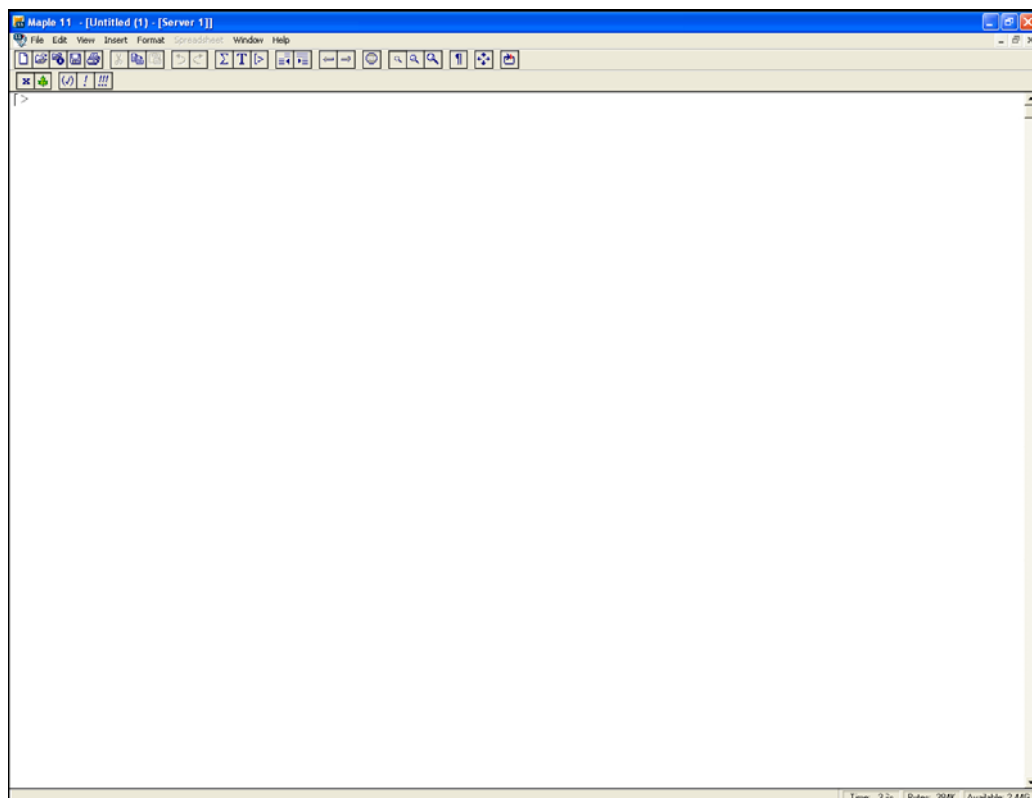


Maple: Въведение

Основи на Maple

Maple е ефективна компютърна програма за изпълняване на алгебрични или символни математически изчисления. С Maple могат да се извършват пресмятания по числен анализ и да се изобразяват графичните резултати по много различни начини. Освен това интерфейсът на Maple с работни листи може да се използва за създаване на документи с интегриран текст, математически символи, графики и хипервръзки. Поради тази многофункционалност Maple понякога се използва като редактор на формули, но основно повечето хора го използват заради мощните му възможности за решаване на уравнения.

Един от ключовите елементи на Maple е способността му да оперира със символни математически изрази. Често пъти, когато работим с Maple, може да отложим точното пресмятане с използване на символна променлива. Maple третира тези недефинирани елементи като символи, които по-късно могат да получат числена стойност.



Фигура 1: Типичен работен лист на Maple

Стартиране на Maple

Работният лист на Maple е съставен от три елемента: команди на Maple (в червено), резултати (в синьо), които Maple извежда за всяка команда, и съобщения за грешки. Ето един пример за команда на Maple:

```
> 5+3;
```

8

За изпълнение на команда в Maple кликваме върху нея с мишката и след това натискаме клавиша **Enter**. Да отбележим също че Maple първо извежда резултатите и тогава курсорът отива автоматично на следващата Maple команда. Всяка команда започва след символа > (“по-голямо”, който е системен промпт (подкана) на Maple), а всяка Maple команда трябва да завършва с точка и запетая или двоеточие. Ако командата завършва с точка и запетая, Maple разпечатва резултата от командата на екрана, а ако командата завършва с двоеточие - не разпечатва резултата на екрана.

```
> 4+2;
```

6

```
> 3+1:
```

Всяка команда може да съдържа коментари, които не са изпълнима част от нея. Знакът „диез” (т.е. #) като част от входния ред на Maple означава коментар, който програмата трябва да игнорира. Всичко, написано след знака „диез” се третира като коментар и не се изпълнява от Maple.

```
> 4+4; # Maple ignores what comes after the hash symbol.
```

Когато курсорът е след празен Maple промпт и натиснете **Enter**, тогава курсорът ще се премести надолу след следващия промпт. Възможно е връщането и редактирането на команди на Maple, които са вече изпълнени. Кликнете върху команда на Maple, която вече е изпълнена и използвайте стрелките и клавиша **Обратна стрелка** за да се преместите върху командата и да я промените. Ако сега натиснете **Enter**, ще се изпълни новата команда.

Ако при въвеждането на команда допуснете някаква грешка и Maple не може да разбере какво се иска да бъде изпълнено, тогава програмата дава съобщение за грешка

вместо резултат. Обикновено, честа грешка е забравянето на точка и запетая (или двоеточие) в края на командата. Ако се опитате да изпълните следната команда

```
> 5+5
```

Warning, premature end of input

Maple ще върне съобщение за грешка, защото е забравена точка и запетая. Забелязваме това и се връщаме на командата, добавяме точка и запетая накрая и отново изпълняваме командата. Има и много други видове грешки, които се допускат в Maple командите. Едно то най-трудните неща при работа с която и да е компютърна програма е да се справяте с грешките и неясните съобщения за грешки, които следват от тях. Изисква се практика и опит, за да се научите как да се справяте с повечето от съобщенията за грешки на Maple. Опитвайте се да пишете командите много внимателно, за да избегнете колкото може повече правописни грешки.

Ако искате можете да напишете повече от една команда на Maple на един ред, но всяка от тях трябва да завършва със собствена точка и запетая или две точки. Следващите три команди ще се изпълнят наведнъж.

```
> 5+5; 4-1; 3+2;
```

10

3

5

Има и друг начин да се изпълняват няколко команди на Maple наведнъж. Това може да се направи чрез въвеждане на командите в изпълнима група. Необходимо е да поставите курсора някъде върху символ от изпълнимата група и да натиснете клавиша **Enter**. Тогава всички команди от групата ще се изпълнят наведнъж една след друга. Да отбележим, че свързването на няколко команди в изпълнима група става с използването на скоби отляво на промпта, (ако не виждаш скобите, използвай “View → Show Group ranges” от менюто или натиснете бутон **F9** от клавиатурата).

```
> 5+5;  
> 4-1;  
> 3+2;
```

10

3

5

Накрая ще кажем че можете да добавяте допълнителни Maple промптове към работния лист като поставяте курсора на който и да е ред и натиснете клавишната комбинация **Ctrl-J**. Появява се нов промпт на реда, в който сте в момента. Например, ако искате да добавите нов ред в самото начало на работния лист, тогава поставете курсора на реда, където искате да е промпта и натиснете **Ctrl-K**.

Maple като калкулатор

Maple може да се използва подобно на обикновен калкулатор, но той може да изпълнява много повече изчисления. Най-добре е всяка Maple сесия да започва с рестарт, което означава да бъдат изтрети всички предишни резултати в паметта на Maple (това е важно за да избегнем възможното объркване, ако вече са изпълнени някакви команди в този работен лист или в някой предишен).

```
> restart;
```

Maple може да прави две неща – “символно пресмятане” и “числено пресмятане”. Ето един пример за символно пресмятане:

```
> 1/3 + 1/3;
```

$\frac{2}{3}$

Ще отбележим, че при изпълнението на тази команда, Maple събира дробите символно и извежда $\frac{2}{3}$. Ако сложим десетична точка в командата, тогава Maple ще изпълни операцията числено, като калкулатор.

```
> 1.0/3 + 1/3; # Notice the decimal point.
```

0.6666666666

Това показва, че в Maple има разлика между символно и числено пресмятане. Понякога казваме, че символното пресмятане е “точно”, а численото е “приближено”.

израза $9^{\frac{1}{8}}$ на $3^{\frac{1}{4}}$. Ако въведем този израз в Maple ще видим, че той може да не направи опростяването. Понякога Maple се нуждае от изрично указване за извършване на опростяване чрез използване на командата **simplify**.

```
> 9^(1/8);  
9^(1/8)  
> simplify(%);  
3^(1/4)
```

Използването на символа **%** като част от команда, означава за Maple да замени последния получен резултат на тази позиция. В примера по-горе **%** ще се замени с $9^{\frac{1}{8}}$.

При работа с дълъг израз е нормално да използваме скоби, за да направим израза по-четлив за потребителя. Но трябва внимателно да поставяме скобите, защото тяхната позиция може да промени реда на операциите, както е показано на примера:

```
> (5-3)/(2+2*3); 5-(3/2+2*3);  
1/4  
-5/2  
> 2*4^2/(6+3)*2; (2*4)^(2)/6+3*2;  
64/9  
50/3
```

Тези примери също показват как да повдигаме израз на степен (^), да изпълним умножение (*) и деление (/).

Maple има вградени всички елементарни функции, не само тези на калкулатор, но и много други. Можем също да дефинираме и наши функции, както следва.

```
> f := t*cos(2*t);  
f := t cos(2t)
```

Сега ние може да пресметнем тази функция използвайки командата **eval** (**eval** е

съкращение на `evaluate`).

```
> eval( f, t=Pi/2 );
```

$$-\frac{\pi}{2}$$

Освен това, можем също да изпълняваме тази функция, използвайки командата `subs` (`subs` е съкращение на `substitute`).

```
> subs( t=Pi/2, f );
```

$$\frac{1}{2} \pi \cos(\pi)$$

Да отбележим, че за разлика от `eval`, `subs` не опростява резултата и затова трябва да използваме непосредствено `simplify`.

```
> simplify( % );
```

$$-\frac{\pi}{2}$$

Накрая може да получим десетичното приближение на тази стойност, използвайки `evalf`.

```
> evalf( % );
```

$$-1.570796327$$

Maple като символен калкулатор

Maple може да изпълнява символни пресмятания с изрази, съдържащи непознати променливи. Преди да покажем алгебра с Maple, ще променим `x` като непозната променлива. По-рано ние дадохме на `x` стойност; сега трябва да я нулираме. Нулирането на стойността на `x` става чрез използване на оператор за присвояване, като присвояваме “нечислово `x`” на `x`. Получаваме “нечислово `x`” като вдясно в кавички поставяме променливата `x`, както е показано:

```
> x := 'x'; # Those are both right quotes.
```

Важно е да запомним как да премахнем присвояването на променливите. Това е нещо, от което се нуждаем често, когато работим с голям брой променливи или когато въвеждаме много дълги работни листи.

Сега, когато определихме `x` като непозната променлива, може да поискаме

Maple да извърши алгебрични манипулации в равенства съдържащи x .

Командата `solve` се използва за решаване на уравнения относно някакво неизвестно. Като аргумент на командата въвеждаме уравнението за решаване и променливата, относно която трябва да се реши. Ето един лесен пример.

```
> solve (4*x + 2 = 6, x);
```

1

Maple ни показва стойността на x , която е решение на уравнението. Може да изглежда чудно защо трябва да указваме на Maple да решава спрямо x , щом x е единствената променлива в уравнението. Това е така, защото уравнението може да има няколко променливи, следователно ние винаги трябва да инструктираме Maple спрямо коя променлива да решава. Например, следващите две команди искат от Maple да реши едно и също уравнение спрямо различни променливи.

```
> solve (a*x + 2*b = 4 , x);
```

$$-\frac{2(b-2)}{a}$$

```
> solve (a*x + 2*b = 4 , b);
```

$$-\frac{ax}{2} + 2$$

Следващата команда на Maple използва оператор за присвояване, за да даде име на уравнението. Името на уравнението ще бъде `quad1`.

```
> quad1 := a*x^2+b*x+c=0;
```

$$quad1 := ax^2 + bx + c = 0$$

Можем сега да използваме присвоеното име на уравнението в други команди на Maple.

```
> solve( quad1, x );
```

$$-\frac{b - \sqrt{b^2 - 4ac}}{2a}, \frac{b + \sqrt{b^2 - 4ac}}{2a}$$

Отговорите на горната команда вече ни е познат. Ние можем също да присвояваме стойности на параметрите a , b и c .

```
> a := 1; b := 2; c := -1;
      a := 1
      b := 2
      c := -1
```

Ще решим `quad1` отново, но този път ще дадем име на решението.

```
> solution := solve( quad1, x );
      solution := -1 + √2, -1 - √2
```

Името `solution` сега се отнася до две решения! Възможно е да извличаме и използваме всяко от отделните решения както следва

```
> solution[1]; solution[2];
      -1 + √2
      -1 - √2
```

Как бихме могли да проверим, че всяко от отделните решения на уравнението удовлетворява квадратното уравнение? Ние правим това чрез заместване на всяко решение в квадратното уравнение и проверяваме дали лявата и дясната страна на уравнението са равни. Използваме командата `subs` за да заместим всяко от решенията обратно в квадратното уравнение.

```
> subs( x=solution[1], quad1 );
      (-1 + √2)2 - 3 + 2√2 = 0
```

Като използваме командата `simplify` за предишния резултат проверяваме дали `solution[1]` е решение на квадратното уравнение.

```
> simplify ( % );
      0 = 0
```

Това потвърждава че `solution[1]` е наистина решение на квадратното уравнение `quad1`. Можем аналогично да проверим че и `solution[2]` е също решение на квадратното уравнение `quad1`.