

## МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ОБЪЕКТОВ ЛИНЕЙНОЙ АЛГЕБРЫ И АНАЛИТИЧЕСКОЙ ГЕОМЕТРИИ В СИСТЕМЕ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MAPLE

© А.Р.Самигуллина

В статье описаны основные принципы математического моделирования и программные процедуры в системе компьютерной математики Maple графического представления основных объектов линейной алгебры и аналитической геометрии.

**Ключевые слова:** математическое моделирование, системы компьютерной математики, алгоритмы, линейная алгебра и аналитическая геометрия.

### 1. Введение

Как известно, линейная алгебра и аналитическая геометрия традиционно тесно связаны между собой, что вызвано как историческими корнями этих разделов математики, так и определяющей зависимостью структур аналитической геометрии структурами и методами линейной алгебры. Особенно тесно эта связь проявляется в области векторной алгебры. В свою очередь все наглядные интерпретации алгебраической структуры векторных пространств фактически являются геометрическими. Проблема обеспечения наглядности этих структур играет важную роль в высшем образовании, т.к. усвоение фундаментальных геометрических понятий подготавливает основу для понимания процесса математического моделирования и овладения методами компьютерного моделирования, что в свою очередь, создает предпосылки для инновационного развития современного образования.

Решение проблемы компьютерной реализации объектов линейной алгебры и аналитической геометрии и создания наглядных геометрических образов (интерпретаций) объектов, структур и свойств предполагает решение трех основных задач:

1. построение математических моделей основных алгебраических структур, объектов и свойств;
2. построение их геометрических интерпретаций, т.е., сопоставление их геометрических моделей;
3. построение многопараметрических компьютерных моделей графических образов объектов.

Заметим, что многопараметричность создаваемых компьютерных моделей является важнейшим фактором компьютерных моделей, позволяющим управлять математической моделью, т.е. проводить компьютерное моделирование. Наиболее эффективное решение этих задач воз-

можно в системах компьютерной математики (СКМ), среди которых для целей образования наиболее удобна система Maple. Основными достоинствами этой системы применительно к задачам образования являются относительно невысокая стоимость (по сравнению с MatLab и Mathematica), дружелюбный и интерактивный интерфейс, великолепные графические возможности, в частности интерактивная трехмерная графика и динамическая (анимация). В этой статье мы рассмотрим основные принципы математического и компьютерного моделирования объектов линейной алгебры и аналитической геометрии в СКМ Maple. Заметим, что для рассмотренных здесь программных процедур конкретная версия Maple, начиная с версии 6, не имеет значения.

### 2. Математическое моделирование решения системы линейных алгебраических уравнений

Системы линейных алгебраических уравнений являются одним из основных объектов линейной алгебры, а теория систем линейных алгебраических уравнений широко применяется во всех разделах аналитической геометрии, т.к. эти системы описывают весьма важные для геометрии линейные образы (плоскости и прямые).

Рассмотрим решение системы линейных алгебраических уравнений. Пусть дана система из  $m$  линейных уравнений относительно  $n$  неизвестных:

$$\mathbf{A}\mathbf{X} = \mathbf{B} \Rightarrow \sum_{i=1}^n a_{ki}x_i = b_k; (k = \overline{1, m}), \quad (1)$$

где  $\mathbf{A}$  – основная матрица системы,  $\mathbf{B}$  – матрица-столбец свободных членов,  $\mathbf{X}$  – матрица-столбец неизвестных:

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}; \quad \mathbf{B} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}; \quad \mathbf{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}.$$

В пакете LinearAlgebra существует программная процедура решения системы (1), однако она неудобна тем, что представляет решение в малопонятной Российскому студенту форме. Поэтому создадим собственную программную процедуру SolLin в библиотеке Algebra, позволяющую находить решение системы (1) в удобном виде:

```
>Algebra[SolLin]:=proc(Eqs) local
n,m,A,B1,B,F,k1,k2:
n:=LinearAlgebra[ColumnDimension](convert(Eqs,
Matrix)):
A:=convert([seq([coeffs(lhs(Eqs[i]))],i=1..n)],Matrix):
m:=LinearAlgebra[RowDimension](A):
B1:=seq(rhs(Eqs[i]),i=1..n):
B:=Vector([B1]):
F:=convert([seq([coeffs(lhs(Eqs[i])),rhs(Eqs[i])],i=
1..n)],Matrix):k1:=linalg[rank](A):k2:=linalg[rank]
(F):
if k1=k2 then
LinearAlgebra[LinearSolve](A,B,free='C'):
else
("Система не совместна"): end if:
end proc:
```

и сохраним созданную процедуру в библиотеке Algebra, файле Alg.m :

```
>save(Algebra,`Alg.m`):
```

Для исполнения этой процедуры необходимо вызвать созданную библиотеку из любой рабочей тетради Maple, имеющей собственное имя:

```
>read "\Path\Alg.m":
```

```
with(Algebra):
```

Покажем пример решения системы уравнений с помощью созданной процедуры:

```
>SolLin([x-y+z-u=3,x+y=2,2*x-2*y+2*z-
2*u=6,x-y-z-u=1]);
```

Исполнение процедуры имеет вид:

$$\begin{bmatrix} 2 - C_2 \\ C_2 \\ 1 \\ -2C_2 \end{bmatrix},$$

где  $C_2$  – постоянная. Заметим, что созданная процедура представляет решение системы уравнений в стандартной для российских вузов форме, при этом индекс произвольных постоянных соответствует их положениям в матрице общего решения.

### 3. Математическое моделирование трехмерных линейных объектов

#### 3.1. Алгебраический этап моделирования

Системы линейных алгебраических уравнений имеет адекватную геометрическую интерпретацию (см., например, [1]): каждое линейное

алгебраическое уравнение в  $n$ -мерном пространстве определяет гиперплоскость, а система линейных алгебраических уравнений определяет взаимное расположение этих гиперплоскостей. Если система уравнений не имеет решения, то гиперплоскости не пересекаются, т.е. параллельны, в случае существования решения гиперплоскости пересекаются по  $k$ -мерным плоскостям, включая 1-мерные (прямые) и 0-мерные (точки). В случае трехмерных пространств, которые важны для общего среднего и нематематического высшего образования, число различных типов взаимного расположения плоскостей или прямых значительно сокращается.

Как известно (см., например, [1]), прямая в евклидовом пространстве  $E_3$  определяется опорной точкой  $M_0$  и ненулевым направляющим вектором  $\mathbf{q} \neq \mathbf{0}$ , как геометрическое место точек  $E_3$ :

$$d(M_0, \mathbf{q}) = \left\{ M \in E_3 \mid \overline{M_0M} = \lambda \mathbf{q} \right\},$$

где параметр  $\lambda$  принимает значения на всем множестве действительных чисел  $\mathbb{R}$ . Плоскость в евклидовом пространстве также определяется опорной точкой  $M_0$  и ненулевым нормальным вектором  $\mathbf{N} \neq \mathbf{0}$ , как геометрическое место точек  $E_3$ :

$$\Pi(M_0, \mathbf{N}) = \left\{ M \in E_3 \mid \left( \mathbf{N}, \overline{M_0M} \right) = 0 \right\}.$$

Соответственно этим определениям, при заданном декартовом репере  $R\{O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ , где  $\mathbf{e}_i$  векторы ортонормированного базиса:

$$(\mathbf{e}_i, \mathbf{e}_k) = \delta_{ik}; \quad (i, k = \overline{1, 3}),$$

прямая  $d(M_0, \mathbf{q})$  описывается параметрическими (или каноническими) уравнениями:

$$d(M_0, \mathbf{q}): \quad x^i = x_0^i + \lambda q^i; \quad (i = \overline{1, 3}), \quad (2)$$

где  $x_0^i, q^i, x^i$  – координаты опорной точки  $M_0$ , направляющего вектора  $\mathbf{q}$ , и текущей точки прямой  $M$ , соответственно. Плоскость же в евклидовом пространстве  $E_3$  описывается общим уравнением:

$$\begin{aligned} \Pi(M_0, \mathbf{N}): \quad (\mathbf{r} - \mathbf{r}_0, \mathbf{N}) = 0 &\Rightarrow A_i(x^i - x_0^i) = 0 \Rightarrow \\ &\Rightarrow A(x - x_0) + B(y - y_0) + C(z - z_0) = 0, \end{aligned} \quad (3)$$

где  $A_i$  – координаты нормального вектора  $\mathbf{N} = (A, B, C)$ .

При построении математических моделей линейных объектов в трехмерном евклидовом пространстве принципиально важными являются три теоремы (см. [1] и более подробно [2]):

**Теорема 1.** Две прямые,  $d(M_1, \mathbf{q}_1)$  и  $d(M_2, \mathbf{q}_2)$ , в евклидовом пространстве  $E_3$ :

1. пересекаются в единственной точке при условии:

$$(\mathbf{q}_1, \mathbf{q}_2, \overline{M_1 M_2}) \neq 0; \quad (4)$$

2. параллельны при условиях:

$$(\mathbf{q}_1, \mathbf{q}_2, \overline{M_1 M_2}) = 0, \quad (5)$$

$$[\mathbf{q}_1, \mathbf{q}_2] \equiv \mathbf{q}_1 \times \mathbf{q}_2 \neq \mathbf{0}; \quad (6)$$

3. совпадают при условиях (5) и:

$$[\mathbf{q}_1, \mathbf{q}_2] \equiv \mathbf{q}_1 \times \mathbf{q}_2 = \mathbf{0} \quad (7)$$

4. или скрещиваются при условии:

$$(\mathbf{q}_1, \mathbf{q}_2, \overline{M_1 M_2}) = 0. \quad (8)$$

**Теорема 2.** Две плоскости,  $\Pi_1(M_1, N_1)$  и  $\Pi_2(M_2, N_2)$ , в евклидовом пространстве  $E_3$ :

1. пересекаются по единственной прямой при условии:

$$[\mathbf{q}_1, \mathbf{q}_2] \neq \mathbf{0}; \quad (9)$$

2. параллельны при условиях:

$$[\mathbf{q}_1, \mathbf{q}_2] = \mathbf{0}, \quad (10)$$

$$M_2 \notin \Pi_1(M_1, N_1); \quad (11)$$

3. или совпадают при условиях (10) и:

$$M_2 \in \Pi_1(M_1, N_1). \quad (12)$$

**Теорема 3.** Прямая  $d(M_0, \mathbf{q})$  и плоскость  $\Pi(M_1, N)$  в евклидовом пространстве  $E_3$ :

1. пересекаются в единственной точке при условии:

$$(\mathbf{q}, N) \neq 0; \quad (13)$$

2. параллельны при условиях:

$$(\mathbf{q}, N) = 0, \quad (14)$$

$$M_0 \notin \Pi(M_1, N); \quad (15)$$

3. и совпадают при условиях (14) и:

$$M_0 \in \Pi(M_1, N). \quad (16)$$

Фактически уравнения (2), (3) и перечисленные три теоремы с алгебраической точки зрения полностью описывают задачу о взаимном расположении линейных объектов<sup>1</sup>. Однако, с геометрической точки зрения задача о взаимном расположении линейных объектов еще далеко не завершена.

### 3.2. Геометрический этап моделирования

При получении ответа о реализации одного из перечисленных типов взаимного расположения линейных объектов в теоремах 1-3 (4 типа для прямых, 3 – для плоскостей и 3 – для плоскостей и прямых) необходимо далее решить конкретную геометрическую задачу. Такими задачами являются:

1. нахождение координат точки пересечения прямых;
2. вычисление угла между прямыми;
3. вычисление расстояния между параллельными прямыми;

4. нахождение угла между скрещивающимися прямыми;
5. нахождение прямой пересечения плоскостей;
6. вычисление угла между плоскостями;
7. вычисление расстояния между параллельными плоскостями;
8. нахождение координат точки пересечения прямой и плоскости;
9. вычисление угла между прямой и плоскостью;
10. вычисление расстояния между параллельными прямой и плоскостью.

Решение каждой из перечисленных выше стандартных геометрических задач обеспечивается рядом определений и алгоритмов, которые дополняют математическую модель и позволяют дать ей четкую геометрическую интерпретацию. Приведем пример моделирования взаимного расположения прямой и плоскости в пространстве в случае их взаимного пересечения (Рис.1). На этом рисунке показано определение угла  $\alpha$  между прямой  $d$  и плоскостью  $\Pi$  как угла между прямой  $d$  и ее ортогональной проекцией  $d_1$  на плоскость  $\Pi$ , при этом численное значение угла

определяется выражением:  $\sin \alpha = \frac{(\mathbf{q}, N)}{|\mathbf{q}| |N|}$ .

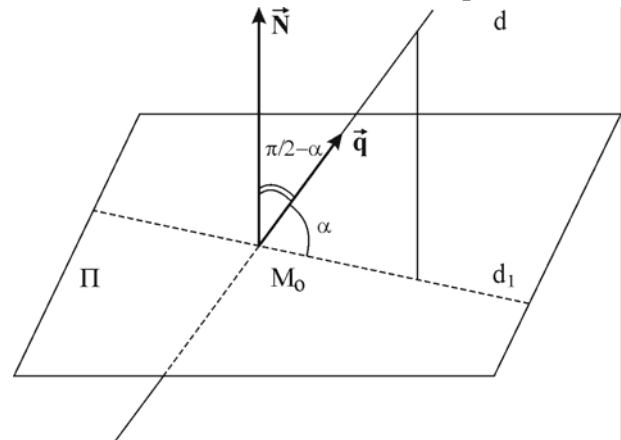


Рис.1. Определение угла  $\alpha$  между прямой  $d$  и плоскостью  $\Pi$

## 4. Компьютерное моделирование трехмерных линейных объектов

После завершения этапа математического моделирования переходим к компьютерному моделированию в пакете Maple.

### 4.1. Компьютерное моделирование взаимного расположения плоскостей

Заметим, что несмотря на внешнюю простоту рассмотренных выше математических моделей их компьютерное моделирование достаточно сложно, поскольку основным требованием к программным продуктам, используемым в учебном процессе, является высокая степень наглядности

<sup>1</sup> Под линейными объектами понимаются фигуры, описываемые системами линейных уравнений.

и простоты их применения. Поэтому ниже мы продемонстрируем лишь некоторые фрагменты программных процедур. Создадим программную процедуру *EqLine3(Eqs,t)* полного исследования и геометрического моделирования взаимного расположения двух плоскостей, заданных общими уравнениями *Eqs* в библиотеке *Anal\_Geom*, позволяющую определить взаимно расположение плоскостей в пространстве<sup>2</sup>:

```
> Anal_Geom[EqLine3]:=proc(Eqs,t) local
n,m,A,B1,B,F,k1,k2,SS,SST,EQS,M0,M1,q,N1,N2,Q1
,Q2,M2,M3,M4,M5,GP1,GP2,GL,z1,z2,P1,P2,W1,W
2:
n:=LinearAlgebra[ColumnDimension](convert(Eqs,
Matrix)):
A:=convert([seq([coeffs(lhs(Eqs[i]))],i=1..n)],Matrix
x):
m:=LinearAlgebra[RowDimension](A):
B1:=seq(rhs(Eqs[i]),i=1..n):
B:=Vector([B1]):
F:=convert([seq([coeffs(lhs(Eqs[i])),rhs(Eqs[i])],i=
1..n)],Matrix):
k1:=linalg[rank](A):
k2:=linalg[rank](F):
if k1=2 and k2=2 then
SS:=LinearAlgebra[LinearSolve](A,B,free='C'):
if SS[1]=C[1] then SST:=subs(C[1]=t,SS):
elif SS[2]=C[2] then SST:=subs(C[2]=t,SS):
elif SS[3]=C[3] then SST:=subs(C[3]=t,SS):
end if:
EQS:=[SST[1],SST[2],SST[3]]:
M0:=convert(subs(t=0,SST),list):
M1:=convert(subs(t=2,SST),list):
N1:=convert(linalg[row](A,1),Vector):
N2:=convert(linalg[row](A,2),Vector):
q:=convert(coeff(SST,t),Vector):
Q1:=convert(linalg[crossprod](q,N1),list):
Q2:=convert(linalg[crossprod](q,N2),list):
M2:=convert(linalg[matadd](M0,Q1,1,1),list):
M3:=convert(linalg[matadd](M1,Q1,1,1),list):
M4:=convert(linalg[matadd](M0,Q2,1,1),list):
M5:=convert(linalg[matadd](M1,Q2,1,1),list):
GL:=plots[spacecurve](EQS,t=0..2,color=red,thick
ness=2):
GP1:=plots[polygonplot3d]([M0,M2,M3,M1]):
GP2:=plots[polygonplot3d]([M0,M4,M5,M1]):
plots[display](GL,GP1,GP2,axes=BOXED,title="Пл
оскости пересекаются по прямой"):
elif k1=1 and k2=1 then
("Плоскости совпадают")
else
z1:=LinearAlgebra[LinearSolve](Matrix([[linalg[ro
w](A,1)],[linalg[row](A,1)]]),Vector([B[1],B[1]]),fr
ee='Z'):

```

```
z2:=LinearAlgebra[LinearSolve](Matrix([[linalg[ro
w](A,2)],[linalg[row](A,2)]]),Vector([B[2],B[2]]),fr
ee='U'):

```

```
P1:=subs({Z[1]=0,Z[2]=0,Z[3]=0},z1):
P2:=subs({U[1]=0,U[2]=0,U[3]=0},z2):
if
z1[1]=Z[1] and z1[2]=Z[2] then
W1:=subs({Z[1]=u,Z[2]=v},z1[3]):
GP1:=plot3d([u,v,W1],u=0..2,v=0..2,color=green):
W2:=subs({U[1]=u,U[2]=v},z2[3]):
GP2:=plot3d([u,v,W2],u=0..2,v=0..2,color=red):
elif
z1[1]=Z[1] and z1[3]=Z[3] then
W1:=subs({Z[1]=u,Z[3]=v},z1[3]):
GP1:=plot3d([u,W1,v],u=0..2,v=0..2,color=green):
W2:=subs({U[1]=u,U[3]=v},z2[3]):
GP2:=plot3d([u,W2,v],u=0..2,v=0..2,color=red):
else
W1:=subs({Z[2]=u,Z[3]=v},z1[1]):
GP1:=plot3d([W1,u,v],u=0..2,v=0..2,color=green):
W2:=subs({U[2]=u,U[3]=v},z2[1]):
GP2:=plot3d([W2,u,v],u=0..2,v=0..2,color=red):
end if:
plots[display](GP1,GP2,axes=BOXED,title="Плос
кости параллельны"):
end if: end proc:

```

Покажем пример ввода этой программной процедуры для исследования взаимного расположения двух плоскостей, заданных парой общих уравнений:

$$\{2x-2y+z=21; \quad x+2y-z=-9\} : -$$

```
> EqLine3([2*x-2*y+z=21,x+2*y-z=-9],t);
```

и ее исполнения (Рис. 2). Как видно из этого рисунка, исполнение команды дает изображение двух плоскостей, линию их пересечения и название рисунка, содержащее информацию о взаимном расположении плоскостей.

Плоскости пересекаются по красной прямой

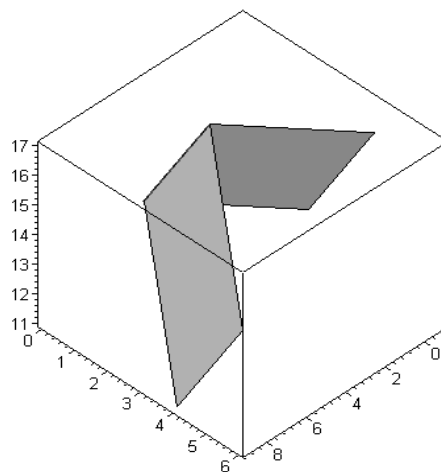


Рис.2. Исполнение команды *EqLine3([2\*x-2\*y+z=21,x+2\*y-z=-9],t)*

<sup>2</sup> Здесь *t* – параметр прямой пересечения.

Рассмотрим другой пример параллельного расположения плоскостей:

$$\{7x-4y+z=0; 21x-12y+3z=29\} : -$$

>EqLine3([7\*x-4\*y+z=0,21\*x-12\*y+3\*z=29],t);  
Плоскости параллельны

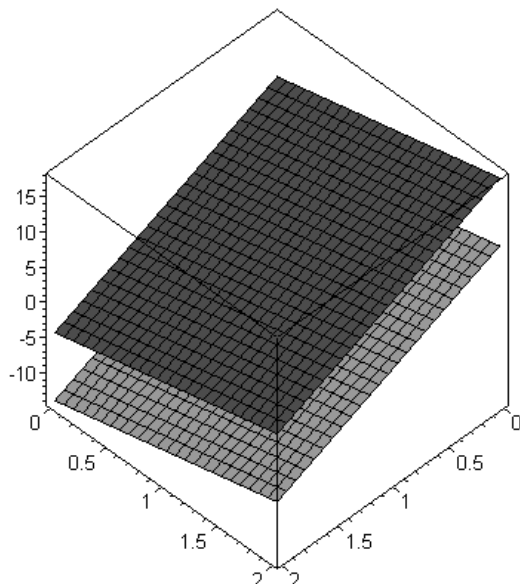


Рис.3. Исполнение команды EqLine3([7\*x-4\*y+z=0,21\*x-12\*y+3\*z=29],t)

#### 4.2. Компьютерное моделирование взаимного расположения прямых

Создадим теперь в библиотеке Anal\_Geom программную процедуру  $graph3D(M1,q1,M2,q2,a,b,A,B,c1,c2,c3)$  полного исследования и геометрического моделирования взаимного расположения двух прямых, заданных парой опорных точек  $M1,M2$  и парой направляющих векторов  $q1,q2$  ( $a,b$  – коэффициенты,  $A,B$  – обозначение прямых,  $c1,c2,c3$  – цветовые опции прямых и их общего перпендикуляра, если таковой существует):

```
> Anal_Geom [
graph3D]:=proc(M1,q1,M2,q2,a,b,A,B,c1,c2,c3) local
gd1,gd2,
M3,MM3,m3,GM3,q3,gdcr,M4,MM4,m4,GM4,GM5,
GM6:
gd1:=graph_d(M1,q1,a,b,c1):
gd2:=graph_d(M2,q2,a,b,c2):
M3:=cross_par(M1,q1,M2,q2)[1]:
M4:=cross_par(M1,q1,M2,q2)[2]:
MM3:=convert(M3,list):
MM4:=convert(M4,list):
m3:=convert(M(op(MM3)),name):
m4:=convert(N(op(MM4)),name):
q3:=cross_par(M1,q1,M2,q2)[3]:
gdcr:=graph_d(M3,q3,0,1,c3):
```

```
GM3:=plots[textplot3d]([op(MM3),m3],align =
{ABOVE,RIGHT},color=NAVY,font=[TIMES,ROMAN,8]);
```

```
GM4:=plots[textplot3d]([op(MM4),m4],align =
{ABOVE,RIGHT},color=NAVY,font=[TIMES,ROMAN,8]);
```

```
GM5:=plots[textplot3d]([M1[1]+0.2,M1[2]+0.2,M1[3]+0.2,`A`],align =
{ABOVE,LEFT},color=c1,font=[TIMES,ROMAN,10]);
```

```
GM6:=plots[textplot3d]([M2[1]+0.2,M2[2]+0.2,M2[3]+0.2,`B`],align =
{BELOW,RIGHT},color=c2,font=[TIMES,ROMAN,10]);
```

```
plots[display](gd1,gd2,gdcr,GM3,GM4,GM5,GM6,axes=BOXED,labels=[X,Y,Z],title=`Скрещивающиеся прямые a и b;
```

```
прямая MN – кратчайший перпендикуляр`):
```

```
end proc:
```

Приведем пример ввода и исполнения программы исследования взаимного расположения двух прямых (Рис. 4):

```
> graph3D([0,1,-2],[2,0,1],[-1,-1,2],[1,2,-1],-2,2,a,b,blue,COLOR(0,0.5,0.5),black);
```

Скрещивающиеся прямые a и b;

прямая MN - кратчайший перпендикуляр

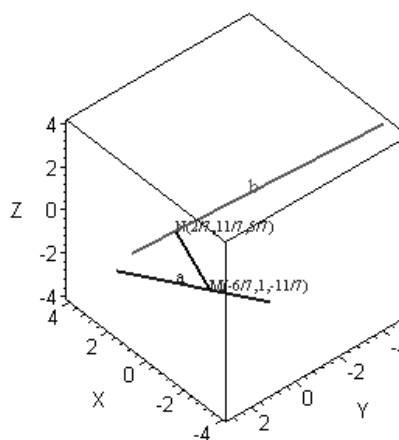


Рис.4. Исполнение команды graph3D([0,1,-2],[2,0,1],[-1,-1,2],[1,2,-1],-2,2,a,b,blue,COLOR(0,0.5,0.5),black)

#### 5. Заключение

Таким образом, создана библиотека программных процедур по курсу линейной алгебры и аналитической геометрии, позволяющая простыми командами проводить исследование линейных алгебраических уравнений и линейных геометрических объектов с графическим отображением результатов исследования. Данная библиотека программных процедур сопровождается подробной инструкцией с конкретными примерами, созданной в пакете Maple.

\*\*\*\*\*

1. *Атанасян Л.С., Базылев В.Т.* Геометрия. – М.: Просвещение, 1987. – Ч.1. – 352 с.
2. *Игнатъев Ю.Г.* Курс лекций по аналитической геометрии. – Казань: Изд-во НИЛИТМО, ТГГПУ, 2005. – Ч.1, II. – 124 с.
3. *Дьяконов В.* Maple 7. Учеб. курс. – СПб.: Питер, 2002. – 672 с.
4. Проблемы информационных технологий в математическом образовании: учеб. пособ. / Под ред. Ю.Г. гнатъева. – Казань: ТГГПУ, 2005. – 118 с.
5. *Матросов А.В.* Maple 6. Решение задач высшей математики и механики. – СПб: БХВ-Петербург, 2001. – 528 с.
6. *Игнатъев Ю.Г., Самигуллина А.Р.* Библиотеки пользовательских программных процедур для методического сопровождения курса высшей математики. Труды Математического центра имени Н.И.Лобачевского: Материалы Восьмой молодежной научной школы-конференции "Лобачевские чтения – 2009". – Казань: Казан. матем. об-во, 2009. – Т.39. – 417 с.
7. *Самигуллина А.Р.* Создание компьютерных средств сопровождения курса линейной алгебры и аналитической геометрии для нематематических факультетов с помощью СКМ Maple. Системы компьютерной математики и их приложения: материалы международной конференции. – Смоленск: СмолГУ, 2009. – Вып.10. – 303 с.
8. *Игнатъев Ю.Г., Самигуллина А.Р.* Библиотека программных графических процедур по курсу алгебры и аналитической геометрии в пакете Maple. Системы компьютерной математики и их приложения: материалы XI международной научной конференции, посвященной 70-летию профессора В.П.Дьяконова. – Смоленск: СмолГУ, 2010. – Вып.11. – 342 с.

## THE MATHEMATICAL MODELLING OF OBJECTS OF LINEAR ALGEBRA AND ANALYTIC GEOMETRY IN THE COMPUTER MATHEMATICAL SYSTEM MAPLE

A.R.Samigullina

The basic principles and programme procedures of the mathematical modelling in the computer mathematical system Maple for the graphical representations of the Linear Algebra and Analytic Geometry basic objects are described in the article.

**Key words:** mathematical modelling, computer mathematical systems, algorithms, Linear Algebra, Analytic Geometry.

\*\*\*\*\*

**Самигуллина Алсу Ринатовна** – аспирант кафедры геометрии и математического моделирования Татарского государственного гуманитарно-педагогического университета.

E-mail: [ffimo@tggpu.ru](mailto:ffimo@tggpu.ru)