

内部数学常数

pi	圆周率
exp(1)	自然对数的底数 e
i 或 j	虚数单位
Inf 或 inf	无穷大

2 基本数学运算符

$a+b$	加法
$a-b$	减法
$a*b$	矩阵乘法
$a.*b$	数组乘法
$a/b$	矩阵右除
$a\backslash b$	矩阵左除
$a./b$	数组右除
$a.\backslash b$	数组左除
$a^b$	矩阵乘方
$a.^b$	数组乘方
$-a$	负号
,	共轭转置

.,	一般转置
----	------

### 3 关系运算符

==	等于
<	小于
>	大于
<=	小于或等于
>=	大于或等于
~=	不等于

### 4 常用内部数学函数

指数函数	$\exp(x)$	以 e 为底数
对数函数	$\log(x)$	自然对数，即以 e 为底数的对数
	$\log_{10}(x)$	常用对数，即以 10 为底数的对数
	$\log_2(x)$	以 2 为底数的 x 的对数
开方函数	$\text{sqrt}(x)$	表示 x 的算术平方根
绝对值函数	$\text{abs}(x)$	表示实数的绝对值以及复数的模
三角函数 (自变量的	$\sin(x)$	正弦函数
	$\cos(x)$	余弦函数
	$\tan(x)$	正切函数

单位为弧度)	$\cot(x)$	余切函数
	$\sec(x)$	正割函数
	$\csc(x)$	余割函数
反三角函数	$\arcsin(x)$	反正弦函数
	$\arccos(x)$	反余弦函数
	$\arctan(x)$	反正切函数
	$\operatorname{arccot}(x)$	反余切函数
	$\operatorname{arcsec}(x)$	反正割函数
	$\operatorname{arccsc}(x)$	反余割函数
双曲函数	$\sinh(x)$	双曲正弦函数
	$\cosh(x)$	双曲余弦函数
	$\tanh(x)$	双曲正切函数
	$\coth(x)$	双曲余切函数
	$\operatorname{sech}(x)$	双曲正割函数
	$\operatorname{csch}(x)$	双曲余割函数
反双曲函数	$\operatorname{arsinh}(x)$	反双曲正弦函数
	$\operatorname{arcosh}(x)$	反双曲余弦函数
	$\operatorname{artanh}(x)$	反双曲正切函数
	$\operatorname{arcoth}(x)$	反双曲余切函数
	$\operatorname{arsech}(x)$	反双曲正割函数
	$\operatorname{acsch}(x)$	反双曲余割函数
求角度函数	$\operatorname{atan2}(y, x)$	以坐标原点为顶点，x 轴正半轴

		为始边，从原点到点 (x, y) 的射线为终边的角，其单位为弧度，范围为 $(-\pi, \pi]$
数论函数	gcd(a, b)	两个整数的最大公约数
	lcm(a, b)	两个整数的最小公倍数
排列组合函数	factorial(n)	阶乘函数，表示 n 的阶乘
复数函数	real(z)	实部函数
	imag(z)	虚部函数
	abs(z)	求复数 z 的模
	angle(z)	求复数 z 的辐角，其范围是 $(-\pi, \pi]$
	conj(z)	求复数 z 的共轭复数
求整函数与截尾函数	ceil(x)	表示大于或等于实数 x 的最小整数
	floor(x)	表示小于或等于实数 x 的最大整数
	round(x)	最接近 x 的整数
最大、最小函数	max([a, b, c, . . . ])	求最大数

	$\min([a, b, c, \dots])$	求最小数
符号函数	$\text{sign}(x)$	$\text{sign}(x) = \begin{cases} 1 & , x > 0 \\ -1 & , x < 0 \\ 0 & , x = 0 \end{cases}$

## 5 自定义函数-调用时：“[返回值列]=M 文件名（参数列）”

function 返回变量=函数名（输入变量）

注释说明语句段（此部分可有可无）

函数体语句

## 6. 进行函数的复合运算

$\text{compose}(f, g)$	返
回值为 $f(g(y))$	
$\text{compose}(f, g, z)$	返回值
为 $f(g(z))$	
$\text{compose}(f, g, x, .z)$	返回值为
$f(g(z))$	
$\text{compose}(f, g, x, y, z)$	返回值为 $f(g(z))$

## 7 因式分解

`syms` 表达式中包含的变量

`factor(表达式)`

## 8 代数式展开

`syms` 表达式中包含的变量

`expand(表达式)`

## 9 合并同类项

`syms` 表达式中包含的变量

`collect(表达式, 指定的变量)`

## 10 进行数学式化简

`syms` 表达式中包含的变量

`simplify(表达式)`

## 11 进行变量替换

`syms` 表达式和代换式中包含的所有变量

`subs(表达式, 要替换的变量或式子, 代换式)`

## 12 进行数学式的转换

调用 Maple 中数学式的转换命令，调用格式如下：

`maple(‘Maple 的数学式转换命令’)` 即：

`maple(‘convert(表达式, form)’)` 将表达式转换成 form 的表示方式  
`maple(‘convert(表达式, form, x)’)` 指定变量为 x，将依赖于变量 x 的函数转换成 form 的表示方式（此指令仅对 form 为 exp 与 sincos 的转换式有用）

## 13 解方程

`solve(‘方程’，‘变元’)`

注：方程的等号用普通的等号：=

## 14 解不等式

调用 maple 中解不等式的命令即可，调用形式如下：

`maple(‘maple 中解不等式的命令’)*`

具体说，包括以下五种：

`maple(‘solve(不等式)’)`

```

maple(' solve (不等式, 变元) ' )

maple(' solve ( {不等式} , 变元) ' )

maple(' solve (不等式, {变元}) ' )

maple(' solve ( {不等式} , {变元}) ' )

```

## 15 解不等式组

调用 maple 中解不等式组的命令即可，调用形式如下：

```
maple(' maple 中解不等式组的命令')
```

```
即: maple(' solve ( {不等式组} , {变元组}) ' )
```

## 16 画图

方法 1：先产生横坐标  $x$  的取值和相应的纵坐标  $y$  的取值，然后执行命令：`plot(x, y)`

方法 2: `fplot(' f(x)', [xmin, xmax])`

```
fplot(' f(x)', [xmin, xmax, ymin, ymax])
```

方法 3: `ezplot(' f(x)')`

```
ezplot(' f(x)' , [xmin, xmax])
```

```
ezplot(' f(x)' , [xmin, xmax, ymin, ymax])
```



## 17 求极限

(1) 极限:

```
syms x  
  
limit(f(x), x, a)
```

(2) 单侧极限:

左极限:

```
syms x  
  
limit(f(x), x, a, 'left' )
```

右极限:

```
syms x  
  
limit(f(x), x, a, 'right' )
```

## 18 求导数

```
diff('f(x)')  
  
diff('f(x)', 'x')
```

或者:

```
Syms x
```

```
Diff(f(x))
```

```
syms x
```

```
diff(f(x), x)
```

## 19 求高阶导数

```
diff('f(x)', n)
```

```
diff('f(x)', 'x', n)
```

或者：

```
syms x
```

```
diff(f(x), n)
```

```
syms x
```

```
diff(f(x), x, n)
```

20 在 MATLAB 中没有直接求隐函数导数的命令，但是我们可以根据数学中求隐函数导数的方法，在中一步一步地进行推导；也可以自己编一个求隐函数导数的小程序；不过，最简便的方法是调用 Maple 中求隐函数导数的命令，调用格式如下：

```
maple('implicitdiff(f(x,y)=0,y,x)')*
```

在 MATLAB 中，没有直接求参数方程确定的函数的导数的命令，只能根据参数方程确定的函数的求导公式

$$\frac{dy}{dx} = \frac{\frac{dy}{dt}}{\frac{dx}{dt}}$$

一步一步地进行推导；或者，干脆自己编一个小程序，应用起来会更加方便。

## 21 求不定积分

```
int('f(x)')
```

```
int('f(x)','x')
```

或者：

```
syms x
```

```
int(f(x))
```

```
syms x
```

```
int(f(x), x)
```

## 22 求定积分、广义积分

```
int('f(x)', a, b)
```

```
int('f(x)', 'x', a, b)
```

或者：

```
syms x
```

```
int(f(x), a, b)
```

```
syms x
```

```
int(f(x), x, a, b)
```

## 23 进行换元积分的计算

自身没有提供这一功能，但是可以调用 Maple 函数库中的 `changevar` 命令，调用方法如下：

```
maple(' with(student)')
```

加载 student 函数库后，

才能使用 changevar 命令

```
maple(' changevar( m(x)=p(u), Int(f(x), x) ) ' )
```

把积分

表达式中的  $m(x)$  代换成  $p(u)$

## 24 进行分部积分的计算

自身没有提供这一功能，但是可以调用 Maple 函数库中的 intparts 命令，调用方法如下：

```
maple(' with(student)')
```

加载 student 函数库后，才

能使用 intparts 命令

```
maple(' intparts(Int(f(x), x), u)')
```

指定  $u$ ，用分部积分公式

$\int u dv = uv - \int v du$  进行计算

## 25 对数列和级数进行求和

```
syms n
```

```
symsum(f(n), n , a , b )
```

## 26 进行连乘

```
maple(' product(f(n), n=a..b)')
```

## 27 展开级数

```
syms x
```

```
Taylor(f(x), x, n, a)
```

## 28 进行积分变换

```
syms s t
```

```
laplace(f(t), t, s)
```

拉

普拉斯变换

```
ilaplace(F(s), s, t)
```

拉

普拉斯变换的逆变换

```
syms t ω
```

```
fourier(f(t), t, ω)
```

傅立叶变换

```
ifourier(F(ω), ω, t)
```

傅

立叶变换的逆变换

```
syms n z
```

```
ztrans( f(n), n, z) Z
```

变换

```
iztrans( F(z), z, n ) Z
```

变换的逆变换

在 matlab 中，矩形法、梯形法和辛普森法求近似积分

可以用自身的命令，也可调用 Maple 的相应命令。调用方法如下：

```
maple('with(student)')
```

```
maple('Maple 中求定积分近似值的命令')
```

## 29 解微分方程

```
Dsolve('微分方程', '自变量')
```

```
dsolve('微分方程', '初始条件或边界条件', '自变量')
```

## 30 解微分方程组

```
Dsolve('微分方程组', '自变量')
```

```
dsolve('微分方程组', '初始条件或边界条件', '自变量')
```

