

Command Line

```

)cd <pathname>
)clear all -- clear workspace
)display op <function> -- function arguments
)set message autoload off -- quietly load algebra
)set message bottom on -- show selection process
)set stream calculate 20 -- number of terms to calculate
)show <domain> -- list all functions
)spool <filename> -- start save session
)spool -- close spool file
)trace <domain> )math -- trace execution
)quit -- exit Axiom
)read <filename>[.input] -- evaluate a file
)sys <command line> -- execute command
_ continues input lines or escapes chars a_ b = "a b"
% is last value
%%(n) is nth value
-- and ++ start comment lines

```

Programming

```

assignment: var := value
            x:=3
conditional: if <pred> then <truecase> else <falsecase>
            if (2 > 4) then 4 else 5
loop: for <pred> repeat (block)
      for i in 1..5 repeat print i
      while i < 3 repeat (print i ; i:=i+1)
function: f(x) = x^2
          f(x)==x^2
anon. function: g:=x --> x+1    g(3) -> 4
Indentation is significant:
          f(x)==(x > 3 => x ; 0)
          f(x)==
              x > 3 => x
              0

```

Basic constants and functions

```

pi = %pi    e = %e    i = %i    infinity = %infinity
+infinity = %plusInfinity    -infinity = %minusInfinity
numeric(%pi) = 3.1415926535 897932385
Functions: sin cos tan sec csc cot sinh cosh tanh

```

```

sech csch coth log ln exp
ab = a*b    a/b = a/b    a^b = a^b    sqrt(x) = sqrt(x)
sqrt[n]{x} = x^(1/n)    |x| = abs(x)    log_b(x) = log(x)/log(b)

```

Operations on expressions

```

factor(...)    expand(...)    simplify(...)
Symbolic equations: f(x)=g(x)
Solve f(x) = g(x):  solve(f(x)=g(x),x)
solve([x^2*y-1,x*y^2-2],.01)
                    -> [[y = 1.5859375, x = 0.79296875]]
complexSolve([x^2*y-1,x*y^2-2],1/1000)
radicalSolve([x^2/a+a*y^3-1,a*y+a+1],[x,y])
sum_{i=k}^n f(i) = reduce(+,[f(i) for i in k..n])
prod_{i=k}^n f(i) = reduce(*,[f(i) for i in k..n])

```

Pattern Matching

```

logrule:=rule log(x)+log(y) == log(x*y) ->
              log(y)+log(x)+%B==log(x y)+%B
f:=log sin x + log x -> log(sin(x))+log(x)
logrule f -> log(x sin(x))

```

Calculus

```

lim_{x->a} f(x) = limit(f(x), x=a)
lim_{x->a^-} f(x) = limit(f(x), x=a, "left")
lim_{x->a^+} f(x) = limit(f(x), x=a, "right")
lim_{x->infinity} f(x) = limit(f(x), x=%plusInfinity)
limit(sin(x)/x,x=%plusInfinity) -> 0
complexLimit(sin(x)/x,x=%infinity) -> "failed"
d/dx (f(x)) = D(f(x),x)
d/dx (f(x,y)) = D(f(x,y),x)
int f(x)dx = integrate(f(x),x)
int_a^b f(x)dx = integrate(f(x),x=a..b)

```

Series

```

x:=series 'x
y:=sin(x) -> x - 1/6 x^3 + 1/120 x^5 - 1/5040 x^7 + O(x^9)
coefficient(y,3) -> -1/6
taylor(f(x),x=a)

```

```

laurent(x/log(x),x=1)
puiseux(sqrt(sec(x)),x=3*%pi/2)

```

2D graphics

```

draw(cos(5*t/8),t=0..16*%pi,coordinates==polar)
f(t:SF):SF == sin(3*t/5)
g(t:SF):SF == sin(t)
draw(curve(f,g),0..%pi)
draw(x^2+y^3-1=0,x,y,range==[-1..1,-1..1])
v1:=draw(Gamma(i),i=-4.2..4,adaptive==true)
v2:=draw(1/Gamma(i),i=-4.2..4,adaptive==true)
putGraph(v2,getGraph(v1,1),2)
makeViewport2D(v2)
options: adaptive clip toScale curveColor pointColor
unit range coordinates

```

3D graphics

```

m(u:SF,v:SF):SF == 1
draw(m,0..2*%pi,0..%pi,coordinates==spherical)
options: title style colorFunction coordinates tubeRadius
tubePoints var1Steps var2Steps space

```

Discrete math

```

[x] = floor(x)    [x] = ceiling(x)
Remainder of n divided by k = rem(n,k) , k|n iff n%k==0
n! = factorial(n)    (x m) = binomial(x,m)
phi(n) = eulerPhi(n)    Tuples: (1,'Hello,x)

```

Type Conversions

```

r:=(2/3)*x^2-y+4/5 -> -y + 2/3 x^2 + 4/5
Type: Polynomial Fraction Integer
r::FRAC POLY INT -> -15y+10x^2+12
15
Type: Fraction Polynomial Integer
s:=(3+4*i)/(7+3*i) -> 33/58 + 19/58 %i
s::FRAC COMPLEX INT -> 3+4%i
7+3%i

```

Equation

```

eq1:=3*x+4*y=5 -> 4y + 3x = 5
eq2:=2*x+2*y=3 -> 2y + 2x = 3
lhs eq1 -> 4y + 3x
rhs eq1 -> 5
eq1+eq2 -> 6y + 5x = 8

```

Factored
g:=factor(4312) → 2 ³ 7 ² 11
unit g → 1
numberOfFactors g → 3
nthFactor(g,2) → 7
nthExponent(g,2) → 2
nthFlag(g,2) → "prime"
map(factor,55739/2520) → $\frac{139}{2^3} \frac{401}{3^2} \frac{7}{5} \frac{7}{7}$

List
a:=[1,2,3,4] → [1, 2, 3, 4]
b:=[3,4,5,6] → [3, 4, 5, 6]
append(a,b) → [1, 2, 3, 4, 3, 4, 5, 6]
cons(10,a) → [10, 1, 2, 3, 4]
empty? a → false
a.2 → 2
a.2 := 99 → [1, 99, 3, 4]
reverse b → [6, 5, 4, 3]

MakeFunction
expr :=(x + a) ³ → $x^3 + 3ax^2 + 3a^2x + a^3$
function(expr,f,x) → f
f(2) → $a^3 + 6a^2 + 12a + 8$
function(expr,g,a) → g
g(2) → $x^3 + 6x^2 + 12x + 8$

Matrix
A:=matrix([[1,2],[3,4]]) → $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
determinant A → -2
v:=vector([1,2]) → [1, 2]
A*v → [5, 11]
A ⁻¹ → $\begin{bmatrix} \frac{2}{3} & \frac{1}{2} \\ \frac{3}{2} & \frac{1}{2} \end{bmatrix}$
transpose(A) → $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$
nrows A → 2
ncols A → 2
nullity A → 0
rank A → 2
trace A → 5

Polynomial
x+1 yields Type Polynomial Integer
z-2.3 yields Type Polynomial Float
y ² -z+3/4 yields Type Polynomial Fraction Integer
p :=(y -1) ² * x * z → $(xy^2 - 2xy + x)z$
q :=(y -1)* x *(z +5) → $(xy - x)z + 5xy - 5x$
gcd(p,q) → $xy - x$
mainVariable p → z
variables p → [z, y, x]
degree(p,y) → 2
totaldegree p → 4
eval(p,x,w) → $(wy^2 - 2wy + w)z$
D(p,x) → $(y^2 - 2y + 1)z$
integrate(p,x) → $(\frac{1}{2}x^2y^2 - x^2y + \frac{1}{2}x^2)z$

PrimeField
x:PrimeField(7):=5 → 5
x ³ → 6
1/x → 3

Set
s :=brace([1,2,3,4,5]) → {1, 2, 3, 4, 5}
t :=brace([2,3,5,7]) → {2, 3, 5, 7}
intersect(s,t) → {2, 3, 5}
union(s,t) → {1, 2, 3, 4, 5, 7}
difference(s,t) → {1, 4}
insert!(7,s) → {1, 2, 3, 4, 5, 7}
remove!(7,s) → {1, 2, 3, 4, 5}
{1, 2, 1, <i>a</i> } = brace([1,2,1,'a]) (= {1, 2, <i>a</i> })
{ <i>f</i> (<i>x</i>) : <i>x</i> ∈ <i>X</i> , <i>x</i> > 0} ≈brace([f(x) for x in X x>0])

Special Functions
[fibonacci(k) for k in 0..] → [0,1,1,2,3,5,...]
[legendre(i,11) for i in 0..5] → [0,1,- 1,1,1,1]
[jacobi(i,15) for i in 0.5] → [0,1,1,0,1,0]
[eulerPhi i for i in 1..] → [1,1,2,2,4,2,...]
[moebiusMu i for i in 1..] → [1,- 1,- 1,0,- 1,1,...]
E1(0.01) → 4.0379295765381134
Gamma(0.01) → 99.432585119150588

Stream
)set streams calculate 6
ints := [i for i in 1..] → [1,2,3,4,5,6,...]
ints.20 → 20
[i for i in ints odd? i] → [1,3,5,7,9,11,...]

String
creation: s:= "Hello"
concatenate "He" "llo" → "Hello"
s(1)='H' s.1='H' s(2..3)='el' s(4..)= 'lo'
split("hi there",char " ") → ["hi","there"]
prefix?("He","Hello") → true
substring?("ll","Hello",3) → true

TwoDimensionalArray
creation: arr :ARRAY2 INT:=new(2,3,0) → $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
nrows arr → 2
ncols arr → 3
setelt(arr,1,1,17) → $\begin{bmatrix} 17 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
arr(1,1) → 17

Univariate Polynomial
creation: p :UP(x,INT):=(3*x-1) ² *(2*x+8)
q :UP(x,INT):=(1-6*x+9*x ²) ²
leadingCoefficient p → 18
degree p → 3
reductum p → $60x^2 - 46x + 8$
gcd(p,q) → $9x^2 - 6x + 1$
lcm(p,q) → $162x^5 + 432x^4 - 756x^3 + 408x^2 - 94x + 8$
resultant(p,q) → 0
p(2) → 300 (used as function)
D(p) → $54x^2 + 120x - 46$ (derivative)

Vector
creation: v := vector([1,2,3,4,5]) → [1, 2, 3, 4, 5]
length: #v → 5
access: v.2 → 2
add: v+v → [2, 4, 6, 8, 10]
multiply: 5*v → [5, 10, 15, 20, 25]
assign: v.2 := 7 → [1, 7, 3, 4, 5]