

ВЪВЕДЕНИЕ В СИСТЕМА MATHEMATICA

(с примери от физиката)

Преподавател: гл. ас. г-р Иван Богуров

1. Интерфейс на система Mathematica

Работните документи на система математика се наричат „тетражки,, (notebooks) и се състоят от двойки „входни,, и „изходни,, клетки. Една клетка е мястото, в което можем да въвеждаме команди на система Mathematica или се получават резултатите от обработката на командата.

Отворете програма Mathematica на вашия компютър. Ще видите празен документ без никакви клетки. Кликнете с мишката някъде на екрана и ще отворите клетка. След това вече можете да въвеждате команди. Въведете например $2 + 2$:

```
In[1]:= 2+2
```

Натиснете клавиша **Enter** от цифровата клавиатура или комбинацията от клавишите **Shift + Enter** на основната клавиатура. Натискането на тези клавиши означава, че искате системата да извърши обработка на израза $2 + 2$. Уверете се че курсорът на мишката се намира в клетката, която искате да обработите. В противен случай нищо няма да се случи. Ще забележите, че в дясната част на клетката се появява квадратна скоба `]`. Тази скоба определя обхвата на клетката, в която сте въвели информация. След известно време ще се появи нова клетка съдържаща резултата от обработката.

```
Out[1] = 4
```

Забележете, че входните и изходните клетки са с надпис „**In[1]**” и „**Out[1]**”. Всяка клетка е заградена от скоба, а двете заедно са обхванати от обща скоба.

Ще ви направи впечатление, че отнема доста дълго време на системата да обработи тази проста операция. Това е така, защото Mathematica натоварва компютъра ви на два етапа. Програмата се състои от две части: Потребителски интерфейс и ядро (kernel). При първоначалното стартиране на системата се зарежда само потребителския интерфейс. Частта от програмата, която всъщност прави изчисленията не се зарежда, докато за първи път не натиснете **Enter** (или **Shift + Enter**).

Mathematica признава основните оператори +, -, *, / и ^ за степен. Също така разбира и основните константи като например: π (въвежда се като Pi), експонента e (въвежда се като Exp или E), безкрайност ∞ (въвежда се като Infinity) и имагинерната единица $i = \sqrt{-1}$ (въвежда се като I). Системата признава и огромен брой от математическите функции като например: Sin[x], Cos[x], ArcSin[x], Log[x], Factorial[n] (или по-просто като n!) и много много други. Mathematica знае стойностите на тези функции за определени стойности на аргументите. Обърнете внимание, че всяка команда започва с главна буква, а аргументите на командата (функцията) са в квадратни скоби. За изпълнението на някои команди от съществено значение е версията на програмата, с която разполагате. Повече информация за версията на системата може да получите след изпълнението на командата `$Version`.

Работните клетки могат да бъдат използвани и за получаване на повече информация относно стойността на дадена константа или действието на определена команда. Това става с въвеждането на `?` знак пред константата или командата. Например:

```
In[2]:= ?Pi
```

```
Pi is  $\pi$  with numerical value ~3.14159 >>
```

Повече информация се получава при поставянето на втори въпросителен знак, т.е. `??Plot` ще ви даде повече информация за командата заедно със синтаксиса. Повече помощ и информация може да получите от менюто **Help** на програмата.

2. Прости математически изчисления

Най-тривиалното използване на системата е като обикновен калкулатор. За разлика от конвенционалните езици за програмиране Mathematica може да прави и символни обработки. В тази част ще бъдат представени някои от тези възможности. Нека изчислим стойността на израза:

$$\sqrt{\frac{(6,9 \cdot 10^{39})(7,4 \cdot 10^{-49})}{3,69762 \cdot 10^{11}}}$$

```
In[1]:= Sqrt[(6.9*10^39)*(7.4*10^-49)/(3.69762*10^11)]
```

```
Out[1]= 1.17511*10^-10
```

Обикновените скоби () се използват за групиране и указване на приоритет на действията. В нова клетка на вашия notebook въведете примера по-долу: След натиска-

нето на **Enter** получавате резултат, който е различен в зависимост от това дали използвате скоби или не.

```
In[1]:= (1+2)*3
Out[1] = 9
In[1]:= 1+2*3
Out[1] = 7
```

В определени случаи можем и да пропуснем използването на * като знак за умножение. Системата е достатъчно „умна“, да се досети за правилното действие.

```
In[2]:= (1+2) 3
Out[2] = 9
```

Но Mathematica може да прави много повече от просто събиране и умножение на числа. Опитайте:

```
In[3] = 102/9
```

Ще получите резултат:

```
Out[3] = 34/3
```

Забелязвате, че Mathematica прави опростяване на дробта до най-обикновена форма до достигане на най-малка стойност на числителя и знаменателя. Също така забелязвате, че резултатът все още се извежда като дроб, не като десетично число. Това е така, защото това е най-точната стойност, която може да бъде получена. При всичките си действия системата се старее да запази възможно най-голяма точност, освен ако не поискате друго от нея. Това може да доведе до известни обърквания, че системата не може да пресметне исканата от вас стойност. Например:

```
In[2]:= Sqrt[4]
Out[2] = 2
In[2]:= Sqrt[5]
Out[2] =  $\sqrt{5}$ 
```

След като не съществува по-точна стойност от $\sqrt{5}$ Mathematica оставя тази команда необработена. Ако аргумента на командата се зададе като десетично число само с поставянето на десетична точка системата го припознава като приближена стойност и извежда резултат с определена точност. По подразбиране програмата работи с точност пет знака след десетичната запетая, но това може да бъде променяно по всяко време.

```
In[2]:= Sqrt[5.]  
Out[2] = 2.23607
```

Използването на знакът = след x означава, че задаваме постоянна стойност на аргумента.

```
In[4] = x=7  
Out[4] = 7
```

Тази стойност Mathematica ще помни до затварянето на програмата и можем да я използваме за всички последващи операции с използването на x .

```
In[5] = x+3  
Out[5] = 10
```

По същия начин можете да задавате постоянни стойности и на групи аргументи в рамките на една сесия.

```
In[6]:= y=20  
Out[6] = 20  
In[7]:= x*y  
Out[7] = 140  
In[8]:= (x+3)*y  
Out[8] = 200  
In[8]:= 2^x  
Out[8] = 128
```

В някои случаи задаването на постоянна стойност на аргумента може да доведе до грешки при изпълнението на определени команди. Например интегриране:

```
In[9]:= Integrate[x^2,x]
```

Вместо очаквания резултат $x^3/3$ получаваме:

```
Out[9] = Integrate[49,7]
```

Когато повече не Ви е необходима въведената стойност за x може да се освободите от нея при изпълнението на командата `Clear[x]` или `Clear[y]` и т. н. Ако искате системата да забрави всички стойности въведени по време на сесията командата придобива вида `Clear["Global'*"]`. Понякога се препоръчва да изпълните тази команда в началото на всеки notebook с цел предотвратяване на последващи конфликти.

След изпълняване на командата `Clear`, ако се върнете в клетката и обработите отново командата за интегриране ще получите правилния резултат.

В началото обърнахме внимание, че работните клетки си имат име и номер в квадратни скоби. Този номер може да използвате за да направите препратка към съответната клетка с цел резултата от обработката на тази клетка да бъде използван при операция в друга клетка. Например:

```
In[1]:= 5
Out[1]= 5
In[2]:= %1+15
Out[2]= 20
In[3]:= %*100
Out[8]= 2000
In[8]:= %%*2
Out[8]= 40
```

Използването на % без номер на клетка означава системата да използва стойността в предишната клетка, а %% - стойността в по-предишната клетка.

3. Прецизност

Както стана ясно по-горе система Mathematica при изпълнение на командите се старее да запази максимално голяма точност на резултата. Ако желаете да получите числена стойност използвайте командата $N[x]$, която извежда числената стойност на x . По подразбиране числената стойност се извежда с точност до петия знак след десетичната запетая (точка). По-висока точност може да бъде получена при добавяне на допълнителен атрибут на командата $N[x,k]$, която ще изведе числената стойност на резултата x с точност k знака след десетичната запетая.

```
In[1]:= Sqrt[5]
Out[1]=  $\sqrt{5}$ 
In[2]:= N[%1]
Out[2]= 2.23607
In[3]:= N[%1,20]
Out[3]=
In[4]:= %%*2
Out[4]= 40
```

4. Дефиниране на функции

Mathematica разполага с огромен брой дефинирани функции, но много често се налага да се дефинира нова функция от страна на потребителя. Тази възможност може да бъде илюстрирана от следния пример, в който се дефинира функцията $f(x) = x + x^2$.

```
In[1]:= f[x_]:=x+x^2
```

```
In[2]:= f[3]
```

```
Out[2]= 12
```

```
In[3]:= f[t]
```

```
Out[3]= t+t^2
```

Обърнете внимание на две неща в рег In[1]:. Първо на долната черта „_“, след x , която се нарича „празна променлива“, което означава, че когато бъде направена заявка към функцията, аргументът може да бъде всякакъв, не е задължително да бъде x . Второ, използването на $:=$, вместо $=$, което означава, че обработката на функцията се отлага докато не бъде направена заявка към нея със съответна стойност на аргумента. Разликите между използването на $:=$ и $=$ могат да бъдат илюстрирани със следните примери.

```
In[10]:= k=10
```

```
Out[10]= 10
```

```
In[11]:= f[x_]:=x+k*x^2
```

```
Out[11]= x+10*x^2
```

```
In[12]:= g[x_]:=x+k*x^2
```

```
In[13]:= g[2]
```

```
Out[13]= 42
```

```
In[13]:= g[5]
```

```
Out[13]= 255
```

В примера по-горе са дефинирани две функции $f[x_]$ и $g[x_]$, в които участва променливата k , за която преди това е присвоена постоянна стойност. Въпреки, че сменяте стойността на променливата x за функция $g[x_]$, стойността на k се запазва. Освен това функцията $f[x_]$, в която също участва x остава необработена защото не сте направили заявка към нея. Трябва да се внимава при поставянето на долна черта след променливите, защото това може да доведе до проблеми.

5. Решаване на уравнения

Командата `Solve[]` се използва за аналитично решаване на уравнения. Въпреки, че в посочените примери ще бъдат представени възможностите за решаване на полиноми на практика чрез тази команда могат да бъдат решавани всякакви уравнения. Командата `Solve` се използва във вида `Solve[уравнение, променлива]`. Нека намерим решението на уравнението $x^2 - 3x - 10 = 0$.

```
In[10]:= reshenie=Solve[x^2-3*x-10==0,x]
Out[10]= {{x->-2},{x->5}}
```

В примера по-горе е показано още едно използване на знака `=`. Чрез използване на един знак `=` се присвоява име на решението на полинома. При въвеждане на уравнение знакът `=` се удвоява задължително. Решението на уравнението се извежда от системата във вид на трансформационно правило.

`Solve` може да бъде използван и за решаване на системи уравнения или на няколко уравнения едновременно. В този случай уравненията и променливите се групират в фигурални скоби `{}`. Нека намерим решението на следната система:

$$\begin{cases} x + y = 10 \\ 3x - 5y = 38 \end{cases}$$

```
In[11]:= Solve[{x+y==10,3*x-5*y==38},{x,y}]
Out[11]= {{x->11},{y->-1}}
```

Използването на `Solve` ви дава общото решение, което може да не е валидно във всички случаи. Например за уравнението $ax = b$ общото решение, което ще получите ще е $x = b/a$, но то не е валидно в случая когато $a = 0$. В този случай от полза е използването на командата `Reduce`, която ще покаже условията, при които намереното решение е валидно. Сравнете

```
In[12]:= Solve[a*x==b,x]
Out[12]= {x->b/a}
In[12]:= Reduce[a*x==b,x]
Out[12]= a==0 && b==0 || x = b/a && a!=0
```

където логическото условие „или„ е записано като `||`, логическото „и„ като `&&` и „различно„ или „не е равно на„ като `!=`.

Численото решение на уравненията се получава чрез прилагане на командата NSolve, като атрибутите на командата са същите като при Solve. Например:

```
In[11]:= NSolve[x^3+5*x^2+5*x+1==0,x]
Out[11]= {{x->-3.73205},{x->-1.},{x->-0.267949}}
```

В този пример трябва да се обърне внимание, че получените решения се извеждат с различна точност достатъчна за да удовлетвори уравнението. Отново решенията се представят във вид на трансформационни правила, т.е. решенията трябва да бъдат прочетени като „x клони към,, или „x е заместено с,,. Трансформационните правила са важни и те се използват често при работа със система Mathematica. Тези правила за трансформация могат да бъдат използвани заедно с оператор за заместване „/.“ при пресмятане на стойност на израз, когато променливата бъде заместена с определена стойност. Например:

```
In[12]:= p^2+2p-79 /. p -> 10
Out[11]= 41
```

Ако трябва да бъде направено заместване с няколко стойности те отново се обединяват с използването на фигурални скоби т. е.

```
In[12]:= m^2+2n /. {m -> 1,n->2}
Out[11]= 5
```