

## К лабораторной работе №3

### Символьные вычисления

#### 1 Использование символьных переменных

Перед использованием символьные переменные и функции в MatLab должны быть объявлены как символьные. Это отличает MatLab от таких программ как Mathematica или Maple, где переменные, если им не присвоено никаких значений или присвоены значения выражений, составленных из числовых констант, рассматриваются как символьные. Объявление символьных переменных выполняется командой **syms**.

Например, **syms x a b**

Здесь мы создали три символьных переменных. Символьные выражения конструируются из символьных переменных

**f=(sin(x)+a)^2\*(cos(x)+b)^2/sqrt(abs(a+b)))**

Выражение **f** автоматически становится символьным. Чтобы увидеть выражение в более привычном виде можно использовать функцию **pretty**

Символьную функцию можно создать без предварительного объявления переменных при помощи команды **sym**, входным аргументом которой является строка с выражением, заключенная в апострофы

**z = sym ('c^2 / (d + 1)')**

Эта же функция **sym** может быть использована для объявления символьных переменных. Команда **syms a, b, c** эквивалентна последовательности **a=sym('a'); b=sym('b'); c=sym('c');**

Функция **sym** может использоваться для работы с символьными константами

**sym(2) / sym(5) + sym(1) / sym(3); syms x; x+3.1**

Символьные переменные могут являться элементами матриц и векторов. Элементы строк матриц при вводе отделяются пробелами или запятыми, а столбцов — точкой с запятой, так же, как и для обычных матриц. В результате образуются символьные матрицы и векторы, к которым применимы матричные и поэлементные операции.

**syms a b c d e f g h**

**A=[a b; c d]; B=[e f; g h]; C=A\*B**

Конструирование блочных символьных матриц не отличается от числовых, требуется следить за размерами соответствующих блоков:

**D=[C A; B C]**

Обращение к элементам символьных матриц и векторов производится при помощи индексации, в том числе двоеточием и вектором со значениями индексов. Например, выделим вторую строку предыдущей матрицы

**d2=D(2,:)**

Из полученного вектора выберем 1 – й, 3 – й и 4 – й элементы

**d=d2([1 3 4])**

Для удаления строк или столбцов символьных матриц используется пустой массив. Для преобразования значения числовой переменной в символьную служит функция **sym**. Введите массив типа *double array* (обычная матрица MatLab)

**A=[1.3 -2.1 4.9 6.9 3.7 8.5]**

Здесь перенос ввода во вторую строку (в командном окне) выполняется нажатием комбинации клавиш Shift – Enter. Соответствующий символьный массив

**B=sym(A)**

Создайте символьный вектор – столбец **d**

**c=[3.2; 0.4; -2.1]**

**d=sym(c)**

Умножьте матрицу **B** на вектор **d** – результат является символьной переменной, причем все вычисления проделаны над рациональными дробями.

```
e=B*d
```

Символьные операции позволяют находить значение выражений и символьных констант со сколь угодно большой точностью. В частности значение рациональных дробей, функций от целых и рациональных аргументов, от числа  $\pi$  и т.д. можно получить с любой степенью точности, т.е. найти сколь угодно много значащих цифр результата. Для этого предназначена функция **vpa**:

```
c=sym('sqrt(2)'); cn=vpa(c)
```

По умолчанию удерживается 32 значащие цифры. Второй дополнительный входной параметр **vpa** служит для задания точности:

```
cn=vpa(c,60)
```

```
pp=sym('pi'); vpa(pp,60)
```

Второй аргумент **vpa** задает удерживаемое число значащих цифр только для данного вызова **vpa**. Важно понимать, что выходной аргумент функции **vpa** является символьной переменной. Для перевода символьных переменных в числовые, т. е. в переменные типа *double array*, используется функция **double cnum=double(cn)**

Переменная **cn** выглядела как число. Однако она являлась символьной переменной и ее можно использовать в арифметических операциях только с другими символьными переменными. Переменную **cnum** можно использовать в обычных выражениях MatLab.

При необходимости очистить рабочее пространство от всех символьных переменных следует использовать команду

```
maple restart\
```

## 2 Графическое представление символьных функций

Построение графика символьной функции одной переменной осуществляется при помощи функции **ezplot**. Самый простой вариант использования **ezplot** состоит в указании символьной функции в качестве единственного входного аргумента. В этом случае в графическое окно выводится график функции на отрезке  $[-2\pi, 2\pi]$

```
f=sym('x^2*sin(x)');
```

```
ezplot(f)
```

Обратите внимание, что автоматически создается соответствующий заголовок. По умолчанию в качестве отрезка, на котором строится график, принимается промежуток пересечения области определения функции и интервала  $[-2\pi, 2\pi]$ . Вторым аргументом может быть задан вектор с границами отрезка, на котором требуется построить график функции **ezplot (f, [-3,2])**

Другие графические функции группы ez... также можно использовать для построения графиков символьных функций. Вот несколько примеров

```
syms s t; ezmesh(exp(-s)*cos(t),exp(-s)*sin(t),t,[0,8,0,4*pi])
```

```
syms u v; ezmesh(u, (abs(sqrt(1-u^2)+v)-abs(sqrt(1-u^2)-v))/2,u,[-1,1,-1,1],21) syms x; ezplot(erf(x)); grid
```

```
syms t; ezplot3(sin(t), cos(t), t, [0,6*pi])
```

```
syms t; ezpolar(1+cos(t)) % график в полярных координатах
```

```
f=sym('x^2 +y^2'); ezsurf(f,[-1 1 -1 1])
```

Функция **ezsurf** отображает график символьной функции только для допустимых значений аргументов, остальные значения отбрасываются, что позволяет исследовать область определения функции двух переменных. Например, **ezsurf('asin(x^y)', [0 6 -7 7])**

Функции **ezcontour**, **ezcontourf** используются для построения контурных графиков – линий уровня функций двух переменных. Второй параметр задает прямоугольную

область изменения аргументов, третий  $n$  – количество ячеек сети  $n \times n$  на которую разбивается область изменения аргументов (по – умолчанию  $n=60$ )

```
syms x y
f = 3*(1-x)^2*exp(-(x^2)-(y+1)^2) - 10*(x/5 - x^3 - y^5)*exp(-x^2-y^2) - 1/3*exp(-(x+1)^2 - y^2); ezcontour(f,[3,3],49)
ezcontourf(f, [-3,3, -3, 3],60)
```

### 3 Алгебраические преобразования

Алгебраические преобразования с символьными переменными рассмотрим на примерах.

```
syms x; 27*x+12*x
```

Операции с полиномами реализуют функции **collect**, **expand**, и **factor**. Функция **expand** раскрывает скобки

```
p=sym('(x+a)^4+(x-1)^3-(x-a)^2-a*x+x-3'); expand(p)
```

```
syms a b; expand((a+b)^5)
```

Аргумент **expand** может быть не только полином, но и символьным выражением, содержащим тригонометрические, экспоненциальную и логарифмическую функции

```
f=sym('sin(arccos(3*x))+exp(2*log(x))'); fe=expand(f)
```

Вычисление коэффициентов при степенях независимой переменной производится с использованием функции **collect**. Введите следующий полином

```
p=sym('(x+a)^4+(x-1)^3-(x-a)^2-a*x+x-3'); pc=collect(p)
```

По умолчанию в качестве переменной выбирается  $x$ , однако можно было считать, что,  $a$  — независимая переменная, а  $x$  входит в коэффициенты полинома, зависящего от  $a$ . Второй аргумент функции **collect** предназначен для указания переменной, при степенях которой следует найти коэффициенты **pca=collect(p,'a')**

Символьные полиномы разлагаются на множители функцией **factor**, если получающиеся множители имеют рациональные коэффициенты

```
p=sym('x^5+13*x^4+215/4*x^3+275/4*x^2-27/2*x-18'); pf=factor(p)
```

```
syms a b; factor(a^2+2*a*b+b^2)
```

```
factor(a^2-4/9)
```

Функция **factor** умеет работать и с выражениями некоторых других типов, например, с тригонометрическими выражениями

```
syms x; factor(sin(x)^2-cos(x)^2)
```

Преобразование выражений общего вида производится при помощи функции **simplify**, которая реализует алгоритм упрощения выражений, содержащих тригонометрические, экспоненциальную, логарифмическую функции, и некоторые специальные функции. Кроме того, **simplify** способна преобразовывать выражения, содержащие суммирование и интегрирование.

```
rho = sym('(1 + sqrt(5))/2'); f = rho^2 - rho - 1
```

```
simplify(f)
```

```
simplify(sin(x)^2+cos(x)^2)
```

```
simplify((x^2+a*x+b*x+a*b)/(a+x))
```

Для вычисления значения выражения в точке следует использовать функцию подстановки в виде **R = subs(S,old,new)**

```
syms x; f = 2*x^2 - 3*x + 1; subs(f,2)
```

Здесь MatLab сам определяет имя символьной переменной в выражении **f** и заменяет ее заданным значением 2. Если символьных переменных несколько, то следует указать имя переменной, которая должна быть заменена значением

```
syms a x; f=a*x^2-x; subs (f, x,2)
```

Одновременная замена нескольких символьных переменных может быть выполнена следующим образом

```
syms a b; subs(cos(a)+sin(b), {a,b},{sym('c'),2})
```

В символьное выражение вместо символьной переменной можно подставить вектор или матрицу. Результатом будет объект той же размерности (вектор или матрица). Вот примеры табулирования функции

```
y=subs ('x^2','x', [0 1 2 3])
```

```
syms x y; subs (x^2+y^2, x, [0 1 2])
```

```
syms x y; subs (x^2+y^2, {x,y},{[0 1 2],[3 4 1]})
```

```
subs (x*y, {x, y}, {[0 1; -1 0], [1 -1; -2 1]})
```

Для вычисления композиции двух функций используется функция

```
compose
```

```
syms x y z t u; f = 1/ (1 + x^2); g = sin(y); compose(f,g)
```

```
compose(f,g,z)
```

Если задана функция  $f(x)$ , то обратная функция  $g = f^{-1}$  равенству  $g(f(x))=x$ . Символьное представление обратной функции может быть получено с помощью функции **finverse**

```
syms u v; finverse(exp(u-2*v), u)
```

Для вычисления целой и дробной части символьного числа можно использовать следующие функции: **fix** – округление до целого в сторону нуля; **floor** – вычисление наибольшего целого, не превосходящего данное; **round** – округление до ближайшего целого; **ceil** – вычисление ближайшего большего целого; **frac** – вычисление дробной части числа  $\text{frac}(X) = X - \text{fix}(X)$

```
x = sym(-5/2); [fix(x) floor(x) round(x) ceil(x) frac(x)]
```

MatLab хорошо умеет работать с разрывными функциями

```
F = 2/3*atan(1/3*tan(1/2*x)); ezplot(F)
```

```
J = sym(2*pi/3)*sym('round(x/(2*pi))');
```

```
ezplot(J,[-6 6])
```

```
ezplot(F1)
```

#### 4 Элементы математического анализа

MatLab умеет выполнять основные операции математического анализа, в частности, дифференцирование и интегрирование символьных выражений

```
diff(x^4,x); diff(x^4,x,3)
```

```
y1=sin(x)/x; limit(y1,x,0); y2=(1-exp(-x))/x;limit(y2,x,inf)
```

Для дифференцирования постоянной ее следует объявить символьной

```
c = sym('5'); diff(c); diff(5; int((x+a)^3,x); int(-2*x/(1+x^2)^2)
```

```
syms x z; int(x/(1+z^2),z); sym s t; int(2*x, sin(t), 1)
```

При вычислении интегралов может получиться следующая ситуация

```
syms a x; f = exp(-a*x^2); int (f, x, -inf, inf)
```

Не зная ничего о символьной переменной **a**, MatLab вернул два результата – первый получается, когда интеграл сходится, второй – когда расходится. Но если объявить переменную **a** положительной, то сразу получаем результат

**syms a positive; f = exp(-a\*x^2); int (f, x,-inf, inf)**

Функция **diff** умеет дифференцировать выражения любой степени сложности. Однако она не всегда возвращает результат в самой простой форме. Тогда результат следует упростить

**diff(sin(x)^2-cos(x)^2, x); simplify(diff(sin(x)^2-cos(x)^2, x))**

Функция **simplify** не справилась со своей задачей упрощения. Однако функция **simple**, которая пытается найти представление выражения с меньшим числом символов, чем исходное, получила более простой результат **simple(diff(sin(x)^2-cos(x)^2, x))**

Функция **simplify** срабатывает, когда в ее аргументе стоит неопределенный интеграл

**Simplify (x^3-int (x^2, x))** Символьное ядро MatLab хорошо умеет работать с конечными и бесконечными суммами и рядами. Вот примеры суммирования рядов

**syms x k; s1 = symsum (1/k^2,1, inf)**

**s2 = symsum (x^k, k, 0, inf)**

**symsum(k^2,0,10); symsum(x^k/sym('k!'), k, 0, inf)**

Разложение функций в ряд Тейлора также возможно

**syms x; f = 1/(5+4\*cos(x)); T = taylor(f,8)**

С некоторыми символьными функциями работать не очень удобно. Например, чтобы вычислить факториал (символьный, следовательно с любым количеством значащих цифр) надо выполнить пару команд

**kfac = sym('k!');**

**subs (kfac, k,50)**

### Задание:

А)

1. Записать символьно выражение  $f=0,5*x+5,5$
2. Записать выражение в привычном виде  $f=(2*x^a)/\sin(x)+5*a*x$

Б) Записать матрицу А в символьном виде, выделить из нее соответствующие столбы или строки и провести над ними операции. Выделить первый столбец и умножить его на вектор d.

$$A = \begin{bmatrix} a*b & 0 & 18*a & 0,5*b \\ 0 & a+b & 1 & 0,5*a \\ 18*a & 0,5*b & 0 & a+b \\ 1 & 0,5+a & a*b & 0 \end{bmatrix}$$
$$d = [0 \ 2,5 \ 5 \ 7,5]$$

Выделить центральную часть матрицы размером (2X2), умножить её на векторный столбец g

$$g = [0,2; 1,5*a].$$

В) Найти значение иррационального числа с заданной точностью.

1. Найти значение корня квадратного из двух с точностью до 22 знака после запятой.
2. Найти значение числа Пи с точностью до 8 знака после запятой

Г) Построить график функции  $f(x)=2-2*\sin(x)+[\sin(x)*\sqrt{|\cos(x)|}]/[\sin(x)+7/5]$  в символьных переменных. (*ezpolar*)

Д) Вычислить коэффициенты при степенях независимой переменной для функции  $f=((a+b)^3)-((a-b)^2)+2*a*b-2$ .

1. Для независимой переменной a
2. Для независимой переменной b

Е) Упростить выражение

$$f=((1-\sqrt{3})/2)^3+((1-\sqrt{3})/2)^2-(1-\sqrt{3})/2+1.$$

Ж) Найти значение функции в точке для переменной b=5 и для a=5 и b=12 вместе.  
 $f=a^2+2*b^2+a*b-5$

З) Продифференцировать выражение  $f=x^4+2*x^3-x^2+5$ . Найти его третью производную по x.

И) Найти предел функции  $f=(-15+3*n)/(\sin(n/2)+6*n)$  при n стремящемся к бесконечности (inf).

К) Найти факториал 32! (в символьном выражении)