

4 Символьные вычисления в среде MATLAB

В последней лабораторной работе мы рассмотрим использование возможностей символьных вычислений математического пакета MATLAB для решения задач вариационного исчисления. Ниже излагаются основные сведения о работе с символьными выражениями. Более детальную и полную информацию можно получить, воспользовавшись встроенной справочной системой MATLAB. Для этого достаточно выполнить команды *help symbolic* или *doc symbolic*.

4.1 Символьные выражения

4.1.1 Объявление символьных переменных и констант

В процессе символьных вычислений используются переменные и константы особого типа, так называемые символьные объекты. Хотя обычно в коде MATLAB тип переменных определяется динамически и нет нужды объявлять его явно, для символьных объектов дело обстоит иначе. Для объявления символьных переменных служит команда *syms*, которая в качестве аргументов принимает имена переменных, перечисленные через пробел. Например, так:

```
>> syms x y
>> syms a b real % объявляемые объекты обозначают вещественные переменные
```

Объявление символьных констант осуществляется при помощи функции *sym*. Она может принимать в качестве аргумента строку, содержащую специальные переменные, численное выражение или вызов функции, как в примерах ниже:

```
>> sym_pi = sym('pi')
>> sym_delta = sym('1/10')
>> sym_sqrt2 = sym('sqrt(2)')
```

Использование символьных констант полезно тем, что вычисления с ними производятся точно (т.е. без вычислительных погрешностей) до тех пор, пока не потребуются вычислить некоторое числовое значение. Заметим, что при выводе содержимого рабочего пространства командой *whos* символьные переменные и константы отображаются как представители класса *sym object*.

4.1.2 Символьные выражения и манипуляции над ними

После объявления, с символьными переменными можно обращаться примерно так же, как и с обычными числовыми. В частности, для них определены операторы $+$ $-$ $*$ $/$ $^$, с помощью которых можно составлять символьные выражения:

```
>> syms s t A
>> f = s^2 + 4*s + 5
f =
s^2 + 4*s + 5
>> g = s + 2
g =
s + 2
>> h = f*g
h =
(s^2 + 4*s + 5)*(s + 2)
>> z = exp(-s*t)
```

```

z =
exp(-s*t)
>> y = A*exp(-s*t)
y =
A*exp(-s*t)

```

В приведенных командах символьные переменные s , t , A используются для составления символьных выражений, создавая новые символьные переменные f , g , h , z , y . При этом последние автоматически объявляются символьными и их значение не вычисляется и никак не преобразуется.

При манипуляциях с символьными объектами часто бывает полезно узнать, какие *независимые* переменные содержатся в выражении, заданном строкой S . Для этой цели можно использовать функцию *findsym*. Например, продолжая пример выше, можно выполнить команду

```

>> findsym(z)
ans =
A, s, t

```

Заметим, что MATLAB всегда остается прежде всего матричным процессором и потому к символьным переменным можно свободно применять матричную и векторную запись и соответствующие встроенные операторы и функции. Приведем пример:

```

>> n = 3;
>> syms x;
>> B = x.^((0:n)')*(0:n))
B =
[ 1, 1, 1, 1]
[ 1, x, x^2, x^3]
[ 1, x^2, x^4, x^6]
[ 1, x^3, x^6, x^9]

```

Продemonстрируем некоторые простые алгебраические манипуляции, которые можно осуществлять над символьными объектами.

<code>expand(S)</code>	Раскрывает скобки в выражении S
<code>factor(S)</code>	Разлагает на множители выражение S
<code>simplify(S)</code>	Упрощает каждый элемент символьной матрицы S
<code>subs(S, oldvar, newvar)</code>	Заменяет в выражении S каждое вхождение символической переменной $oldvar$ новой переменной $newvar$

Примеры использования можно увидеть ниже:

• Раскрытие скобок

```

>> syms s;
>> A = s + 2;
>> B = s + 3;
>> C = A*B
C =
(s+2)*(s+3)
>> C = expand(C)

```

```
C =
s^2 + 5*s + 6
```

- **Разложение на множители**

```
>> syms s;
>> D = s^2 + 6*s + 9;
>> D = factor(D)
D =
(s+3)^2
>> p = s^3 - 2*s^2 - 3*s + 10;
>> P = factor(p)
P =
(s+2)*(s^2 - 4*s + 5)
```

- **Сокращение общего множителя**

```
>> syms s;
>> H = (s^3 + 2*s^2 + 5*s + 10)/(s^2 + 5);
>> H = simplify(H)
H =
s+2
>> factor(s^3 + 2*s^2 + 5*s + 10)
ans =
(s+2)*(s^2 + 5)
```

- **Подстановка переменной**

Пусть имеется выражение $H(s) = \frac{s+3}{s^2+6s+8}$ и требуется вычислить $G(s) = H(s)|_{s=s+2}$.

```
>> syms s;
>> H = (s + 3)/(s^2 + 6*s + 8);
>> G = subs(H, s, s+2)
G =
(s+5)/((s+2)^2 + 6*s + 20)
>> G = collect(G)
G =
(s+5)/(s^2 + 10*s + 24)
```

Таким образом, $G(s) = \frac{s+5}{s^2+10s+24}$.

Отметим, что конечный результат символьных преобразований далеко не всегда выглядит так, как хотелось бы пользователю и бывает неудобно работать с ним дальше.

4.2 Вычисление символьных выражений и отрисовка

Рано или поздно наступает момент, когда нам необходимо вычислить числовое значение результата выкладок или нарисовать график функции, выраженной символьным объектом. Рассмотрим, какие возможности для этого предоставляет нам MATLAB.

4.2.1 Вычисления значения выражения

`double(S)` Преобразует символьную матрицу S , заменяя ее элементы их численными значениями. Матрица S не должна содержать символьных переменных

Как следует из описания функции, в символьическом выражении перед вычислением его численного значения необходимо сделать все возможные подстановки, чтобы избавиться от свободных символьных переменных. Рассмотрим пример:

```
>> syms s;  
>> E = s^3 -14*s^2 + 65*s - 100);  
>> F = subs(E, s, 7.1)  
F =  
13671/1000  
>> G = double(F)  
G =  
13.6710
```

4.2.2 Рисование при помощи функции *ezplot*

Символическое выражение может быть нарисовано непосредственно при помощи функции *ezplot*:

`ezplot(f)` Рисует график функции $f(x)$, где f является математической функцией переменной x . Интервал, на котором изображается график, по умолчанию равен $[-2\pi, 2\pi]$.

`ezplot(f,xmin,xmax)` Рисует график на заданном интервале. Например, график кубического многочлена

$$A(s) = s^3 + 4s^2 - 7s - 10$$

на интервале $[-1, 3]$ можно нарисовать следующим образом:

```
>> syms s;  
>> A = s^3 + 4*s^2 - 7*s - 10;  
>> ezplot(A, -1, 3), ylabel('A(s)')
```

В результате мы получим график наподобие того, что приведен ниже. Отметим, что название оси абсцисс и всего графика определяется автоматически. Они могут быть изменены обычным способом (см. команды *xlabel*, *title*).

- 4.2.3 Рисование при помощи функции *plot*
- 4.3 Решение уравнений и систем уравнений
 - 4.3.1 Элементы линейной алгебры
 - 4.3.2 Решение уравнений при помощи функции *solve*
 - 4.3.3 Численное решение уравнений, заданных символьным выражением
- 4.4 Элементы математического анализа
 - 4.4.1 Дифференцирование
 - 4.4.2 Интегрирование
- 4.5 Решение дифференциальных уравнений
 - 4.5.1 Решение ОДУ при помощи функции *dsolve*
 - 4.5.2 Численное решение уравнений, заданных символьным выражением
- 4.6 Дополнительная литература