

## ЛЕКЦИИ 11

### Тема. МОДЕЛИРОВАНИЕ ОБЪЕКТОВ АЛГЕБРЫ И АНАЛИТИЧЕСКОЙ ГЕОМЕТРИИ И МАТЕМАТИЧЕСКОГО АНАЛИЗА В MAPLE

#### Программные процедуры решения систем линейных алгебраических уравнений

Учебный модуль высшей алгебры является составной и базовой частью курса высшей математики для нематематических факультетов. Основная цель этого модуля – изучение теории линейных алгебраических уравнений, необходимой, как в других модулях курса высшей математики и предметах естественно - научного цикла, так и имеющей самостоятельную ценность для решения многочисленных прикладных задач. Кроме того, в этот модуль включается изучение основ матричного исчисления и теории определителей, необходимых для изучения теории систем линейных алгебраических уравнений (СЛАУ) и также имеющих многочисленные приложения в других модулях курса высшей математики и имеющих самостоятельную ценность. Поэтому информатизация этого модуля имеет большое значение для изучения курса высшей математики.

В работе [151] были рассмотрены основные принципы математического и компьютерного моделирования объектов линейной алгебры и аналитической геометрии в системе компьютерной математики (СКМ) Maple. В частности, в указанной работе была представлена программа автоматизированного решения системы линейных алгебраических уравнений в СКМ Maple с выводом решений в стандартной для Российской системы образования форме. Однако, описанная в [151] программная процедура, обладает существенным недостатком: при наличии нулевых коэффициентов перед неизвестными эти переменные не считываются и соответствующие нули не попадают в расширенную матрицу системы. Кроме того, встроенные программные процедуры Maple не содержат программы нахождения фундаментальных решений СЛАУ, имеющих большую ценность в многочисленных приложениях и составляющих основу математической культуры. Решение указанных двух задач программными способами в СКМ Maple потребовало значительного усложнения программных процедур. Здесь следует отметить общее правило создания программных продуктов: чем менее профессиональным является пользователь программного продукта, тем большая степень программного сервиса требуется от этого продукта, а значит, и более сложные со-ответствующие программные процедуры. Ниже мы укажем пути решения этих задач и опишем библиотеку соответствующих программных процедур Algebra, [152]. Указанные ниже коды описаны в программе [153].1

$$AX = B \Rightarrow \sum_{i=1}^n a_{ki}x_i = b_k; \quad (k = \overline{1, m}), \quad (\text{II.1})$$

где,  $A$  - основная матрица системы,  $B$  - матрица-столбец свободных членов,  $X$  - матрица-столбец неизвестных:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}; \quad B = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}; \quad X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}. \quad (\text{II.2})$$

Пользователь вводит указанную систему в Maple не в матричной, а в стандартной

форме упорядоченного списка уравнений:

$$\text{Sys} := [\alpha_1 * x + \alpha_2 * y + \dots + \alpha_n * u = b_1, \beta_1 * x + \beta_2 * y + \dots + \beta_n * u = b_2, \dots, \delta_1 * x + \delta_2 * y + \dots + \delta_n * u = b_m],$$

Где,  $\alpha_i, \beta_i, \dots, \delta_i, b_j$  коэффициенты уравнений (V.5), т.е., конкретные числа, а  $x, y, \dots, u$  - имена переменных, которые в конкретном задании могут быть произвольными. Поэтому первой задачей является компьютерное распознавание системы (III.3) и приведение её к стандартному виду (V.5), а также нахождение расширенной матрицы системы:

$$\tilde{A} = \left( \begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right) \quad (\text{II.4})$$

Итак, опишем библиотеку Algebra автоматизированного решения систем линейных алгебраических уравнений. Для создания библиотеки вводим пустую таблицу с указанным наименованием.

```
>restart:
Algebra:=table():
```

Процедуры распознавания системы линейных алгебраических уравнений. Для распознавания системы (V.5) библиотека Algebra имеет программную процедуру Algebra[InfoEq], которая предоставляет информацию о одиночном линейном уравнении в виде упорядоченного списка, первый элемент которого есть упорядоченный список имен переменных уравнения, второй элемент - упорядоченный список коэффициентов при этих переменных, третий элемент - правую часть уравнения.

```
> Algebra[InfoEq]:=proc(Eq) local n,i,xxx:
n:=Algebra[number_members](Eq):
xxx:=(i)->Algebra[coef_var](Eq,i):
[[seq(xxx(i)[2],i=1..n)], [seq(xxx(i)[1],i=1..n)],
rhs(Eq)]:end proc:
```

Здесь использована процедура number\_coef, которая находит коэффициент у  $i$ -той переменной, причем если коэффициент равен 1, т.е., множитель перед переменной отсутствует, то результат действия программы будет 1, в других ненулевых случаях результат равен  $i$ :

Процедура number\_members находит количество переменных в левой части уравнения

```
> Algebra[number_members]:=(Eq)->nops(lhs(Eq)): Пример. Количество переменных уравнения  $x+2*y-3*z=7$  равно 3: (x,y,z) Продemonстрируем действие этой процедуры:
```

```
> Algebra[number_members](x+2*y-3*z=7);
```

Процедура coef\_var создает упорядоченный список [k,z] для  $i$ -той переменной в уравнении, где k-коэффициент перед  $i$ -той переменной в уравнении, z-имя этой переменной

```

> Algebra[coef_var]:=proc(Eq,i) local xx,xxx: >
  xx:=[op([op([op(lhs(Eq))]))][i]):
> if Algebra[number_coef](Eq,i)=2 then xxx:=xx:> else
xxx:=[1,op(xx)]:end if:xxx:end proc:

```

Пример

```

> Algebra[coef_var](x+2*y-3*z=7,2); [2; y]
> Algebra[coef_var](x+2*y-3*z=7,1); [1; x]

```

Процедура InfoEq представляет информацию о единичном линейном уравнении в виде упорядоченного списка, первый элемент которого представляет собой упорядоченный список имен переменных уравнения, второй элемент - упорядоченный список коэффициентов при этих переменных, третий элемент - правую часть уравнения, т.е., фактически записывает строку расширенной матрицы, соответствующей этому уравнению.

```

> Algebra[InfoEq]:=proc(Eq) local n,i,xxx: >
  n:=Algebra[number_members](Eq):
> xxx:=(i)->Algebra[coef_var](Eq,i):
> [[seq(xxx(i)[2],i=1..n)],[seq(xxx(i)[1],i=1..n)],rhs(Eq)]:end
proc:

```

Пример: > Algebra[InfoEq](x+2\*y-3\*z=7); [[x; y; z]; [1; 2; 3]; 7]

Процедура associate находит объединение подмножеств x, представленных в виде  $x=\{a_1, \dots, a_r\}, \{b_1, b_2, \dots, b_s\}, \dots, \{c_1, \dots, c_m\}$ . Конечное множество записывает в виде упорядоченного списка:

```

> Algebra[associate]:=proc(x) local elem,u: >
  u:=x[1]:
> for elem in x
do > u:=u union
  elem:
> end do:[op(u)]: end proc:

```

Пример:

```

> Algebra[associate]([x,y,z],{y,u,z},{y,t,w},{r,u,s});
[r; s; t; u; w; y; z; x]

```

```

> Algebra[associate]:=proc(x) local elem,u: >
  u:=x[1]:
> for elem in x
do > u:=u union
  elem:
> end do:[op(u)]: end proc:

```

Пример:

```

> Algebra[associate]([x,y,z],{y,u,z},{y,t,w},{r,u,s});
[r; s; t; u; w; y; z; x]

```

Формирование системы уравнений в стандартном виде. Процедура StandartSys представляет систему линейных уравнений в стандартном виде с унифицированными именами переменных  $X_i$ , где  $i$  изменяется в пределах от 1 до nnn - числа неизвестных:

```

> Algebra[StandartSys]:=proc(Eqs)

```

```

> local nn,Eq,i,vars,Vars,nnn,k,SB,
> EQS:
> nn:=nops(Eqs):
> Eq:=(i)->Eqs[i]:
> vars:=(i)->{op(Algebra[InfoEq](Eq(i))[1])}:
> Vars:=Algebra[associate]({seq(vars(i),i=1..nn)}):>
nnn:=nops(Vars):
> SB:={seq(Vars[k]=X[k],k=1..nnn)}:
> EQS:=subs(SB,Eqs):
> EQS:
> end proc:

```

Пример:

```

> Algebra[StandartSys]([x+2*y-z=5,\ >
x+y-3*z=7,5*x-3*y+2*z=9,x+y+z=1]);

```

```

>[X[1]+2*X[2]-X[3] = 5, X[1]+X[2]-3*X[3] = 7,
5*X[1]-3*X[2]+2*X[3] = 9,
X[1]+X[2]+X[3] = 1]][[X_{{1}}+2*X_{{2}}-X_{{3}}=5,
X_{{1}}+X_{{2}}-3*X_{{3}}=7, 5*X_{{1}}-
3*X_{{2}}+2*X_{{3}}=9, X_{{1}}+X_{{2}}+X_{{3}}=1]]

```

Процедура Algebra[MatrSys] создает упорядоченный список, состоящий из двух матриц - основной матрицы системы и расширенной:

```

Algebra[MatrSys]:=proc(Eqs) local nn,Eq,i,vars,JJ,Vars,nnn,k,SB,
EQS,aa,bb,AA,AB,AAA,BBB:
nn:=nops(Eqs):

```

Созданная библиотека процедур приспособлена для студентов нематематических факультетов, удобна и проста в работе. Она может быть использована и преподавателями для проверки работ студентов, а также для генерации заданий. Кроме того, библиотека может быть использована и исследователями нематематического профиля для проведения исследовательских работ.