فصل ۱۱

ریاضیات نمادی

MATLAB An Introduction With Applications, 6th Edition
Dr. Amos Gilat
The Ohio State University

Slide deck by
Dr. Greg Reese
Miami University

در تمام فصول گذشته، فقط از اعداد در MATLAB استفاده شد. ورودی ها، خروجی ها، نتایج و غیره عدد بودند. گاهی نیاز به جوابهایی کلی تر هست. یک نوع از جوابهای کلی استفاده از یک نماد است که نشان دهنده هر عددی باشد

ریاضیات نمادی

- می تواند پاسخ مسأله را به صورت عبارات کوتاه بیان کند
- جواب دقیق بدون هیچ خطای عددی می دهد
- جوابهای دقیق در زمان امکان پذیر نبودن حل عددی ارائه می دهد
- می تواند منجر به درک بهتر مسأله و پاسخ آن شود

معادله درجه دو $ax^2 + bx + c = 0$

و حل آن را در نظر بگیرید

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad a \neq 0$$

- مختصر – یک معادله همه جوابها را برای بینهایت مقدار a و b و c می دهد
- حل دقیق – برای $a=9$, $b=0$, $c=-1$ جواب دقیق $x=\pm 1/3$ است و جواب عددی $x=0.333333333333333$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \qquad a \neq 0$$

- حتی اگر حل عددی ممکن نباشد مناسب است
  - $b=10^{200}, a = 1, c = 1$ به جواب نمیرسد زیرا b^2 خارج از دامنه قابل محاسبه است
- حل نمادی نشان میدهد که معادله همیشه دقیقاً دو ریشه دارد و b^2 باید حداقل برابر 4ac باشد تا ریشه ها حقیقی باشند
  - به دست آوردن این نتایج از راه حل عددی سخت تر است

جعبه ابزار Symbolic در MATLAB عملیات ریاضیات نمادی را انجام می دهد

یک شئ نمادی از متغیرها و اعدادی ساخته شده که MATLAB آنها را نمادی ارزیابی میکند، نه عددی

- می تواند متغیر یا عدد باشد

یک عبارت نمادی عبارتی ریاضی است با حداقل یک شئ نمادی

- حتی اگر یک عبارت نمادی مانند عبارت عددی به نظر برسد، MATLAB تفاوت را متوجه شده و آن را نمادی ارزیابی می کند

یک شئ نمادی میتواند یک متغیر یا یک عدد باشد. شئ نمادی را با این دستور ایجاد کنید

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

'string'نام شئ نمادی است و می تواند

- یک حرفی یا بیشتر باشد (بدون فاصله)
- ترکیبی از اعداد و ارقام باشد که با حرف شروع شود
- یک عدد باشد

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

مرسوم است (ولی نه لازم) که `object_name` با `'string'` یکسان باشد

```
>> 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.

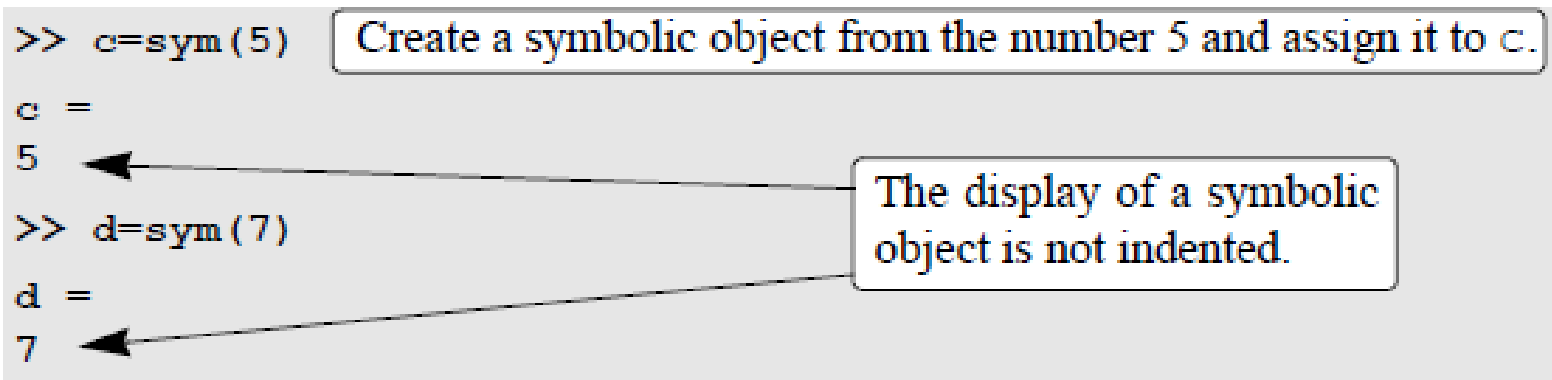
The name of the symbolic object can be different from the name of the variable. For example:

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

g =
gamma
```

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

زمانی که یک عدد نمادی را تعریف می کنید، لازم نیست آن را بین دو آپوستروف قرار دهید



```
syms variable_name1 variable_name2 variable_name3
```

- چند شئ نمادی را همزمان ایجاد می کند
- اشیاء نام یکسان با نام متغیرها دارند

```
>> syms y z d
>> y
y =
y
```

The variables created by the syms command are not displayed automatically. Typing the name of the variable shows that the variable was created,

برای ایجاد یک عبارت نمادی از این استفاده کنید

```
expression_name = mathematical expression
```

A few examples are:

```
>> syms a b c x y
>> f=a*x^2+b*x + c
f =
a*x^2 + b*x + c
```

Define a, b, c, x, and y as symbolic variables.

Create the symbolic expression $ax^2 + bx + c$ and assign it to f.

The display of the symbolic expression is not indented.

When a symbolic expression, which includes mathematical operations that can be executed (addition, subtraction, multiplication, and division), is entered, MATLAB executes the operations as the expression is created. For example:

```
>> g=2*a/3+4*a/7-6.5*x+x/3+4*5/3-1.5
```

$\frac{2a}{3} + \frac{4a}{7} - 6.5x + \frac{x}{3} + 4 \cdot \frac{5}{3} - 1.5$ is entered.

```
g =
(26*a)/21 - (37*x)/6 + 31/6
```

$\frac{26a}{21} - \frac{37x}{6} + \frac{31}{6}$ is displayed.

Notice that all the calculations are carried out exactly, with no numerical approximation. In the last example, $\frac{2a}{3}$ and $\frac{4a}{7}$ were added by MATLAB to give $\frac{26a}{21}$, and $-6.5x + \frac{x}{3}$ was added to $\frac{37x}{6}$. The operations with the terms that contain only numbers in the symbolic expression are carried out exactly. In the last example, $4 \cdot \frac{5}{3} + 1.5$ is replaced by $\frac{31}{6}$.

The difference between exact and approximate calculations is demonstrated in the following example, where the same mathematical operations are carried out—once with symbolic variables and once with numerical variables.

```
>> a=sym(3); b=sym(5);     Define a and b as symbolic 3 and 5, respectively.
>> e=b/a+sqrt(2)           Create an expression that includes a and b.
e =                        An exact value of e is displayed as a sym-
2^(1/2) + 5/3              bolic object (the display is not indented).
>> c=3; d=5;               Define c and d as numerical 3 and 5, respectively.
>> f=d/c+sqrt(2)           Create an expression that includes c and d.
f =                        An approximated value of f is displayed
    3.0809                 as a number (the display is indented).
```

An expression that is created can include both symbolic objects and numerical variables. However, if an expression includes a symbolic object (or several), all the mathematical operations will be carried out exactly. For example, if *c* is replaced by *a* in the last expression, the result is exact, as it was in the first example.

```
>> g=d/a+sqrt(2)

g =
2^(1/2) + 5/3
```

نکات دیگری در خصوص عبارات و اشیاء نمادی:

- Symbolic expressions can include numerical variables that have been obtained from the execution of numerical expressions. When these variables are inserted in symbolic expressions their exact value is used, even if the variable was displayed before with an approximated value. For example:

```
>> h=10/3
h =
    3.3333
```

h is defined to be 10/3 (a numerical variable).

An approximated value of h (numerical variable) is displayed.

```
>> k=sym(5); m=sym(7);
>> p=k/m+h
p =
85/21
```

Define k and m as symbolic 5 and 7, respectively.

h, k, and m are used in an expression.

The exact value of h is used in the determination of p.
An exact value of p (symbolic object) is displayed.

از دستور double(S) برای تبدیل یک شئ یا عبارت نمادی S به حالت عددی استفاده کنید

```
>> pN=double(p)

pN =
    4.0476

>> y=sym(10)*cos(5*pi/6)

y =
-5*3^(1/2)

>> yN=double(y)

yN =
   -8.6603
```

p is converted to numerical form (assigned to pN).

Create a symbolic expression y.

Exact value of y is displayed.

y is converted to numerical form (assigned to yN).

- A symbolic object that is created can also be a symbolic expression written in terms of variables that were not first created as symbolic objects. For example, the quadratic expression $ax^2 + bx + c$ can be created as a symbolic object named `f` by using the `sym` command:

```
>> f=sym('a*x^2+b*x+c')
f =
a*x^2 + b*x +c
```

It is important to understand that in this case, the variables a, b, c, and x included in the object do not exist individually as independent symbolic objects (the whole expression is one object). This means that it is impossible to perform symbolic math operations associated with the individual variables in the object. For example, it will not be possible to differentiate `f` with respect to x. This is different from the way in which the quadratic expression was created in the first example in this section, where the individual variables are first created as symbolic objects and then used in the quadratic expression.

میتوان از یک عبارت نمادی موجود برای ایجاد عبارات نمادی جدید استفاده کرد – کافی است از نام عبارت موجود در عبارت جدید استفاده کنید

```
>> syms x y
>> SA=x+y, SB=x-y
SA =
x+y
SB =
x-y
>> F=SA^2/SB^3+x^2
F =
(x+y)^2/(x-y)^3+x^2
```

Define x and y as symbolic variables.

Create two symbolic expressions SA and SB.

$SA = x + y$

$SB = x - y$

Create a new symbolic expression F using SA and SB.

$$F = SA^2 / SB^2 + x^2 = \frac{(x+y)^2}{(x-y)^3} + x^2$$

The `findsym` command can be used to find which symbolic variables are present in an existing symbolic expression. The format of the command is:

`findsym(S)` or `findsym(S,n)`

The `findsym(S)` command displays the names of all the symbolic variables (separated by commas) that are in the expression `S` in alphabetical order. The `findsym(S,n)` command displays `n` symbolic variables that are in expression `S` in the default order. For one-letter symbolic variables, the default order starts with x, and followed by letters, according to their closeness to x. If there are two letters equally close to x, the letter that is after x in alphabetical order is first (y before w, and z before v). The default symbolic variable in a symbolic expression is the first variable in the default order. The default symbolic variable in an expression S can be identified by typing `findsym(S,1)`. Examples:

```
>> syms x h w y d t
>> S=h*x^2+d*y^2+t*w^2
S =
t*w^2 + h*x^2 + d*y^2
>> findsym(S)
ans =
d, h, t, w, x, y
>> findsym(S,5)
ans =
x,y,w,t,h
```

Define x, h, w, y, d, and t as symbolic variables.

Create a symbolic expression S.

Use the findsym(S) command.

The symbolic variables are displayed in alphabetical order.

Use the findsym(S,n) command (n = 5).

Five symbolic variables are displayed in the default order.

```
>> findsym(S,1)
ans =
x
```

Use the findsym(S,n) command with n = 1.

The default symbolic variable is displayed.

عبارات نمادی اغلب به ساده ترین صورت یا صورت مورد نظر نیستند. میتوان آنها را اینگونه تغییر داد

- جمع کردن جملات با توان یکسان
- بسط دادن حاصلضرب ها
- فاکتورگیری
- استفاده از اتحادهای جبری و مثلثاتی
- بسیاری راههای دیگر

دستور collect :

```
collect(S)  collect(S,variable_name)
```

- جملاتی در عبارت که داری توان یکسانی از متغیر هستند را جمع می کند.
- در عبارت جدید، جملات به ترتیب نزولی توان مرتب می شوند
- حالت `collect(S)` زمانی مناسب تر است که عبارت تنها یک متغیر نمادی داشته باشد

`collect(S)` `collect(S,variable_name)`

- اگر عبارت بیش از یک متغیر داشته باشد MATLAB ابتدا جملات مربوط به یک متغیر را جمع می کند، سپس دومی و ...
  - ترتیب متغیرها توسط MATLAB تشخیص داده می شود
  - کاربر می تواند متغیر اول را با استفاده از این حالت تعیین کند `collect(S, variable_name)`

```
>> syms x y
```
Define x and y as symbolic variables.

```
>> S=(x^2+x-exp(x))*(x+3)
S =
(x + 3)*(x - exp(x) + x^2)
```
Create the symbolic expression $(x+3)(x-e^x+x^2)$ and assign it to S.

```
>> F = collect(S)
```
Use the collect command.

```
F =
x^3+4*x^2+(3-exp(x))*x-3*exp(x)
```
MATLAB returns the expression: $x^3+4x^2+(3-e^x)x-3e^x$.

```
>> T=(2*x^2+y^2)*(x+y^2+3)
T =
(2*x^2+y^2)*(y^2+x+3)
```
Create the symbolic expression T $(2x^2+y^2)(y^2+x+3)$.

```
>> G=collect(T)
```
Use the collect(T) command.

MATLAB returns the expression: $x^3+(2y^2+6)x^2+y^2x+y^2(y^2+3)$.

```
G =
2*x^3+(2*y^2+6)*x^2+y^2*x+y^2*(y^2+3)
>> H=collect(T,y)
```
Use the collect(T,y) command.

```
H =
y^4+(2*x^2+x+3)*y^2+2*x^2*(x+3)
```
MATLAB returns the expression $y^4+(2x^2+x+3)y^2+2x^2(x+3)$.

Note that when collect(T) is used, the reformatted expression is written in order of decreasing powers of x, but when collect(T,y) is used, the reformatted expression is written in order of decreasing powers of y.

دستور expand :

`expand(S)`

دستور expand به دو صورت کار می کند

1. عباراتی که شامل جمع هستند (حداقل یکی از عبارات) را در هم ضرب می کند
2. از اتحادهای مثلثاتی و قوانین توان و لگاریتم برای بسط دادن عباراتی که شامل جمع هستند استفاده می کند

```
>> syms a x y
>> S=(x+5)*(x-a)*(x+4)
S =
-(a-x)*(x+4)*(x+5)
>> T=expand(S)
T =
20*x-20*a-9*a*x-a*x^2+9*x^2+x^3
>> expand(sin(x-y))

ans =
cos(y)*sin(x)-cos(x)*sin(y)
```

Define a, x, and y as symbolic variables.

Create the symbolic expression $-(a-x)(x+4)(x+5)$ and assign it to S.

Use the expand command.

MATLAB returns the expression $20x - 20a - 9ax - ax^2 + 9x^2 + x^3$.

Use the expand command to expand $\sin(x-y)$.

MATLAB uses trig identity for the expansion.

دستور factor :

`factor(S)`

`factor` عبارتی را که به صورت چند جمله ای باشد به گونه ای تغییر می دهد تا به صورت حاصلضرب چند جمله ای هایی با توان کمتر در آید

```
>> syms x
```
Define x as a symbolic variable.

```
>> S=x^3+4*x^2-11*x-30
S =
x^3+4*x^2-11*x-30
```
Create the symbolic expression $x^3+4x^2-11x-30$ and assign it to S.

```
>> factor(S)
```
Use the `factor` command.

```
ans =
(x+5)*(x-3)*(x+2)
```
MATLAB returns the expression $(x+5)(x-3)(x+2)$.

دستور simplify :

The `simplify` command is a tool for simplifying the form of an expression. The `simplify` command uses mathematical operations (addition, multiplication, rules of fractions, powers, logarithms, etc.) and functional and trigonometric identities to generate a simpler form of the expression. The format of the `simplify` command is:

```
simplify(S)
```

where either `S` is the name of the existing expression to be simplified, or an expression to be simplified can be typed in for `S`.

Two examples are:

```
>> syms x y
>> S=(x^2+5*x+6)/(x+2)
S =
(x^2+5*x+6)/(x+2)
>> SA = simplify(S)
SA =
x+3
>> simplify((x+y)/(1/x+1/y))

ans =
x*y
```

- `>> syms x y` — Define x and y as symbolic variables.
- `>> S=(x^2+5*x+6)/(x+2)` — Create the symbolic expression $(x^2+5x+6)/(x+2)$, and assign it to `S`.
- `>> SA = simplify(S)` — Use the `simplify` command to simplify `S`.
- `SA = x+3` — MATLAB simplifies the expression to $x+3$.
- `>> simplify((x+y)/(1/x+1/y))` — Simplify $(x+y)/(\frac{1}{x}+\frac{1}{y})$.
- `ans = x*y` — MATLAB simplifies the expression to xy).

`pretty(S)` عبارات نمادی را مشابه نحوه نوشتاری آن ها در ریاضی نمایش می دهد.

مثال:

```
>> syms a b c x
>> S=sqrt(a*x^2 + b*x + c)
S =
(a*x^2+b*x+c)^(1/2)
>> pretty(S)
                      2              1/2
               (a x  + b x + c)
```

Define a, b, c, and x as symbolic variables.

Create the symbolic expression $\sqrt{ax^2 + bx + c}$, and assign it to

The `pretty` command displays the expression in a math format.

MATLAB می تواند معادلات خطی با یک یا چند مجهول را به صورت نمادی حل کند

Solving a single equation:

An algebraic equation can have one or several symbolic variables. If the equation has one variable, the solution is numerical. If the equation has several symbolic variables, a solution can be obtained for any of the variables in terms of the others. The solution is obtained by using the `solve` command, which has the form

`h = solve(eq)` or `h = solve(eq,var)`

- The argument `eq` can be the name of a previously created symbolic expression, or an expression that is typed in. When a previously created symbolic expression `S` is entered for `eq`, or when an expression that does not contain the = sign is typed in for `eq`, MATLAB solves the equation `eq` = 0.
- An equation of the form $f(x) = g(x)$ can be solved by typing the equation (including the = sign) as a string for `eq`.
- If the equation to be solved has more than one variable, the `solve(eq)` command solves for the default symbolic variable (see Section 11.1.3). A solution for any of the variables can be obtained with the `solve(eq,var)` command by typing the variable name for `var`.
- If the user types `solve(eq)`, the solution is assigned to the variable `ans`.
- If the equation has more than one solution, the output `h` is a symbolic column vector with a solution at each element. The elements of the vector are symbolic objects. When an array of symbolic objects is displayed, each row is enclosed with square brackets (see the following examples).

```
>> syms a b x y z
>> h=solve(exp(2*z)-5)
h =
log(5)/2
>> S=x^2-x-6
S =
x^2-x-6
>> k=solve(S)
k =
 -2
  3
>> solve('cos(2*y)+3*sin(y)=2')
ans =
     pi/2
     pi/6
 (5*pi)/6
>> T= a*x^2+5*b*x+20
T =
a*x^2+5*b*x+20
>> solve(T)
ans =
 -(5*b+5^(1/2)*(5*b^2-16*a)^(1/2))/(2*a)
 -(5*b-5^(1/2)*(5*b^2-16*a)^(1/2))/(2*a)
>> M = solve(T,a)
M =
-(5*b*x+20)/x^2
```

Define `a`, `b`, `x`, `y`, and `z` as symbolic variables.

Use the `solve` command to solve $e^{2z} - 5 = 0$.

The solution is assigned to `h`.

Create the symbolic expression $x^2 - 6 - x$, and assign it to `S`.

Use the `solve(S)` command to solve $x^2 - 6 - x$.

The equation has two solutions. They are assigned to `k`, which is a column vector with symbolic

Use the `solve` command to solve $\cos(2y) + 3\sin(y) = 2$. (The equation is typed as a string in the command.)

The solution is assigned to `ans`.

Create the symbolic expression $ax^2 + 5bx + 20$, and assign it to `T`.

Use the `solve(S)` command to solve $T = 0$.

The equation $T = 0$ is solved for the variable x, which is the default variable.

Use the `solve(eq,var)` command to solve $T = 0$.

The equation $T = 0$ is solved for the variable a.

حل دستگاه معادلات:

از solve برای حل دستگاه معادلات استفاده کنید

- اگر تعداد معادلات با تعداد مجهولات برابر باشد، حل به صورت عددی انجام خواهد شد
- اگر تعداد مجهولات بیشتر از تعداد معادلات باشد، حل به صورت نمادی برای مجهولات مورد نظر بر حسب مجهولات دیگر خواهد بود
- اگر دستگاه یک جواب داشته باشد، هر مجهول یک مقدار عددی در جواب خواهد داشت
- اگر دستگاه چند جواب داشته باشد، هر مجهول می تواند چند مقدار داشته باشد

```
output = solve(eq1,eq2,....,eqn)
```

or

```
output = solve(eq1,eq2,...,eqn,var1,var2,...,varn)
```

- The arguments `eq1,eq2,...,eqn` are the equations to be solved. Each argument can be a name of a previously created symbolic expression, or an expression that is typed in as a string. When a previously created symbolic expression `S` is entered, the equation is `S = 0`. When a string that does not contain the = sign is typed in, the equation is `expression = 0`. An equation that contains the = sign must be typed as a string.
- In the first format, if the number of equations n is equal to the number of variables in the equations, MATLAB gives a numerical solution for all the variables. If the number of variables is greater than the number of equations n, MATLAB gives a solution for n variables in terms of the rest of the variables. The variables for which solutions are obtained are chosen by MATLAB according to the default order (Section 11.1.3).
- When the number of variables is greater than the number of equations n, the user can select the variables for which the system is solved. This is done by using the second format of the `solve` command and entering the names of the variables `var1,var2,...,varn`.

خروجی `solve` می تواند دو حالت متفاوت داشته باشد – یک آرایه سلولی یا یک ساختار

- *آرایه سلولی* – یک آرایه است که در آن هر درایه میتواند یک آرایه باشد
  - آرایه های هر درایه می توانند ابعاد مختلفی داشته باشند
- *ساختار* – یک آرایه که در آن درایه ها نامهای متنی دارند. درایه ها $field$ نامیده می شوند

When a cell array is used in the output of the `solve` command, the command has the following form (in the case of a system of three equations):

```
[varA, varB, varC] = solve(eq1,eq2,eq3)
```

- Once the command is executed, the solution is assigned to the variables `varA`, `varB`, and `varC`, and the variables are displayed with their assigned solution. Each of the variables will have one or several values (in a column vector) depending on whether the system of equations has one or several solutions.
- The user can select any names for `varA`, `varB`, and `varC`. MATLAB assigns the solution for the variables in the equations in alphabetical order. For example, if the variables for which the equations are solved are x, u, and t, the solution for t is assigned to `varA`, the solution for u is assigned to `varB`, and the solution for x is assigned to `varC`.

The following examples show how the `solve` command is used for the case where a cell array is used in the output:

```
>> syms x y t
>> S=10*x+12*y+16*t;
>> [xt yt]=solve(S, '5*x-y=13*t')
xt =
2*t
yt =
-3*t
```

Define `x`, `y`, and `t` as symbolic variables.

Assign to `S` the expression $10x + 12y + 16t$.

Use the `solve` command to solve the system: $10x + 12y + 16t = 0$
$5x - y = 13t$

Output in a cell array with two cells named `xt` and `yt`.

The solutions for x and y are assigned to `xt` and `yt`, respectively.

In the example above, notice that the system of two equations is solved by MATLAB for x and y in terms of t, since x and y are the first two variables in the default order. The system, however, can be solved for different variables. As an example, the system is solved next for y and t in terms of x (using the second form of the `solve` command:

```
>> [tx yx]=solve(S,'5*x-y=13*t',y,t)
tx =
x/2
yx =
-(3*x)/2
```

The variables for which the system is solved (y and t) are entered.

The solutions for the variables for which the system is solved are assigned in alphabetical order. The first cell has the solution for t, and the second cell has the solution for y.

When a structure is used in the output of the `solve` command, the command has the form (in the case of a system of three equations)

```
AN = solve(eq1,eq2,eq3)
```

- AN is the name of the structure.
- Once the command is executed the solution is assigned to AN. MATLAB displays the name of the structure and the names of the fields of the structure, which are the names of the variables for which the equations are solved. The size and the type of each field is displayed next to the field name. The content of each field, which is the solution for the variable, is not displayed.
- To display the content of a field (the solution for the variable), the user has to type the address of the field. The form for typing the address is: `structure_name.field_name` (see example below).

As an illustration the system of equations solved in the last example is solved again using a structure for the output.

```
>> syms x y t
>> S=10*x+12*y+16*t;
>> AN=solve(S,'5*x-y=13*t')
AN =
   x: [1x1 sym]
   y: [1x1 sym]
>> AN.x
ans =
2*t
>> AN.y
ans =
-3*t
```

Use the `solve` command to solve the system: $10x + 12y + 16t = 0$
$5x - y = 13t$

MATLAB displays the name of the structure AN and the names of its fields x and y (size and type), which are the names of the variables for which the equations are solved.

Type the address of the field x.

The content of the field (the solution for x) is displayed.

Type the address of the field y.

The content of the field (the solution for y) is displayed.

برای مشتق گیری نمادی از این دستور استفاده کنید

`diff(S)` یا `diff(S,var)`

- S نام یک عبارت نمادی موجود است یا می توان به جای آن یک عبارت نوشت
- در `diff(S)`
  - اگر فقط یک متغیر در S باشد، دستور مشتق گیری را بر حسب آن متغیر انجام می دهد
  - اگر چند متغیر در S باشد، دستور مشتق گیری را بر حسب اولین متغیر در ترتیب پیشفرض انجام می دهد
- `diff(S,var)` مشتق گیری را بر حسب `var` انجام می دهد

برای به دست آوردن مشتق nام از این دستور استفاده کنید

diff(S,n) یا diff(S,var,n)

- مثلاً diff(S,2) مشتق دوم را می دهد و diff(S,3) مشتق سوم و ...

`>> syms x y t`	Define x, y, and t as symbolic variables.
`>> S=exp(x^4);`	Assign to S the expression e^{x^4}.
`>> diff(S)`	Use the `diff(S)` command to differentiate S.
`ans =` `4*x^3*exp(x^4)`	The answer $4x^3e^{x^4}$ is displayed.
`>> diff((1-4*x)^3)`	Use the `diff(S)` command to differentiate $(1-4x)^3$.
`ans =` `-12*(1-4*x)^2`	The answer $-12(1-4x)^3$ is displayed.
`>> R=5*y^2*cos(3*t);`	Assign to R the expression $5y^2\cos(3t)$.
`>> diff(R)`	Use the `diff(R)` command to differentiate R.
`ans =` `10*y*cos(3*t)`	MATLAB differentiates R with respect to y (default symbolic variable); the answer $10y\cos(3t)$ is displayed.
`>> diff(R,t)`	Use the `diff(R,t)` command to differentiate R w.r.t. t.
`ans =` `-15*y^2*sin(3*t)`	The answer $-15y^2\sin(3t)$ is displayed.
`>> diff(S,2)`	Use `diff(S,2)` command to obtain the second derivative of S.
`ans =` `12*x^2*exp(x^4)+16*x^6*exp(x^4)`	The answer $12x^2e^{x^4}+16x^6e^{x^4}$ is displayed.

از `int(S)` یا `int(S,var)` برای محاسبه یک انتگرال نمادی معین یا نامعین استفاده کنید

- Either `S` can be the name of a previously created symbolic expression, or an expression can be typed in for `S`.
- In the `int(S)` command, if the expression contains one symbolic variable, the integration is carried out with respect to that variable. If the expression contains more than one variable, the integration is carried out with respect to the default symbolic variable (Section 11.1.3).
- In the `int(S,var)` command, which is used for integration of expressions with several symbolic variables, the integration is carried out with respect to the variable `var`.

مثال

```
>> syms x y t
>> S=2*cos(x)-6*x;
>> int(S)
ans =
2*sin(x)-3*x^2
>> int(x*sin(x))
ans =
sin(x)-x*cos(x)
>>R=5*y^2*cos(4*t);
>> int(R)
ans =
(5*y^3*cos(4*t))/3
>> int(R,t)
ans =
(5*y^2*sin(4*t))/4
```

- Define x, y, and t as symbolic variables.
- Assign to S the expression $2\cos x - 6x$.
- Use the int(S) command to integrate S.
- The answer $2\sin x - 3x^2$ is displayed.
- Use the int(S) command to integrate $x\sin(x)$.
- The answer $\sin x - x\cos x$ is displayed.
- Assign to R the expression $5y^2\cos(4t)$.
- Use the int(R) command to integrate R.
- MATLAB integrates R with respect to y (default symbolic variable); the answer $5y^3\cos(4t)/3$ is displayed.
- Use the int(R,t) command to integrate R w.r.t. t.
- The answer $5y^2\cos(4t)/4$ is displayed.

For definite integration the form of the command is:

`int(S,a,b)` or `int(S,var,a,b)`

- `a` and `b` are the limits of integration. The limits can be numbers or symbolic variables. $(\sin y - 5y^2)dy$

For example, determination of the definite integral $\int_0^{\pi}(\sin y - 5y^2)dy$ with MATLAB is:

```
>> syms y
>> int(sin(y)-5*y^2,0,pi)
ans =
2 - (5*pi^3)/3
```

- It is possible also to use the `int` command by typing the expression to be integrated as a string without having the variables in the expression first created as symbolic objects. However, the variables in the integrated expression do not exist as independent symbolic objects.
- Integration can sometimes be a difficult task. A closed-form answer may not exist, or if it exists, MATLAB might not be able to find it. When that happens MATLAB returns `int(S)` and the message `Explicit integral could not be found.`

یک معادله دیفرانسیل معمولی مرتبه اول شامل مشتق اول متغیر وابسته است

- مثلاً $\frac{dy}{dt} = f(t,y)$ که در آن t متغیر مستقل و y=$y(t)$ متغیر وابسته است

یک معادله دیفرانسیل معمولی مرتبه دوم شامل مشتق دوم (و احیاناً اول) متغیر وابسته است، یعنی $\frac{d^2y}{dt^2} = f(t,y,\frac{dy}{dt})$

جواب یک تابع به صورت $y(t)$ است که در معادله دیفرانسیل صدق کند

- یک *جواب عمومی* دارای ضرایب نامشخص است
- در یک *جواب اختصاصی* مقادیر ضرایب با استفاده از شرایط اولیه تعیین می شوند

از `dsolve` برای به دست آوردن جواب کلی یا معین معادلات دیفرانسیل استفاده می شود

جواب عمومی:

For obtaining a general solution, the `dsolve` command has the form:

`dsolve('eq')` or `dsolve('eq','var')`

- `eq` is the equation to be solved. It has to be typed as a string (even if the variables are symbolic objects).
- The variables in the equation don't have to first be created as symbolic objects. (If they have not been created, then, in the solution the variables will not be symbolic objects.)
- Any letter (lowercase or uppercase), except `D` can be used for the dependent variable.
- In the `dsolve('eq')` command the independent variable is assumed by MATLAB to be `t` (default).
- In the `dsolve('eq','var')` command the user defines the independent variable by typing it for `var` (as a string).
- In specifying the equation the letter `D` denotes differentiation. If y is the dependent variable and t is the independent variable, `Dy` stands for $\frac{dy}{dt}$. For example, the equation $\frac{dy}{dt} + 3y = 100$ is typed in as `'Dy + 3*y = 100'`.
- A second derivative is typed as `D2`, third derivative as `D3`, and so on. For example, the equation $\frac{d^2y}{dt^2} + 3\frac{dy}{dt} + 5y = \sin(t)$ is typed in as: `'D2y + 3*Dy + 5*y = sin(t)'`.
- The variables in the ODE equation that is typed in the `dsolve` command do not have to be previously created symbolic variables.
- In the solution MATLAB uses `C1`, `C2`, `C3`, and so on, for the constants of integration.

For example, a general solution of the first-order ODE $\frac{dy}{dt} = 4t + 2y$ is obtained by:

```
>> dsolve('Dy=4*t+2*y')

ans =
C1*exp(2*t) - 2*t - 1
```

The answer $y = C_1 e^{2t} - 2t - 1$ is displayed.

A general solution of the second-order ODE $\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + x = 0$ is obtained by:

```
>> dsolve('D2x+2*Dx+x=0')

ans =
C1/exp(t)+(C2*t)/exp(t)
```

The answer $x = C_1 e^{-t} + C_2 t e^{-t}$ is displayed.

The following examples illustrate the solution of differential equations that contain symbolic variables in addition to the independent and dependent variables.

The following examples illustrate the solution of differential equations that contain symbolic variables in addition to the independent and dependent variables.

```
>> dsolve('Ds=a*x^2')

ans =
a*t*x^2 + C1
```

The independent variable is t (default). MATLAB solves the equation $\frac{ds}{dt} = ax^2$.

The solution $s = ax^2t + C_1$ is displayed.

```
>> dsolve('Ds=a*x^2','x')

ans =
(a*x^3)/3 + C1
```

The independent variable is defined to be x. MATLAB solves the equation $\frac{ds}{dx} = ax^2$.

The solution $s = \frac{1}{3}ax^3 + C_1$ is displayed.

```
>> dsolve('Ds=a*x^2','a')

ans =
(a^2*x^2)/2 + C2
```

The independent variable is defined to be a. MATLAB solves the equation $\frac{ds}{da} = ax^2$.

The solution $s = \frac{1}{2}a^2x^2 + C_1$ is displayed.

جواب اختصاصی:

A particular solution of an ODE can be obtained if boundary (or initial) conditions are specified. A first-order equation requires one condition, a second-order equation requires two conditions, and so on. For obtaining a particular solution, the `dsolve` command has the form

First-order ODE: `dsolve('eq','cond1','var')`

Higher-order ODE: `dsolve('eq','cond1','cond2',....,'var')`

- For solving equations of higher order, additional boundary conditions have to be entered in the command. If the number of conditions is less than the order of the equation, MATLAB returns a solution that includes constants of integration (`C1`, `C2`, `C3`, and so on).
- The boundary conditions are typed in as strings in the following:

Math form	MATLAB form
$y(a) = A$	`'y(a)=A'`
$y'(a) = A$	`'Dy(a)=A'`
$y''(a) = A$	`'D2y(a)=A'`

- The argument `'var'` is optional and is used to define the independent variable in the equation. If none is entered, the default is t.

For example, the first-order ODE $\frac{dy}{dt} + 4y = 60$, with the initial condition $y(0) = 5$ is solved with MATLAB by:

```
>> dsolve('Dy+4*y=60','y(0)=5')
ans =
15 - 10/exp(4*t)
```

The answer $y = 15 - 10/e^{4t}$ is displayed.

The second-order ODE $\frac{d^2y}{dt^2} - 2\frac{dy}{dt} + 2y = 0$, $y(0) = 1$, $\left.\frac{dy}{dt}\right|_{t=0} = 0$, can be solved with MATLAB by:

```
>> dsolve('D2y-2*Dy+2*y=0','y(0)=1','Dy(0)=0')

ans =
exp(t)*cos(t)-exp(t)*sin(t)
```

The answer $y = e^t \cos t - e^t \sin t$ is displayed.

```
>> factor(ans)
```

The answer can be simplified with the `factor` command.

```
ans =
exp(t)*(cos(t)-sin(t))
```

The simplified answer $y = e^t(\cos t - \sin t)$ is displayed.

Additional examples of solving differential equations are shown in Sample Problem 11-5.

If MATLAB cannot find a solution, it returns an empty symbolic object and the message `Warning: explicit solution could not be found.`

عبارات نمادی را می توان به سادگی با استفاده از `ezplot` رسم کرد

- اگر عبارت نمادی S شامل یک متغیر نمادی `var` باشد، این دستور `S(var)` را در برابر `var` رسم می کند
- اگر شامل دو متغیر نمادی باشد، این دستور `S(var1,var2)=0` را در صفحه رسم میکند

11.7 PLOTTING SYMBOLIC EXPRESSIONS

To plot a symbolic expression `S` that contains one or two variables, the `ezplot` command is:

Domain of independent variable.

Domain of dependent variable.

```
ezplot(S)
```

or

```
ezplot(S,[min,max])
```

or

```
ezplot(S,[xmin,xmax,ymin,ymax])
```

- `S` is the symbolic expression to be plotted. It can be the name of a previously created symbolic expression, or an expression can be typed in for `S`.
- It is also possible to type the expression to be plotted as a string without having the variables in the expression first created as symbolic objects.
- If `S` has one symbolic variable, a plot of $S(var)$ versus (var) is created, with the values of *var* (the independent variable) on the abscissa (horizontal axis), and the values of $S(var)$ on the ordinate (vertical axis).

- If the symbolic expression `S` has two symbolic variables, `var1` and `var2`, the expression is assumed to be a function with the form $S(var1,var2) = 0$. MATLAB creates a plot of one variable versus the other variable. The variable that is first in alphabetic order is taken to be the independent variable. For example, if the variables in S are x and y, then x is the independent variable and is plotted on the abscissa and y is the dependent variable plotted on the ordinate. If the variables in S are u and v, then u is the independent variable and v is the dependent variable.

- In the `ezplot(S)` command, if `S` has one variable ($S(var)$), the plot is over the domain $-2\pi \le var \le 2\pi$ (default domain) and the range is selected by MATLAB. If `S` has two variables ($S(var1,var2)$), the plot is over $-2\pi \le var1 \le 2\pi$ and $-2\pi \le var2 \le 2\pi$.

- In the `ezplot(S,[min,max])` command the domain for the independent variable is defined by `min` and `max`:—$min \le var \le max$ —and the range is selected by MATLAB.

- In the `ezplot(S,[xmin,xmax,ymin,ymax])` command the domain for the independent variable is defined by `xmin` and `xmax`, and the domain of the dependent variable is defined by `ymin` and `ymax`.

از `ezplot` می توان برای رسم یک منحنی دو بعدی نمادی استفاده کرد، مانند $x=x(t)$ و $y=y(t)$

```
ezplot(S1,S2)
```

or

```
ezplot(S1,S2,[min,max])
```

Domain of independent parameter.

- `S1` and `S2` are symbolic expressions containing the same single symbolic variable, which is the independent parameter. `S1` and `S2` can be the names of previously created symbolic expressions, or expressions can be typed in.
- The command creates a plot of $S2(var)$ versus $S1(var)$. The symbolic expression that is typed first in the command (`S1` in the definition above) is used for the horizontal axis, and the expression that is typed second (`S2` in the definition above) is used for the vertical axis.
- In the `ezplot(S1,S2)` command the domain of the independent variable is $0 < var < 2\pi$ (default domain).
- In the `ezplot(S1,S2,[min,max])` command the domain for the independent variable is defined by `min` and `max`: $min < var < max$.

نکات تکمیلی:

زمانی که یک ترسیم با ezplot ایجاد کنید، میتوانید آن را مشابه ترسیمات ایجاد شده با plot تنظیم کنید

- با رسم یک عبارت نمادی با ezplot عبارت به صورت خودکار در بالای ترسیم نشان داده می شود

جدول ۱-۱۱ کتاب مثالهای مختلفی از ترسیم عبارات نمادی را نشان می دهد

پس از انجام عملیات با عبارت نمادی، اغلب نیاز به بدست آوردن مقدار عددی آن است. این کار را می توان با دستور subs انجام داد که مقادیر عددی را به جای متغیرهای نمادی جایگذاری می کند

جایگذاری یک مقدار عددی به جای متغیر نمادی :

```
R = subs(S,var,number)
```

The name of the symbolic expression.

The variable for which a numerical value is substituted.

The numerical value (or values) assigned to `var`.

- `number` can be one number (a scalar), or an array with many elements (a vector or a matrix).
- The value of `S` is calculated for each value of `number` and the result is assigned to `R`, which will have the same size as `number` (scalar, vector, or matrix).
- If `S` has one variable, the output `R` is numerical. If `S` has several variables and a numerical value is substituted for only one of them, the output `R` is a symbolic expression.

An example with an expression that includes one symbolic variable is:

```
>> syms x
>> S=0.8*x^3+4*exp(0.5*x)
S =
4*exp(x/2) + (4*x^3)/5
>> SD=diff(S)
SD =
2*exp(x/2)+(12*x^2)/5)
>> subs(SD, x, 2)
ans =
   15.0366
>> SDU=subs(SD, x, [2:0.5:4])

SDU =
   15.0366   21.9807   30.5634   40.9092   53.1781
```

Define x as a symbolic variable.

Assign to S the expression $0.8x^3 + 4e^{(0.5x)}$.

Use the diff(S) command to differentiate S.

The answer $2e^{x/2} + 12x^2/5$ is assigned to SD.

Use the subs command to substitute $x = 2$ in SD.

The value of SD is displayed.

Use the subs command to substitute $x = [2, 2.5, 3, 3.5, 4]$ (vector) in SD.

The values of SD (assigned to SDU) for each value of x are displayed in a vector.

In the last example, notice that when the numerical value of the symbolic expression is calculated, the answer is numerical (the display is indented). An example of substituting numerical values for one symbolic variable in an expression that has several symbolic variables is:

```
>> syms a g t v
>> Y=v^2*exp(a*t)/g
Y =
v^2*exp(a*t)/g
>> subs(Y,t,2)
ans =
v^2*exp(2*a)/g
>> Yt=subs(Y,t,[2:4])

Yt =
[ v^2*exp(2*a)/g, v^2*exp(3*a)/g, v^2*exp(4*a)/g]
```

Define a, g, t, and v as symbolic variables.

Create the symbolic expression $v^2 e^{at} / g$ and assign it to Y.

Use the subs command to substitute t = 2 in SD.

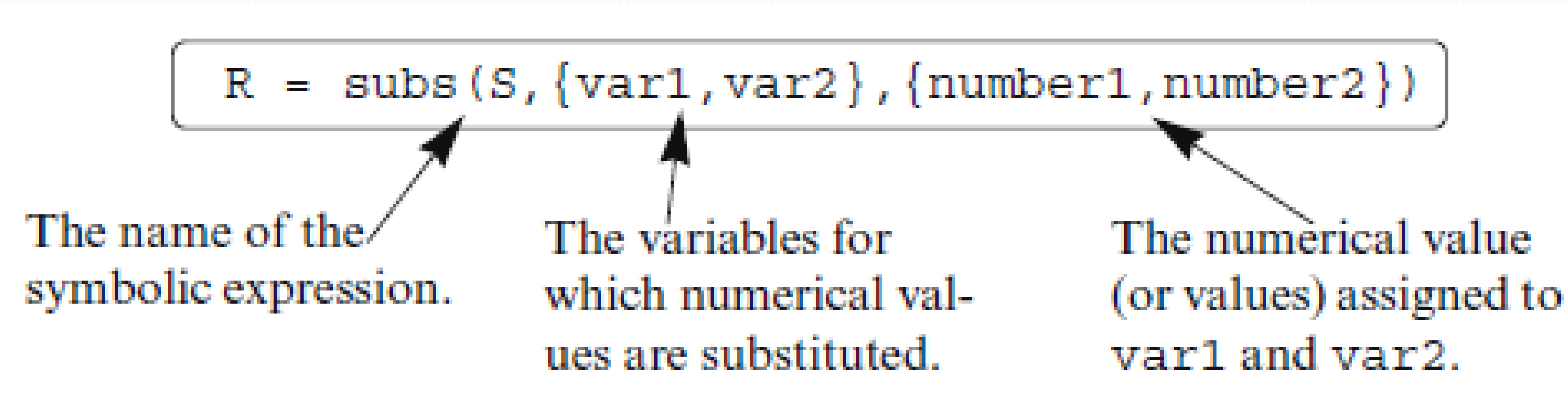
The answer $v^2 e^{2a} / g$ is displayed.

Use the subs command to substitute t = [2, 3, 4] (vector) in Y.

The answer is a vector with elements of symbolic expressions for each value of t.

جایگذاری مقدار عددی به جای دو یا چند متغیر نمادی :

از subs می توان برای جایگذاری دو یا چند متغیر نمادی استفاده کرد. مثلاً، برای دو متغیر نمادی



- The variables `var1` and `var2` are the variables in the expression `S` for which the numerical values are substituted. The variables are typed as a cell array (inside curly braces { }). A cell array is an array of cells where each cell can be an array of numbers or text.
- The numbers `number1,number2` substituted for the variables are also typed as a cell array (inside curly braces { }). The numbers can be scalars, vectors, or matrices. The first cell in the numbers cell array (`number1`) is substituted for the variable that is in the first cell of the variable cell array (`var1`), and so on.
- If all the numbers that are substituted for variables are scalars, the outcome will be one number or one expression (if some of the variables are still symbolic).
- If, for at least one variable, the substituted numbers are an array, the mathematical operations are executed element-by-element and the outcome is an array of numbers or expressions. It should be emphasized that the calculations are performed element-by-element even though the expression `S` is not typed in the element-by-element notation. This also means that all the arrays substituted for different variables must be of the same size.
- It is possible to substitute arrays (of the same size) for some of the variables and scalars for other variables. In this case, in order to carry out element-by-element operations, MATLAB expands the scalars (array of 1s times the scalar) to produce an array result.

```
>> syms a b c e x
>> S=a*x^e+b*x+c
S =
a*x^e+b*x+c
>> subs(S,{a,b,c,e,x},{5,4,-20,2,3})
ans =
   37
>> T=subs(S,{a,b,c},{6,5,7})
T =
5*x+ 6*x^e+7
>> R=subs(S,{b,c,e},{[2 4 6],9,[1 3 5]})
R =
[  2*x+a*x+9,  a*x^3+4*x+9,  a*x^5+6*x+9]
>> W=subs(S,{a,b,c,e,x},{[4 2 0],[2 4 6],[2 2 2],[1 3 5],[3 2 1]})
W =
    20      26       8
```

Define a, b, c, e, and x as symbolic variables.

Create the symbolic expression $ax^e + bx + c$ and assigned it to S.

Substitute in S scalars for all the symbolic variables.

Cell array. Cell array.

The value of S is displayed.

Substitute in S scalars for the symbolic variables a, b, and c.

The result is an expression with the variables x and

Substitute in S a scalar for c, and vectors for b and e.

The result is a vector of symbolic expressions.

Substitute in S vectors for all the variables.

The result is a vector of numerical values.

همچنین می توان مقادیر عددی را در یک عبارت نمادی با ابتدا نسبت دادن مقدار عددی به متغیر نمادی و سپس استفاده از subs جایگذاری کرد

R = subs(S)

- دقت کنید زمانی که متغیر نمادی را به صورت متغیر عددی بازتعریف می کنید، نمی توانید دیگر از آن به عنوان متغیر نمادی استفاده کنید

```
>> syms A c m x y
```
Define A, c, m, x, and y as symbolic variables

```
>> S=A*cos(m*x)+c*y
S =
c*y+A*cos(m*x)
```
Create the symbolic expression $A\cos(mx) + cy$ and assign it to S.

```
>> A=10; m=0.5; c=3;
```
Assign numerical values to variables A, m, and c.

```
>> subs(S)
```
Use the subs command with the expression S.

```
ans =
3*y + 10*cos(x/2)
```
The numerical values of variables A, m, and c are substituted in S.

```
>> x=linspace(0,2*pi,4);
```
Assign numerical values (vector) to variable x.

```
>> T = subs(S)
```
Use the subs command with the expression S.

```
T =
[ 3*y+10, 3*y+5, 3*y-5, 3*y-10]
```
The numerical values of variables A, m, c, and x are substituted. The result is a vector of symbolic expressions.

شماره تمرین های منتخب

- ۶
- ۷
- ۸
- ۹
- ۱۰
- ۱۱
- ۱۲
- ۱۳
- ۱۴
- ۱۵
- ۱۷
- ۱۸
- ۱۹
- ۲۰
- ۲۱
- ۲۴
- ۲۵
- ۲۶
- ۲۸
- ۲۹
- ۳۱