

Operadores y funciones matemáticas en Gmsh

Gmsh 4.6.0

STEVEN VANEGAS GIRALDO
Universidad Nacional de Colombia - Sede Manizales

1. Comentarios y comandos

Para realizar los comentarios de una o para varias líneas de comandos se usa la siguiente sintaxis.

```
1 // comentario de una linea
2
3 /* comentarios de
4 varias lineas */
```

2. Constantes

La constante π se puede obtener con Pi.

3. Comandos

Cada comando debe terminar siempre en un ;

4. Operadores

Algunos operadores matemáticos que pueden ser usados en el programa Gmsh son presentados a continuación:

4.1. Operadores generales

Operador	Acción	Ejemplo	Resultado
=	Asignación	x = 7	x toma el valor de 7
+	Adición	x = 7 + 5	x toma el valor de 12
-	Sustracción	x = 7 - 5	x toma el valor de 2
*	Multiplicación	x = 7 * 7	x toma el valor de 49
/	División	x = 8 / 2	x toma el valor de 4
%	Módulo	x = 8 % 3	x toma el valor de 2
^	Potenciación	x = 8 ^ 2	x toma el valor de 64
+=	Asigna suma	x = 7 ; x += 7	x toma el valor de 14
-=	Asigna resta	x = 7 ; x -= 7	x toma el valor de 0
*=	Asigna multiplicación	x = 7 ; x *= 7	x toma el valor de 49
/=	Asigna división	x = 7 ; x /= 7	x toma el valor de 1

4.2. Operadores de relación

Operador	Tipo de relación
==	Igualdad
!=	Desigualdad
>	Mayor
>=	Mayor o igual
<	Menor
<=	Menor o igual

4.3. Operadores booleanos

Operador	Operación booleana
!	NOT lógico
&&	AND lógico
	OR lógico

5. Algunas funciones

Algunas funciones de programación en el programa Gmsh son presentadas a continuación :

Función	Operación	Consideraciones
Sin(x)	Seno de x	x en radianes
Cos(x)	Coseno de x	x en radianes
Tan(x)	Tangente de x	x en radianes
Asin(x)	Seno inverso de x	x comprendido entre $[-1, 1]$. Retorna un valor entre $[-\pi/2, \pi/2]$
Acos(x)	Coseno inverso de x	x comprendido entre $[-1, 1]$. Retorna un valor entre $[0, \pi]$
Atan(x)	Tangente inversa de x	Retorna un valor entre $[-\pi/2, \pi/2]$
Atan2(y, x)	Tangente inversa de y/x	Retorna un valor entre $[-\pi, \pi]$
Exp(x)	e^x	Retorna el valor de e^x
Fabs(x)	Valor absoluto de x	Retorna el valor de $ x $
Log(x)	Logaritmo natural de x	Retorna $\ln(x)$, con $x > 0$
Log10(x)	Logaritmo en base 10 de x	Retorna $\log(x)$, con $x > 0$
Sqrt(x)	Raíz cuadrada de x	Retorna \sqrt{x} con $x \geq 0$
Max(x1, x2)	Máximo de 2 argumentos	Retorna el máximo de x1 y x2
Min(x1, x2)	Mínimo de 2 argumentos	Retorna el mínimo de x1 y x2
Sinh(x)	Seno hiperbólico de x	
Cosh(x)	Coseno hiperbólico de x	
Tanh(x)	Tangente hiperbólico de x	
Rand(x)	Número aleatorio entre 0 y x	
Hypot(x1, x2)	$\sqrt{x1^2 + x2^2}$	
Floor(x)	Redondea x	Redondea al entero más cercano inferior
Round(x)	Redondea x	Redondea al entero más cercano
Ceil(x)	Redondea x	Redondea al entero más cercano superior

6. Ciclos y condicionales

6.1. Condicional if

La estructura del condicional if se presenta a continuación:

```
1 If (condicion1)
2     // segmento de codigo cuando condicion1 es verdadera
3
4 ElseIf (condicion2)
5     // segmento de codigo cuando condicion1 es falsa y condicion2 es verdadera
6
7     ...
8
9 Else
10    // segmento de codigo cuando son falsas las condiciones anteriores
11
12 EndIf
```

6.2. Condicional for

La estructura del condicional for y sus variaciones se presenta a continuación:

```
1 For i In {x1:x2} // itera i desde el valor de x1 al valor de x2 con un incremento de uno
2
3     // segmento de codigo dentro del ciclo
4
5 EndFor
```

```
1 For i In {x1:x2:paso} // itera i desde el valor de x1 al valor de x2 con un incremento de paso
2                       // paso puede ser positivo o negativo
3
4     // segmento de codigo dentro del ciclo
5
6 EndFor
```

7. Listas

La expresión de las listas tienen la siguiente sintaxis:

cadena = {elemento1, elemento2, ..., elementon};

Para hacer referencia algún elemento de una lista se puede realizar de la siguiente manera:

cadena[#elemento]

La numeración de los elementos empieza desde cero.

La nomenclatura se puede reducir con el operador :

pi : pf : paso

Donde:

- **pi** punto inicial (número entero)
- **pf** punto final (número entero)

8. Definir variables en ciclos For

Se puede usar la siguiente expresión para definir variables dentro de ciclos **for** con la concatenación de cadenas de texto.

cadena {variable} = cadena_variable

Un ejemplo de esto es nombrar variables a través de un ciclo **for**:

```
1 For i In {1:3}
2
3     x~{i} = i;
4
5 EndFor
```

El anterior código es equivalente a:

```
1 x_1 = 1;  
2 x_2 = 2;  
3 x_3 = 3;
```
