j,i} = a_{i,j}$ , где  $i = 1, 2, 3, \dots, n; j = 1, 2, 3, \dots, m$

$$\bar{A}^T = [\bar{A}]^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & \cdot & a_{nm} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdot & \cdot & a_{n1} \\ a_{12} & a_{22} & a_{32} & \cdot & \cdot & a_{n2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1m} & a_{2m} & a_{3m} & \cdot & \cdot & a_{nm} \end{bmatrix}$$

*Программа для вычисления транспонированной матрицы*  
Вызываемая процедура

```
Sub TranM(Nrow As Integer, Ncol As Integer, _
          A() As Single, At() As Single)
```

```
Dim I As Integer, j As Integer
```

```
For i = 1 to Nrow
```

```
  For j = 1 to Ncol
```

```
    At(j,i)=A(i,j)
```

```
  Next j
```

```
Next i
```

```
End Sub
```

**Скалярное произведение векторов** – это значение суммы произведений соответствующих компонент двух векторов.

$$z = (\vec{a}, \vec{b}) = \vec{a}^T \cdot \vec{b} = \sum_{i=1}^n a_i \cdot b_i, \quad i = 1, 2, 3, \dots, n$$

**Пример.**  $\vec{a} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$   $\vec{b} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$   $z = [2 \ 3] \cdot \begin{bmatrix} 4 \\ 5 \end{bmatrix} = 2 \cdot 4 + 3 \cdot 5 = 23$

Вызываемая функция

```
Function Scal(Nrow As Integer, a() As Single, b() As Single)
```

```
Dim i As Integer, s As Single
```

```
s = 0
```

```
For i = 1 to Nrow
```

```
s = s + a(i) * b(i)
```

```
Next i
```

```
Scal = s
```

```
End Function
```

**Угол между векторами.** Косинус угла  $\cos(\theta) = \frac{\vec{a}^T \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$

Вызываемая функция ( обращается к функциям Scal и NVec )

```
Function Cost(n As Integer, a() As Single, b() As Single)
```

```
Cost = Scal(n,a,b)/(NVec(n,a) * NVec(n,b))
```



**Ортогональность векторов**  $\cos(\theta) = 0$  т.е.  $\vec{a}^T * \vec{b} = 0$

Могут возникнуть две задачи:

Проверка ортогональности двух векторов даны вектора

$$\vec{a} = \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix} \quad \vec{b} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \quad z = [-1 \ 3 \ -2] * \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} = -1*1 + 3*3 - 2*4 = 0$$

Эти вектора ортогональны.

Найти вектор ортогональный данному.

$$\vec{a} = \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix} \quad [-1 \ 3 \ -2] * \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = (-1)*b_1 + 3*b_2 + (-2)*b_3 = 0$$

задаем  $b_1 = 3 \quad b_2 = 5$

$$\text{получаем } (-3) + 15 + (-2)*b_3 = 0 \quad 12 - 2*b_3 = 0 \quad b_3 = 6 \quad \vec{b} = \begin{bmatrix} 3 \\ 5 \\ 6 \end{bmatrix}$$

**Линейная зависимость векторов**

Вектора  $\vec{a}^{(i)}$  называются линейно зависимыми, если соотношение  $\sum_{i=1}^m \beta_i * \vec{a}^{(i)} = \vec{0}$  справедливо, хотя бы при одном множителе  $\beta_i$  отличным от нуля.

Пример:

$$\vec{a}^{(1)} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}; \quad \vec{a}^{(2)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \beta_1 * \vec{a}^{(1)} + \beta_2 * \vec{a}^{(2)} = \vec{0} \quad \beta_1 * \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \beta_2 * \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\beta_1 * 2 + \beta_2 * 1 = 0 \quad \beta_2 = -2 * \beta_1$$

$$\beta_1 * 4 + \beta_2 * 2 = 0 \quad \beta_1 * 4 - \beta_1 * 4 = 0 \quad \beta_1 * 0 = 0 \text{ при любом } \beta_1$$

$$\vec{a}^{(1)} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}; \quad \vec{a}^{(2)} = \begin{bmatrix} +2 \\ -1 \end{bmatrix} \quad \beta_1 * \vec{a}^{(1)} + \beta_2 * \vec{a}^{(2)} = \vec{0} \quad \beta_1 * \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \beta_2 * \begin{bmatrix} +2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\beta_1 * 1 + \beta_2 * 2 = 0 \quad \beta_1 = -2 * \beta_2$$

$$\beta_1 * 2 - \beta_2 * 1 = 0 \quad -\beta_2 * 4 - \beta_2 * 1 = 0 \quad \beta_2 * (-5) = 0 \text{ только при } \beta_2 = 0$$

Если  $m > n$ , то вектора линейно зависимы, где  $m$  - количество векторов, а  $n$  – размерностью пространства.

### Умножение матриц.

$$\overset{=}{C} = \overset{=}{A} \cdot \overset{=}{B} \neq \overset{=}{B} \cdot \overset{=}{A} \quad \overset{=}{C}_{nm} = \overset{=}{A}_{nk} \cdot \overset{=}{B}_{km}$$

Количество столбцов матрицы  $\overset{=}{A}$  должно равняться количеству строк матрицы  $\overset{=}{B}$ . Элемент  $c_{ij}$  вычисляется как скалярное произведение  $i$ -й строки матрицы  $\overset{=}{A}$  и  $j$ -го столбца матрицы  $\overset{=}{B}$ :

$$c_{ij} = \sum_{l=1}^k a_{il} \cdot b_{lj}$$

Вызываемая процедура

Sub MultM((NrowA as Integer, NcolA As Integer, NcolB As Integer, \_  
A() As Single, B() As Single, C() As Single)

Dim i As Integer, j As Integer, l as Single

Dim s As Single

For i = 1 to NrowA

For j = 1 to NcolA

s = 0

For l = 1 to NcolB: s = s + A(i, l) \* B(l, j): Next l

C(i, j)=s

Next j

Next i

End Sub

**Пример.** Вычислить матрицу  $\overset{=}{C} = \overset{=}{A} \cdot \overset{=}{B}$ .

$$\overset{=}{A} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad \overset{=}{B} = \begin{bmatrix} 1 & 0 & 1 & 2 \\ 3 & 1 & 0 & 1 \\ 1 & 2 & 1 & 2 \end{bmatrix}$$

$$\overset{=}{C} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 & 2 \\ 3 & 1 & 0 & 1 \\ 1 & 2 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 2 & 5 \\ 8 & 4 & 2 & 6 \end{bmatrix}$$

**Пример.** Вычислить значение матричного выражения R. Сохраните промежуточные вычисления. Составьте программу для вычисления R, используя язык программирования VBA. Проведите расчёт по программе и сравните результаты. Операция  $\text{sled}([\text{матрица}])$  – произведение диагональных элементов матрицы.

$$R = \left\| \overset{=}{A} \cdot \overset{=}{B} \cdot c + \overset{=}{A} \cdot \overset{=}{d} \right\|$$

$$\overset{=}{A} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \end{bmatrix}; \quad \overset{=}{B} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}; \quad \overset{\rightarrow}{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}; \quad \overset{\rightarrow}{d} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\overset{=}{RM} = \overset{=}{A} \cdot \overset{=}{B} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 6 \\ 18 & 7 \end{bmatrix}$$

$$\overset{=}{RV} = \overset{=}{RM} \cdot \overset{\rightarrow}{c} = \begin{bmatrix} 9 & 6 \\ 18 & 7 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 36 \\ 57 \end{bmatrix} \quad \overset{=}{RV1} = \overset{=}{A} \cdot \overset{\rightarrow}{d} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 9 \\ 10 \end{bmatrix}$$

$$\overset{=}{RV1} = \overset{=}{RV1} + \overset{=}{RV} = \begin{bmatrix} 9 \\ 10 \end{bmatrix} + \begin{bmatrix} 36 \\ 57 \end{bmatrix} = \begin{bmatrix} 45 \\ 67 \end{bmatrix} \quad R = \sqrt{\sum_{i=1}^2 rv1_i} = \sqrt{45^2 + 67^2} = 80.709$$

*Программа вычисления значения матричного выражения*

**Исполняемая процедура с использованием вызываемых процедур и функций**

Option Explicit

Option Base 1

Sub main()

Dim A!(), B!(), C!(), D!(), RM!(), RV!(), RV1!()

Dim n%, m%

n=cells(2,1): m=cells(2,3)

Redim A(n, m), B(m, n), C(n), D(m), RM(n, n), RV(n), RV1(n)

Call InputMatrix("лист1", 5, 1, n, m, A)

Call InputMatrix("лист1", 5, 5, m, n, B)

Call InputVector("лист1", 5, 8, n, C)

Call InputVector("лист1", 5, 10, m, D)

Call MultMatr(n, m, n, A, B, RM)

Call MultMatrVect(n, n, RM, C, RV)

Call MultMatrVect(n, m, A, D, RV1)

Call AddVect(n, RV, RV1, RV1)

Cells(2, 5) = NormVect(n, RV1)

End Sub

'Умножение матриц

Public Sub MultMatr(LeftRow%, Col\_Row%, RightCol%, \_

LeftMatr() As Single, RightMatr() As Single, ResMatr() As Singl)

Dim s As Single, i As Integer, j As Integer, l As Integer

For i = 1 To LeftRow Step 1

For j = 1 To RightCol Step 1

s = 0

```

For l = 1 To Col_Row Step 1
    s = s + LeftMatr(i, l) * RightMatr(l, j)
Next l
ResMatr(i, j) = s
Next j
Next i
End Sub

```

'Сложение векторов

```

Public Sub AddVect(Row As Integer, LeftVect() As Single, _
    RightVect() As Single, ResVect() As Single)
Dim i As Integer
For i = 1 To Row Step 1
    ResVect(i) = LeftVect(i) + RightVect(i)
Next i
End Sub

```

'Умножение матрицы на вектор'

```

Public Sub MultMatrVect(Row As Integer, ColRow As Integer, _
    Matr() As Single, Vect() As Single, ResVect() As Single)
Dim i As Integer, j As Integer, s As Single
For i = 1 To Row Step 1
    s = 0
    For j = 1 To ColRow Step 1
        s = s + Matr(i, j) * Vect(j)
    Next j
    ResVect(i) = s
Next i
End Sub

```

'Норма вектора

```

Public Function NormaVect(Row As Integer, Vect() As Single) As Single
Dim norma As Single, i As Integer
norma = 0
For i = 1 To Row Step 1
    norma = norma + Vect(i) * Vect(i)
Next i
NormaVect = Sqr(norma)
End Function

```

'Ввод матрицы (sh – Лист)

```

Public Sub InputMatrix(sh As String, UpRow As Integer, _
    leftCol As Integer, Row As Integer, _
    Col As Integer, Matr() As Single)
For i = 1 To Row Step 1

```

```

For j = 1 To Col Step 1
  Matr(i, j) = Worksheets(sh).Cells(UpRow - 1 + i, leftCol - 1 + j)
Next j
Next i
End Sub

```

```

'Ввод вектора
Public Sub InputVector(sh As String, UpRow As Integer, _
  leftCol As Integer, Row As Integer, _
  Vect() As Single)
For i = 1 To Row Step 1
  Vect(i) = Worksheets(sh).Cells(UpRow - 1 + i, leftCol)
Next i
End Sub

```

### Обращение матрицы методом Гаусса-Жордана

**Обратной** матрицей называется такая квадратная матрица  $A_{nn}^{-1}$ , при умножении которой на исходную как справа, так и слева получается единичная матрица

$$A^{-1} \cdot A = A \cdot A^{-1} = E$$

Обращение матрицы  $A$  методом Гаусса-Жордана заключается в построении расширенной матрицы

$$\left[ A \mid E \right]$$

и преобразовании расширенной матрицы так, чтобы на месте исходной получилась единичная матрица, тогда на месте единичной получится обратная матрица:

$$\left[ E_{nn} \mid A^{-1} \right]$$

Текстуальный алгоритм метода Гаусса-Жордана состоит из четырёх этапов.

1. Строим расширенную матрицу дописав к исходной квадратной матрице единичную матрицу того же размера

$$\bar{C}_{n,2n} = \left[ \bar{A}_{nn} \mid \bar{E}_{nn} \right], \text{ и задаём номер ведущей строки } k=1.$$

2. Все элементы  $k$ -й строки делим на элемент  $c_{kk}$

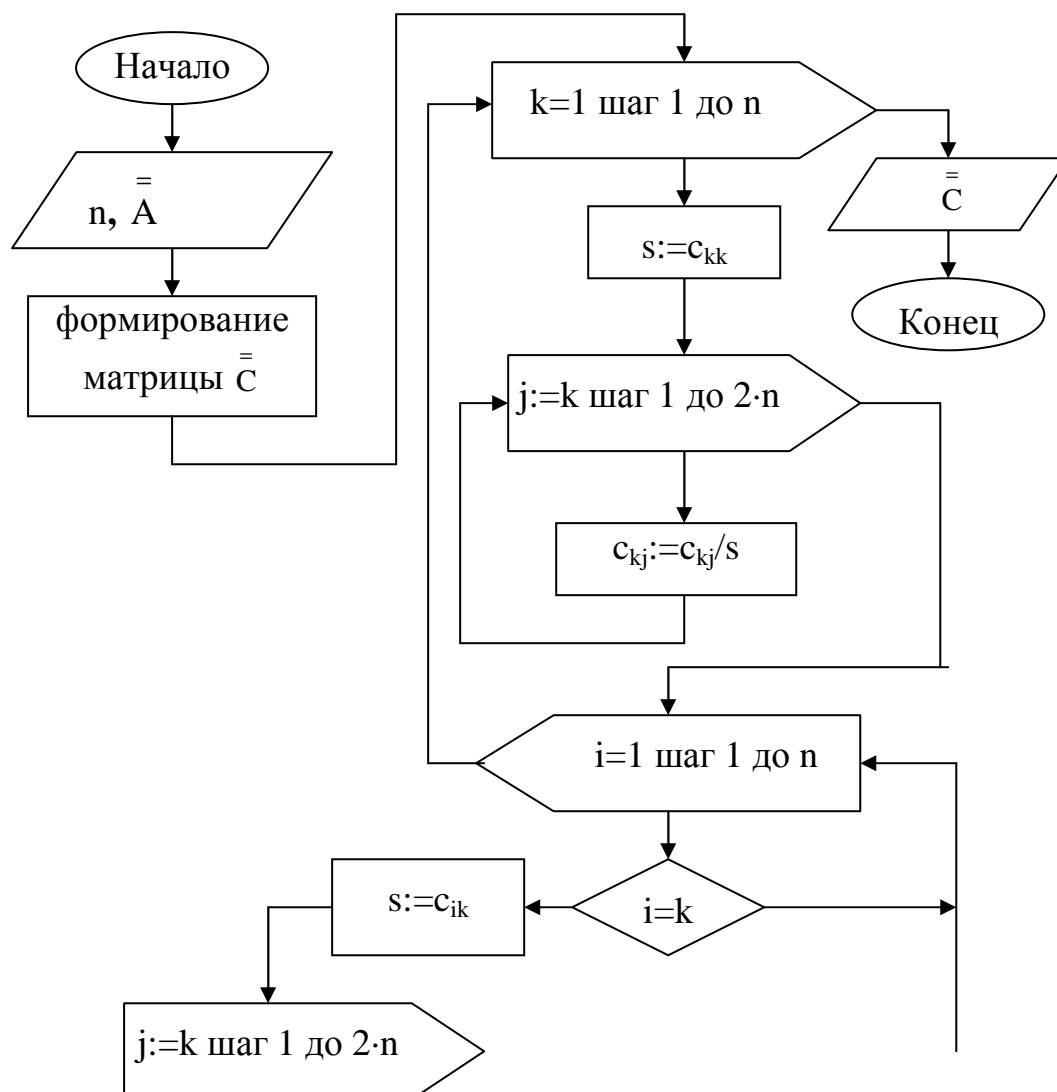
$$c_{kj} = \frac{c_{kj}}{c_{kk}}, \quad j = k, k+1, k+2, \dots, 2 \cdot n \quad \text{т.е. } c_{kk} = 1.$$

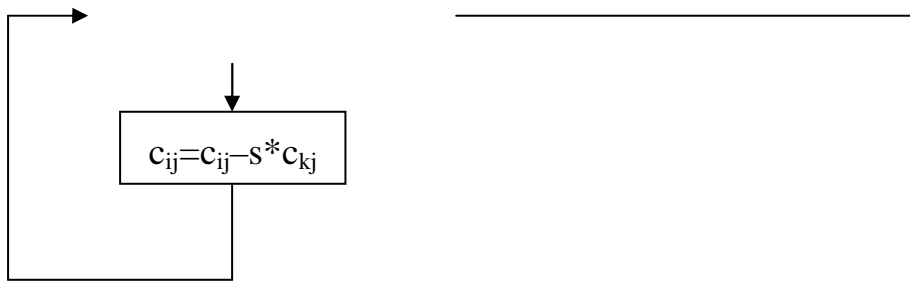
3. Преобразуем все  $i$ -е строки кроме  $k$ -й,  $i=1, 2, 3, \dots, n \quad i \neq k$  так, чтобы элементы  $c_{ik}=0$ . Для этого из каждого элемента  $i$ -й строки вычитаем соответствующий элемент  $k$ -й строки, умноженный на элемент  $c_{ik}$ , т.е.

$$c_{ij} = c_{ij} - c_{kj} \cdot c_{ik}, \quad j = k, k+1, k+2, \dots, 2 \cdot n$$

4. Проверяем условие  $k < n$ , если оно справедливо, то  $k=k+1$  и выполняем алгоритм с пункта 2, иначе выводим полученную обратную матрицу, расположенную на месте единичной.

Блок-схема метода Гаусса-Жордана





**Пример.** Найти обратную матрицу.

$$\bar{A} = \begin{bmatrix} 2.00 & 1.00 & 2.00 \\ 2.00 & 3.00 & 1.00 \\ 1.00 & 2.00 & 3.00 \end{bmatrix} \quad \bar{C} = \left[ \begin{array}{ccc|ccc} 2.00 & 1.00 & 2.00 & 1.00 & 0.00 & 0.00 \\ 2.00 & 3.00 & 1.00 & 0.00 & 1.00 & 0.00 \\ 1.00 & 2.00 & 3.00 & 0.00 & 0.00 & 1.00 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.50 & 1.00 & 0.50 & 0.00 & 0.00 \\ 2.00 & 3.00 & 1.00 & 0.00 & 1.00 & 0.00 \\ 1.00 & 2.00 & 3.00 & 0.00 & 0.00 & 1.00 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.50 & 1.00 & 0.50 & 0.00 & 0.00 \\ 0.00 & 2.00 & -1.00 & -1.00 & 1.00 & 0.00 \\ 0.00 & 1.50 & 2.00 & -0.50 & 0.00 & 1.00 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.50 & 1.00 & 0.50 & 0.00 & 0.00 \\ 0.00 & 1.00 & -0.50 & -0.50 & 0.50 & 0.00 \\ 0.00 & 1.50 & 2.00 & 0.50 & 0.00 & 1.00 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.00 & 1.25 & 0.75 & -0.25 & 0.00 \\ 0.00 & 1.00 & -0.50 & -0.50 & 0.50 & 0.00 \\ 0.00 & 0.00 & 2.75 & 0.25 & -0.75 & 1.00 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.00 & 1.25 & 0.75 & -0.25 & 0.00 \\ 0.00 & 1.00 & -0.50 & -0.50 & 0.50 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.09 & -0.27 & 0.36 \end{array} \right]$$

$$\left[ \begin{array}{ccc|ccc} 1.00 & 0.00 & 0.00 & 0.64 & 0.09 & -0.45 \\ 0.00 & 1.00 & 0.00 & -0.45 & 0.36 & 0.18 \\ 0.00 & 0.00 & 1.00 & 0.09 & -0.27 & 0.36 \end{array} \right]$$

$$\text{Результат } A^{-1} = \left[ \begin{array}{ccc} 0.64 & 0.09 & -0.45 \\ -0.45 & 0.36 & 0.18 \\ 0.09 & -0.27 & 0.36 \end{array} \right]$$