Function Generator

This support note shows how to use the Function Generator of **WSxM** *program. It will describe each one of the controls and buttons that appears in this feature.*

Function Generator is a part of the WSxM program. In this dialog, you can edit mathematical or logical functions and use them to generate curves and images. This tool is available from WSxM v4.0. A more reduced version of the Function Generator is accessible in WSxM v3.0

Introduction

Opening the Function Generator

The Function Generator is available from the Representation Frame. To open this dialog, you can either click on the button  of the toolbar, or selecting **Function Generator…** in the **File** menu.

Creating a new function

First, you have to choose the **Dimension** of your new function. It can be one of the following values:

1. f(x)
2. f(x, y)
3. f(r, theta)

The **Dimension** will determine the number and types of variables that you can use in your expression**.**
Later you can select the interval for your function. Depending on the Dimension chosen, there may be up to four values to enter. You should also specify the physical units of the axis (X, Y and Z).

Now, you must enter a mathematical expression in the **Function Edit Box**. It can include numbers, arithmetic and logic operators and functions (any of the ones shown in the **List of Functions**).

When you have finished, you can press the **Apply** button to compile the expression. If the expression is correctly constructed, a preview of your function will be drawn in the **Preview Frame**. Otherwise, an error message describing the error will appear in **Compilation Results**.

Using images and curves

As any other number in your expression, you can use the value of an SPM image in some points. The available SPM functions are described in the **function overview**, in this document. For 1dSPM functions, you can use curves such as FZs, IZs, IVs, ZVs, etc. even you can specify one of the lines in the curve if there are many. For 2dSPM functions, you can use any SPM image. Remember that you have to open the image before you can use it. You cannot use these functions with f (r, theta) functions.

Selecting images and curves

When you are writing your function in the **Function Edit Box,** you can select an opened image for use with SPM functions. To do this, select one of these functions from the **List of Functions** by double-clicking in it. Now, click in the |spm| link. A new dialog will be opened, allowing you to select the desired image. When you have selected an image, you will see something like |spm_1435|. These numbers are only valid during this execution of WSxM. Therefore, although you copy the expression, you will have to change them next time. The whole sentence will appear "`2dSPM (x, y, |spm_1435|)`".

Dialog controls

APPLY

Compiles the mathematical expression written in the **Function Edit Box**. If that is correct, it shows the curve / image generated into the **Preview Frame.**

CREATE WINDOW

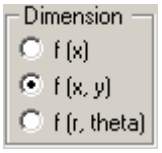
Creates a new window with the Image shown in the **Preview Frame.**

RESET

Clears the contents of the **Function Edit Box.**

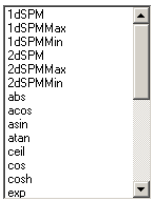
FINISH

Closes the Function Generator dialog.



DIMENSION

You can use these control to choose among the three supported function types.



LIST OF FUNCTIONS

That list includes all available functions**.** You can double-click on it to add the function to the **Function Edit Box**. You can apply a function after another by double-clicking in the list when there is something selected. For example, if you want to make the function cos (sin(x)), you can first write sin(x) and select it in the **Function Edit Box**. Now, if you double-click the function cos in the **List of Functions**, you will have cos (sin(x)).

FUNCTION EDIT BOX

You must write here your expression before generating the function with the **Apply** button. Function Generator is not case-sensitive, so you can write either 'abs' or 'ABS'.

If you want to know which parenthesis closes another one, you only have to select it (with cursor or Shift + arrows). The closing parenthesis will appear in bold style. When selecting certain functions in the **List of Functions** a hyperlink may appear. The text under the link will determine the type of the argument that you have to pass to the function:

|#|: you have to write a numeric expression. This includes numbers, functions returning a number, numeric variables, or any combination of them.
|Spm|: you have to write an image identifier.
|Curve|: you have to write a curve identifier.
|Var|: you have to write a valid variable identifier (i, j, my_x, etc.)

Since you cannot know which identifier corresponds to each image or curve, you have to click on the link to open a dialog that allows you to select the desired image.

LIMIT BOXES

In those boxes, you can enter the interval for the x and y-axis of your function, as well as the number of points. If you open the Function Generator when an image or curve is selected, these boxes are initialized with its data.

OPERATORS AND SYMBOLS

Many buttons represent operators, numbers and other symbols. By pressing them, you will add the corresponding item to the **Function Edit Box**.

DOUBLE PARENTHESES

By pressing this button, you will add '(' and ')' to both ends of the selected text.

The viewing settings controls:

Buttons in this group let you select some representation settings for the generated images.

CURVE SETTINGS

This button lets you choose which viewing settings will have the generated curve.

Original: sets the same settings as last selected curve.
Default: sets the WSxM default representation settings.

VIEW TYPE

When you are generating an image, you can select here between 2-d and 3-d representation.

PALETTE

This button lets you choose which palette will have the generated image.

Original: sets the same palette as last selected image.
Default: sets the WSxM default palette.

3D SETTINGS

This button lets you choose which 3-d settings will have the generated image.

Original: sets the same 3-d settings as last selected image.
Default: sets the WSxM default 3-d settings.

Note that these settings will be used although 2-d is selected in the "view type" button.

Function overview

Math functions

Here is a brief description of the mathematical functions that you can use in the Function Generator. They are also available in Lithography.

ABS function

- Syntax:
  `Abs (expression)`
- Return value:
  The absolute value of the given numeric expression.

ACOS function

- Syntax:
  `Acos (expression)`
- Return value:
  The angle, in radians, whose cosine is the given numeric expression.

ASIN function

- Syntax:
  `Asin (expression)`
- Return value:
  The angle, in radians, whose sine is the given numeric expression.

ATAN function

- Syntax:
  `Atan (expression)`
- Return value:
  The angle, in radians, whose tangent is the given numeric expression.

CEIL function

- Syntax:
  `Ceil (expression)`
- Return value:
  The smallest integer that is greater or equal to the given numeric expression.

COS function

- Syntax:
  `Cos (expression)`
- Return value:
  The cosine of the given numeric expression (in radians).

COSH function

- Syntax:
  `Cosh (expression)`
- Return value:
  The hyperbolic cosine of the given numeric expression (in radians).

EXP function

- Syntax:
  `Exp (expression)`
- Return value:
  The exponential value of the given numeric expression.

FLOOR function

- Syntax:
  `Floor (expression)`
- Return value:
  The largest integer that is less or equal than the given numeric expression.

H function (Heaviside step function)

- Syntax:
  `H (expression)`
- Return value:
  If expression is positive or zero, it returns 1.0 else, it returns 0.0.

LN function

- Syntax:
  `Ln (expression)`
- Return value:
  The natural logarithm of the given numeric expression.

LOG function

- Syntax:
  `Log (expression)`
- Return value:
  The base-10 logarithm of the given numeric expression.

PROD function

- Syntax:
  `Prod (var, start, end, step, expression)`
- Return value:
  The multiplicative series of expression `exp` from `var` = `start` to `end` with the defined `step`.
- Remarks:
  - You can use prod to implement a loop and of course, you can nest loops by using prod inside `exp`.
  - This function is not available in Lithography.

RAND function

- Syntax:
  `Rand ()`
- Return value:
  A pseudo-random number between zero and one (both inclusive).

SIN function

- Syntax:
  `Sin (expression)`
- Return value:
  The sine of the given numeric expression (in radians).

SINH function

- Syntax:
  `Sinh (expression)`
- Return value:
  The hyperbolic sine of the given numeric expression (in radians).

SQRT function

- Syntax:
  `Sqrt (expression)`
- Return value:
  The square root of the given numeric expression.

SUM function

- Syntax:
  `Sum (var, start, end, step, expression)`
- Return value:
  The summation of expression `exp` from `var` = `start` to `end` with the defined `step`.
- Remarks:
  - You can use sum to implement a loop and of course, you can nest loops by using sum inside `exp`.
  - This function is not available in Lithography.

TAN function

- Syntax:
  `Tan (expression)`
- Return value:
  The tangent of the given numeric expression (in radians).

TANH function

- Syntax:
  `Tanh (expression)`
- Return value:
  The hyperbolic tangent of the given numeric expression (in radians).

SPM functions

SPM functions are unique to Function Generator and you cannot use them in Lithography.

1dSPM function

- Syntax:
  `1dSPM (expression, curve_id, line_n)`
- Return value:
  The value of the curve with identifier `curve_id`, at line `line_n` when x = `expression`.

- Remarks:
  - The argument `curve_id` is optional. Last selected curve will be used if omitted.
  - The argument `line_n` is optional. Line 0 is the default line.
  - If you provide a value for `line_n`, you must also specify the value of `curve_id`.

1dSPMXByIndex / 1dSPMYByIndex functions

- Syntax:

```
1dSPMXByIndex (expression, curve_id, line_n)
1dSPMYByIndex (expression, curve_id, line_n)
```

- Return value:

  The value of the x / y coordinate, in real units, of the nth point within the curve with identifier `curve_id`, at line `line_n` . Points are referenced by zero-based indexes.

- Remarks:
  - The argument `curve_id` is optional. Last selected curve will be used if omitted.
  - The argument `line_n` is optional. Line 0 is the default line.
  - If you provide a value for `line_n`, you must also specify the value of `curve_id`.
  - If expression is not in the range [0, npoints-1], 0 is returned.

1dSPMMin / 1dSPMMax functions

- Syntax:

```
1dSPMMin(curve_id, line_n)
1dSPMMax(curve_id, line_n)
```

- Return value:

  The minimum (or maximum) value of line `line_n` in curve `curve_id`.

- Remarks:
  - The argument `curve_id` is optional. Last selected curve will be used if omitted.
  - The argument `line_n` is optional. It must be an integer in the range [-1, n-1], where n is the number of lines in the curve. If omitted (or –1 is put), the maximum (or minimum) value of the whole curve is used.
  - If you provide a value for `line_n`, you must also specify the value of `curve_id`.

2dSPM function

- Syntax:
  `2dSPM (expression_x, expression_y, image_id)`
- Return value:
  The z-value of image `image_id` at point (`expression_x, expression_y`).
- Remarks:
  The argument `image_id` is optional. Last selected image will be used if omitted.

2dSPMMin /2dSPMMax functions

- Syntax:
  `2dSPMMin (image_id)`
  `2dSPMMax (image_id)`
- Return value:
  The minimum (or maximum) value of image `image_id`.
- Remarks:
  The argument `image_id` is optional. Last selected image will be used if omitted.

3dSPM function

- Syntax:
  `3dSPM (expression_x, expression_y, fr, mov_id)`
- Return value:
  The z-value of the frame `fr`, in movie `mov_id` at point (`expression_x, expression_y`).
- Remarks:
  - The argument `fr` is an integer between 0 and N – 1, where N represents the number of frames in the movie.
  - The argument `mov_id` is optional. Last selected movie will be used if omitted.
  - The argument `fr` is optional. First frame will be used if omitted.

Other functions

These functions are also available in Lithography.

IF function

- Syntax:
  `If (condition, value1, value2)`
- Return value:
  The 'if' function tests 'condition'. If it is nonzero, the return value of this function is 'value1', else 'value2'.
- Remarks:
  Note that condition can be any valid numeric expression. However, you will better use Boolean expressions such as (x > y), (x < 0), (y > 0), etc. instead of (x), (log(y+1)), etc.

MAX function

- Syntax:
  `Max (expression1, expression2)`
- Return value:
  The maximum value of the two expressions.

MIN function

- Syntax:
  `Min (expression1, expression2)`
- Return value:
  The minimum value of the two expressions.

Operator overview

The Function Generator provides you many operators. These are:

Arithmetic	**Relational**	**Logical**
Exponentiation (^)	Less than (<)	Or
Negation (-)	Less than or equal to (<=)	And
Product (*)	Greater than (>)	Xor
Division (/)	Greater than or equal to (>=)	Not
Modulus (%)	Equality (==) **	
Addition subtraction (+, -)	Inequality (!=) **	

** Read note at the end of this section.

Although Function Generator lets you use any operator between any data type, it is a good practice to apply Arithmetic operators only to numeric values, logical operators to Boolean values and relational operators to values of the same type (either numeric or Boolean). However, if you need it, you can use numeric values where expecting Booleans. Function Generator will treat then zero values as '`false`' and non-zero values as '`true`', and vice versa. Relational and logical operators return Boolean values, while arithmetic ones return numeric values. By now, Boolean expressions are only useful in the **IF** function. You do not need (and cannot) write '`false`' and '`true`' explicitly so if you have to write an expression such as IF ((x > y) == true, x, y), you must write only this: IF ((x > y), x, y).

OR operator

- Syntax:

```
Expression1 or expression2
```

- Return value:

  '`true`' if either or both expressions evaluate to '`true`', else '`false`'.

AND operator

- Syntax:

```
Expression1 and expression2
```

- Return value:

  '`true`' if both expressions evaluate to '`true`', else '`false`'.

XOR operator

- Syntax:

```
Expression1 xor expression2
```

- Return value:

  '`true`' if one, and only one of the expressions evaluate to '`true`' , else '`false`'.

NOT operator

- Syntax:

```
Not expression
```

- Return value:

  The logical negation of expression.

NOTE: numbers as 0.0000001 are not zero and Function Generator may treat them as true. If you need to write an expression like "IF (x – y ! = 0.0, …)" try to use IF (abs(x-y) < 0.0000001, …) instead.

Predefined variables

Function Generator provides some useful variables that you can include in your expressions. Of course, depending on the **Dimension** chosen you will have access to x, y, r or theta. In addition to that, you can refer to some user-defined values with these variables:

xmin, xmax, xpoints, ymin, ymax, ypoints

They hold the limits entered in the **Limit** boxes.

Using Function Generator to process images

You can utilize Function Generator to apply several processes to an image, for example:
`F (x, y) = - (2dSPM(x, y))` is equivalent to a **Reverse** process.
`F (x, y) = 2dSPM (1-x, 1-y)` is the same as a 180° **Rotation** process (if both x and y go from 0 to 1).

Nanotec Electrónica S.L.
Centro Empresarial Euronova 3
Ronda de Poniente 2
Edificio 2, 1ª planta, oficina A
E28760 Tres Cantos (Madrid)
SPAIN
Phone: + 34 91 804 33 47
Fax : + 34 91 804 33 48
Web: www.nanotec.es
CIF: ES000B82006057

07/06/2005