

```
/////////////////////////////////////////////////////////////////
//
//  MATLAB 95 COMMON SYNTAX
//
//
/////////////////////////////////////////////////////////////////
```

**Summary :**

1. Functions
2. Programming functions
3. L-Systems
4. Programming L-Systems

Appendix : A - Further developpements  
              B - Details

## I. Functions :

=====

\* Defined operators :

-Unary :

**!**VAL : NOT VAL

**-**VAL : ...

**|**VAL : as you imagine

**[**VAL : lower integer

**(**VAL) : override precedence

VAL**!** : factorial

-Binary :

**+** : Add

**-** : Subtract

**\*** : Multiply

**/** : Divide

**^** : exponent

**!=** : Different

**=** : Equal

**<** : Less

**>** : Above

**<=** : Less or equal

**>=** : Above or equal

**&** : AND

**~** : OR

**%** : MODULO

\* Defined functions :

-Constants :

pi,e,euler

random : returns a random number.

nothing : returns a mathematical error.

-One parameter :

sqrt,cur,ln,log,exp,ten

sin,cos,tan,cot

arcsin,arccos,arctan,arccot

sinh,cosh,tanh,coth

arcsh,arcch,arcth,arccth

intu

rnd

gamma

frac

fcos,fsin

INT

-Two parameters :

bessel,bessel2

L,T

logb

d

-Three parameters :

if

Note : For informations about any of these defined functions  
you should consult the programm menu Options,item "Grammars".

## II. Programming functions :

---

Specify a function :

[FUNCTION NAME] { ([variable1],[variable2],...) } := [definition] ;

Specify a draw :

draw [FUNCTION NAME] {[FUNCTION NAME] [...]} IN [DRAW TYPE] { AS [TYPE] } ;

where :

[DRAW TYPE] can be :  
PLANE  
SPACE

[TYPE] can be :  
in plane :  
DIFF,INTEG,SUMM,INTEG  
in space :  
DIFF

Specify number of vertices :

Vertices := [XVERTS] { , [YVERTS] } ;

Specify ranges :

Range1 := [START] { , [END] } ;

Range2 := [START] { , [END] } ;

Range3 := [START] ;

Specify a color :

Color := [TYPE] ;  
[TYPE] is either Blue or Grey

Include a library :

Include := [LIB NAME] ;

Notes :

The functions ARE case sensitive, but the rest not.

You can enter 20 functions.

You can specify up to ten [INCLUDE]

If you do not specify a [AS], the system will find the right draw (this is used for all standard draws). In case of an ambiguous draw-type, the [AS] declaration is expected.

### III. L-SYSTEMS

**F** : Draw a segment of length in the current direction

**f** : moves to next position without drawing.

**+** : Inc direction in X

**-** : Dec direction in X

**|** : Turn X direction

**&** : Inc direction in Z

**^** : Dec direction in Z

**/** : Inc direction in Y

**\** : Dec direction in Y

**[** : Push all on stack

**]** : Pop all from stack

**{** : Start a polygon

**}** : Draw a polygon

**!** : Decrease Width

**'** : Increase color-index

**%** : multiply length by factor

**\$** : divide length by factor

### IV. Programming L-SYSTEMS

Specify a generator :

[GENERATOR] -> [RULE] ;

Specify a probabilistic generator :

[GENERATOR] ( [PROBABILITY] ) -> [RULE] ;

note : the probabilities for a generator must be one.

a single probability can neither be greater than one, nor less or equal to zero.

Specify a draw :

draw IN [DRAW TYPE] { AS [TYPE] } ;

where :

[DRAW TYPE] can be :  
PLANE  
SPACE

[TYPE] can be :  
in plane :  
EL  
in space :  
CYL

Specify the Axiom :

Axiom := [AXIOM] ;

Specify the Angle :

Angle := [NUMBER] {°} {RAD};

Specify the iteration :

Iter := [INTEGER];

Specify the factor :

LFactor := [NUMBER];

Specify the width :

Width := [NUMBER];

Specify a color :

Color := [TYPE] ;

[TYPE] can be :  
Black Red Green Blue Brown Yellow

Notes :

- The EL system uses the **L** and **R** predicates, which draw edges  
(Left or Right ones).

- The CYL draw type uses cylindrical coordinates instead of spherical  
ones. Too see the changes, just try...

**A.**

**Further developpements** -if any :

- Faster computation of render for slow computers.
- Possibility of ray-tracing for perfect render
- Possible Texture mapping
- Poincaré section of 3-D Curves in any direction (not so easy to do).
- Superposition of functions ( in space)
- New standart functions
- Context-sensitive grammars for L-Systems (made hard due to brackets)
- Computation of constraint-functions in plane.
- Possibility of redefining the whole grammar,not only the identifiers
- New gadgets ( calculator etc. see MathLab II)
- Possible AVI support
- Symbolic computation support

## **B.**

### **Details**



General :

I wrote this program in two weeks, using Microsoft's fantastic Visual C++ 4.1 ( even if the linker is quite slow).

This was my first program for Windows 95, so be indulgent...

Using MathLab 95 :

You can either load a document, or start a new one.

If you do so, you can give your functions etc. in the left window.

Pressing the « Apply » button will either show you the errors, or will draw in the right window what you have asked.

Click in this window, then you can with the keypad rotate the curve etc...

(the 0 (Numpad must be on) will change representation type, and the 5 will (de-)activate polygon ordering...)

The Right button click will display info about the current drawing, and will allow you some operations.