

can also be used as separate independent software. That software uses the MuPAD language, which has a completely different structure and commands than MATLAB. The Symbolic Math Toolbox is included in the student version of MATLAB. In the standard version, the toolbox is purchased separately. To check if the Symbolic Math Toolbox is installed on a computer, the user can type the command `ver` in the Command Window. In response, MATLAB displays information about the version that is used as well as a list of the toolboxes that are installed.

The starting point for symbolic operations is symbolic objects. Symbolic objects are made of variables and numbers that, when used in mathematical expressions, tell MATLAB to execute the expression symbolically. Typically, the user first defines (creates) the symbolic variables (objects) that are needed, and then uses them to create symbolic expressions that are subsequently used in symbolic operations. If needed, symbolic expressions can be used in numerical operations

The first section in this chapter describes how to define symbolic objects and how to use them to create symbolic expressions. The second section shows how to change the form of existing expressions. Once a symbolic expression has been created, it can be used in mathematical operations. MATLAB has a large selection of functions for this purpose. The next four sections (11.3–11.6) describe how to use MATLAB to solve algebraic equations, to carry out differentiation and integration, and to solve differential equations. Section 11.7 covers plotting symbolic expressions. How to use symbolic expressions in subsequent numerical calculations is explained in the following section.

## 11.1 SYMBOLIC OBJECTS AND SYMBOLIC EXPRESSIONS

A symbolic object can be a variable (without a preassigned numerical value), a number, or an expression made of symbolic variables and numbers. A symbolic expression is a mathematical expression containing one or more symbolic objects. When typed, a symbolic expression may look like a standard numerical expression. However, because the expression contains symbolic objects, it is executed by MATLAB symbolically.

### 11.1.1 Creating Symbolic Objects

Symbolic objects can be variables or numbers. They can be created with the `sym` and/or `syms` commands. A single symbolic object can be created with the `sym` command:

```
object_name = sym('string')
```

where the string, which is the symbolic object, is assigned to a name. The string can be:

- A single letter or a combination of several letters (no spaces). Examples: `'a'`, `'x'`, `'yad'`.

- A combination of letters and digits starting with a letter and with no spaces  
Examples: 'xh12', 'r2d2'.
- A number. Examples: '15', '4'.

In the first two cases (where the string is a single letter, a combination of several letters, or a combination of letters and digits), the symbolic object is a symbolic variable. In this case it is convenient (but not necessary) to give the object the same name as the string. For example, *a*, *bb*, and *x*, can be defined as symbolic variables as follows:

```
>> a=sym('a')  
a =  
a  
>> bb=sym('bb')  
bb =  
bb  
>> x=sym('x');  
>>
```

Create a symbolic object a and assign it to a.

The display of a symbolic object is not indented.

The symbolic variable x is created but not displayed, since a semicolon is typed at the end of the command.

The name of the symbolic object can be different from the name of the variable. For example:

```
>> g=sym('gamma')  
g =  
gamma
```

The symbolic object is gamma, and the name of the object is g.

As mentioned, symbolic objects can also be numbers. The numbers don't have to be typed as strings. For example, the `sym` command is used next to create symbolic objects from the numbers 5 and 7 and assign them to the variables *c* and *d*, respectively.

```
>> c=sym(5)  
c =  
5  
>> d=sym(7)  
d =  
7
```

Create a symbolic object from the number 5 and assign it to c.

The display of a symbolic object is not indented.

As shown, when a symbolic object is created and a semicolon is not typed at the end of the command, MATLAB displays the name of the object and the object itself in the next two lines. The display of symbolic objects starts at the beginning of the line and is not indented as is the display of numerical variables. The difference is illustrated below, where a numerical variable is created.