Operaciones aritméticas en Maple

José Luis Torres Rodríguez*

Febrero 2011

Gracias a los diversos tipos de datos disponibles, a las constantes simbólicas (por ejemplo **Pi**), a las funciones aritméticas y de manipulación numérica con las que cuenta, es posible utilizar Maple para realizar calculos numéricos y evaluar expresiones aritméticas. Esto, aunado a su capacidad para manejar números de precisión infinita y al manejo de aritmética racional, real y compleja, nos permiten considerar a este sistema como un poderoso ambiente para el manejo de operaciones artitméticas en general. Tales características de Maple son tratadas a lo largo de esta sección.

1. Funciones aritméticas

Anteriormente habian sido presentados los operadores artiméticos soportados por Maple, así como los operadores relacionales y algunas de las funciones elementales predefinidas en este sistema (vease el capítulo **Elementos Básicos de Maple** y el capítulo **Uso de Funciones**). Además de las ya mencionadas, existen algunas otras funciones, útiles en la manipulación de expresiones aritméticas. Entre ellas tenemos las que se muestran en la tabla 1.

Nombre	**Función**
Cociente entero	iquo(x)
Residuo entero	irem(x)
Máximo común divisor entre a y b:	mcd(a, b)
Mínimo común multiplo entre a y b	lcm(a, b)
Signo	signum(x)
Signo complejo	csignum(x)
Factorización en productos de primos	ifactor(x)

Cuadro 1: Funciones aritméticas

Otras funciones útiles para realizar cálculos numéricos son las trigonométricas, con sus respectivas formas inversas, hiperbólicas e hiperbólicas inversas, además de los operadores lógicos (todos ellos mencionados anteriormente). Ejemplos sobre el uso de éstos son presentados en las siguientes secciones.

*Coordinación de Cómputo, Facultad de Ciencias, UNAM

2. Uso de Maple como calculadora

Maple tiene la capacidad de realizar cualquier tipo de operación aritmética, por ejemplo:

```
> 4 + 5;
```

$$9$$

```
> 123.45*5423 - 82971;
```

$$586498{,}35$$

```
> 892/432.98765;
```

$$2{,}060104948$$

```
> 7/36 + 5/8 + 3/5;
```

$$\frac{511}{360}$$

Además, puede manipular números muy grandes, tanto enteros como de punto flotante (en ocasiones es utilizada la notación decimal).

```
> 100!;
```

$$\begin{aligned}&93326215443944152681699238856266700490715968264381621468592963895\backslash\\&217599993229915608941463976156518286253697920827223758251185\backslash\\&210916864000000000000000000000000\end{aligned}$$

```
> exp(245.0);
```

$$0{,}2524341263\,10^{107}$$

Nota: ademas del uso tradicional que se le da al operador "/" para expresar cocientes en la forma: **a/b**, Maple 8 también permite usarlo de la siguiente manera:

‘/‘(a, b)

más aun, puede ser usado como:

‘/‘(c)

lo cual es equivalente a la expresión:

1/c

Nótese que para hacer uso del operador como se describe aqui, éste debe estar delimitado por acentos agudos.

Por ejemplo:

```
> ‘/‘(7,36) = 7/36;
```

$$\frac{7}{36} = \frac{7}{36}$$

```
> ‘/‘(Pi*3!) = 1/(Pi*3!);
```

$$\frac{1}{6\,\pi} = \frac{1}{6\,\pi}$$

Este sistema nos proporciona diversas funciones por medio de las cuales podemos hacer manipulaciones sobre expresiones aritméticas o números, por ejemplo:

- Obtención de la parte fraccionaria.

```
> frac(evalf(sqrt(2)));
```

$$0{,}414213562$$

- Factorización de un entero, en productos de potencias de números primos.

```
> ifactor(-29!);
```

$$-(2)^{25}\ (3)^{13}\ (5)^6\ (7)^4\ (11)^2\ (13)^2\ (17)\ (19)\ (23)\ (29)$$

Este tipo de expresiones aritméticas pueden ser expandidas usando la función **expand**:

```
> expand(%);
```

$$-8841761993739701954543616000000$$

- Obtención del máximo común divisor y el mínimo común multiplo para dos enteros.

```
> gcd(26764, 4686);
```

$$2$$

```
> lcm(49234, 18932);
```

$$466049044$$

- Podemos verificar si una expresión arirmética corresponde a un número primo.

```
> isprime(11);
```

$$\mathit{true}$$

```
> isprime(23!);
```

$$\mathit{false}$$

```
> isprime(1323);
```

$$\mathit{false}$$

- También podemos evaluar expresiones como la conjetura de Fermat (según la cual $2^{(2^n)}+1$ es un número primo para n natural), para diferentes valores. Probaremos para $n = 1, 2, 3, 4, 5$.

```
> isprime(2^(2^1) + 1);
```

$$\mathit{true}$$

```
> isprime(2^(2^2) + 1);
```

$$\mathit{true}$$

```
> isprime(2^(2^3) + 1);
```

$$\mathit{true}$$

```
> isprime(2^(2^4) + 1);
```

$$\mathit{true}$$

```
> isprime(2^(2^5) + 1);
```

$$\mathit{false}$$

- Podemos calcular el primer primo mayor que un número cualquiera. Por ejemplo, calcularemos el primer número primo mayor que 4568.

```
> nextprime(4568);
```

$$4583$$

Verifiquemos el resultado:

```
> isprime(%);
```

$$true$$

- También podemos calcular el primer primo menor que un número. Calculemos el primero menor que 4568 y verifiquemos el resultado:

```
> prevprime(4568);
```

$$4567$$

```
> isprime(%);
```

$$true$$

Recuérdese que generalmente Maple realiza las operaciones utilizando artimética racional, lo cual quiere decir que las operaciones con enteros o fracciones siempre generan resultados enteros o fracciones. Por ejemplo:

```
> 21! + 56^2 + 8/5 - 16/9;
```

$$\frac{2299092397726924941112}{45}$$

No obstante, Maple realiza automáticamente algunas simplificaciones sencillas sobre racionales de tal manera que sean expresados en su forma más simple. Por ejemplo:

```
> 4/8;
```

$$\frac{1}{2}$$

```
> 876324/876432;
```

$$\frac{73027}{73036}$$

Cuando trabajamos con expresiones racionales, es necesario aplicar **evalf** para obtener un número de punto flotante, o bien, se debe incluir en la expresión por lo menos un número de punto flotante.

```
> evalf(21! + 56^2 + 8/5 - 16/9);
```

$$0{,}5109094217\,10^{20}$$

```
> 21! + 56^2 + 8/5 - 16/9.0;
```

$$0{,}5109094217\,10^{20}$$

Este forma de operar de Maple permite obtener resultados más exactos, pues se evitan todos los errores que puedan surgir al hacer redondeos o aproximaciones de números.

- Por otro lado, podemos también calcular el residuo y el cociente de una expresión artitmética racional:

```
> irem(89234, 45);
```

$$44$$

```
> iquo(89234, 45);
```

$$1982$$

- Otras operaciones útiles son las exponenciaciones:

```
> 8^8;
```

$$16777216$$

```
> 25^(2^3);
```

$$152587890625$$

En estas operaciones es necesario utilizar paréntesis para asociar. Por ejemplo:

```
> 2^2^2^2;
Error, `^` unexpected
```

Nos genera un mensaje de error, la forma correcta es:

```
> 2^(2^(2^2));
```

$$65536$$

Las operaciones de exponenciación pueden ser utilizadas en general para calcular x^p, donde p es un exponente entero, racional o decimal. Por ejemplo:

```
> 4^(1/2);
```

$$\sqrt{4}$$

```
> evalf(%);
```

$$2{,}000000000$$

```
> evalf(8^(2.8));
```

$$337{,}7940252$$

```
> evalf(34^Pi);
```

$$64756{,}43258$$

- También podemos calcular modulos:

```
> 98762 mod 8;
```

$$2$$

```
> 8! mod 3;
```

$$0$$

```
> 9749867432 mod 7;
```

$$4$$

La expresión "a mod b"nos devuelve el residuo al calcular la división entera $\frac{a}{b}$. Recuérdese que esta operación únicamente está definida para $\boldsymbol{a}$ y $\boldsymbol{b}$ enteros.

Veamos otro ejemplo. Comprobaremos a continuación que $\sqrt{2}+\sqrt{3}+\sqrt{5}$ es una raíz del polinomio $x^8 - 40\,x^6 + 352\,x^4 - 960\,x^2 + 576$.

```
> subs(x = sqrt(2) + sqrt(3) + sqrt(5), x^8 - 40*x^6 + 352*x^4 -
> 960*x^2 + 576);
```

$$\%1^8 - 40\ \%1^6 + 352\ \%1^4 - 960\ \%1^2 + 576$$
$$\%1 := \sqrt{2}+\sqrt{3}+\sqrt{5}$$

```
> evalf(%);
```

$$0{,}00007$$

Existen varias funciones más que nos permiten llevar a cabo operaciones sobre números, a continuación listaremos algunas de ellas:

- **add**. Nos permite calcular la suma de una suceción de valores.
- **mul**. Nos permite calcular el producto de una sucesión de valores.
- **Si**, **Ci**. Estas funciones están definidas de la siguiente forma: Para toda **x** compleja:
  Si(x) = int(sin(t)/t, t=0..x) Ci(x) = gamma + ln(x) + int((cos(t)-1)/t, t=0..x)
- **factorset(n)**. Calcula los factores primos de un entero **n**.
- **ithprime(i)**. Calcula el i-esimo primo.
- **Randpoly**, **Randprime**. Permiten calcular polinomios aleatorios sobre un campo finito.
- **rand**. Genera números aleatorios.
- **numer**, **denom**. Devuelven el numerador y denominador de una expresión, respectivamente.
- **frem**. Esta función devuelve **r = x - y*n**, donde **n** es el entero mas cercano a **x/y**.
- **irem**, **iquo**. Devuelven el residuo entero y cociente entero de dos expresiones, respectivamente.
- **rem**, **quo**. Calculan el residuo y cociente de dos polinomios.

También puede consultarse la ayuda del paquete **numtheory**, para obtener información de diversas funciones propias de Teoría de Números.

3. Precisión de una operación y aproximaciones de punto flotante

Maple es capaz de manejar números de punto flotante de precisión arbitraria, para ello es necesario utilizar la función **evalf**; ésta nos permite indicar a Maple con cuantos dígitos debe realizarse un cálculo numérico, con lo cual podemos obtener aproximaciones de diferente orden, para expresiones aritméticas.

Para ejemplificar lo anterior, calcularemos a continuación el cuadrado del número **Pi** con quinientos digitos de precisión:

```
> evalf(Pi^2, 500);
```

9,869604401089358618834490999876151135313699407240790626413349376\
2004482241920524300177340371855223182402591377402314407777\
3481220300467276106176779851976609903998562065756305715060\
4123284032878086935276934216493966657151904453873526177941\
3820258260581693412515592048309818873270033076266671104358\
9508715041003257885365952763577528379226833187450864045463\
5412502697372956695833422785815000636522709547249085975607\
2669264752779005285336452206669808264158968771057327889291\
746901545510069254432457036 3

A continuación obtendremos los primeros 200 dígitos de la parte fraccionaria del resultado de la instrucción anterior:

```
> frac(evalf(%, 200));
```

$$0{,}8696044010893586188344909998761511353136994072407906264133493762\backslash$$
$$2004482241920524300177340371855223182402591377402314407772\backslash$$
$$3481220300467276106176779851976609903998562065756305715060 4\backslash$$
$$12328403287808694$$

De forma predeterminada, Maple solo muestra los primeros 10 dígitos en cualquier operación de punto flotante (tomando en cuenta parte entera y fraccionaria). Por ejemplo:

```
> evalf(exp(4.8));
```

$$121{,}5104175$$

Este número de dígitos está determinado por una variable global definida por Maple, conocida como **Digits**. (Consultese la página de ayuda de **Digits** y otras variables de ambiente usando **?envvar**). Veamos el valor que contiene esta variable:

```
> Digits;
```

$$10$$

Al modificar el valor de esta variable, también modificamos el número de dígitos utilizados al hacer este tipo de evaluaciones. Por ejemplo, asignemosle el valor de 25:

```
> Digits := 25;
```

$$Digits := 25$$

Calculemos nuevamente evalf($e^{4.8}$):

```
> evalf(exp(4.8));
```

$$121{,}5104175187348807570481$$

Nótese como ahora obtenemos 25 dígitos en lugar de 10.

La cantidad de dígitos a desplegar tambien puede ser ajustado por medio de la opción **Preferences** del menú **File**, seleccionando la ficha **Numerics**.

4. Números complejos

Los números complejos son operados de la misma forma que los reales. Estos números son considerados por Maple como polinómios en términos del símbolo **I**, el cual representa la raiz cuadrada de menos uno. Maple utiliza siempre la relación $i^2 + 1 = 0$ para evaluar expresiones que contienen este tipo de números. Veamos un ejemplo:

```
> (4 + 2*I)/(2 - 5*I);
```

$$\frac{-2}{29} + \frac{24}{29}\,I$$

```
> evalf(%);
```

$$-0{,}06896551724137931034482759 + 0{,}8275862068965517241379310\,I$$

Podemos observar que al aplicar **evalf** a un complejo, esta función evalúa tanto la parte real como la parte imaginaria de este número.

```
> (32768 - sin(234)*I)*(9879 + exp(4)*Pi*I);
```

$$(32768 - \sin(234)\,I)\,(9879 + e^4\,\pi\,I)$$

Nótese que este último resultado no está evaluado. Existe una función que nos permite hacer este tipo de evaluaciones para complejos, su nombre es **evalc** (vease su ***página de ayuda***).

```
> evalc(%);
```

$$323715072 + \sin(234)\, e^4\, \pi + (-9879 \sin(234) + 32768\, e^4\, \pi)\, I$$

Otras funciones útiles en el manejo de complejos son:

- **Re**. Parte real de un complejo.
- **Im**. Parte imaginaria.
- **conjugate**. Conjugado complejo.
- **abs**. Módulo.
- **argument**. Argumento de un complejo.
- **csgn**. Para un número complejo determina en que parte del plano se encuentra éste.
- **signum**. Devuelve el sígno de una expresión real o compleja.
- **polar**. Calcula la forma polar de un número complejo.

Veamos a continuación algunos ejemplos.
Considerese el siguiente número:

```
> com1 := (12/4746 + 87641*I)*(234/92 - sin(9)*I);
```

$$\mathit{com1} := \left(\frac{2}{791} + 87641\, I\right)\left(\frac{117}{46} - \sin(9)\, I\right)$$

Calcularemos sus partes real e imaginaria:

```
> Re(com1);
```

$$\frac{117}{18193} + 87641 \sin(9)$$

```
> Im(com1);
```

$$-\frac{2}{791} \sin(9) + \frac{10253997}{46}$$

Calcularemos su conjugado, su módulo, su argumento y su signo como número complejo:

```
> conjugate(com1);
```

$$\left(\frac{2}{791} - 87641\, I\right)\left(\frac{117}{46} + \sin(9)\, I\right)$$

```
> abs(com1);
```

$$\frac{1}{36386}\sqrt{4805821274088965}\,\sqrt{13689 + 2116 \sin(9)^2}$$

```
> argument(com1);
```

$$\arctan\left(\frac{-\frac{2}{791}\sin(9) + \frac{10253997}{46}}{\frac{117}{18193} + 87641\sin(9)}\right)$$

```
> csgn(com1);
```

$$1$$

Este último resultado es calculado de la siguiente forma:

- **1**: si la parte real es positiva o si la parte real es cero y la imaginaria positiva.
- **0**: si el complejo es 0, y
- **-1**: en otro caso.

Podemos incluir variables dentro de expresiones dadas en términos de complejos. En este caso, Maple asume que dichas variables representan números reales, por lo cual son manejadas como tales. Por ejemplo:

```
> (a + b*I)*c;
```

$$(a + b\,I)\,c$$

```
> Re(%);
```

$$\Re((a + b\,I)\,c)$$

```
> Im(%%);
```

$$\Im((a + b\,I)\,c)$$

Las expresiones obtenidas representan las partes real e imaginaria del numero complejo, se muestran de forma simbólica ya que contienen valores indeterminados.

5. Sumatorias

Maple puede efectuar sumas tanto de números reales como de complejos. Una forma de hacer esta operación es a través de la función **sum**. La sintaxis es la siguiente:

sum(expr(x), x=a..b);

Donde **expr(x)** es una expresión (real o compleja), dada en términos de **x**; mientras que **a** y **b** son enteros, tales que **a<b**. Esta función calcula la suma **expr(a) + expr(a + 1) + expr(a + 2) +...+ expr(b)**. Veamos algunos ejemplos:

```
> sum(x + x^2*sin(x), x=1..10);
```

$$55 + \sin(1) + 4\sin(2) + 9\sin(3) + 16\sin(4) + 25\sin(5) + 36\sin(6) + 49\sin(7) + 64\sin(8) \\ + 81\sin(9) + 100\sin(10)$$

```
> evalf(%);
```

$$89{,}09859337474518314059243$$

```
> sum((4*r + Pi*r*I), r=2..20);
```

$$836 + 209\,I\,\pi$$

```
> evalf(%);
```

$$836. + 656{,}5928646002667868386924\,I$$

Recuérdese que esta función tiene una forma inerte. Veamos otros ejemplos:

```
> Sum((1 - i)/(1 + i), i=1..15);
```

$$\sum_{i=1}^{15} \frac{1-i}{1+i}$$

```
> value(%);
```

$$\frac{-3689561}{360360}$$

```
> evalf(%);
```

$$-10{,}23854201354201354201354$$

Esta función también nos permite manipular sumas indefinidas:

```
> Sum((1 - i)/(1 + i), i=1..n);
```

$$\sum_{i=1}^{n} \frac{1-i}{1+i}$$

```
> subs(n = 17,%);
```

$$\sum_{i=1}^{17} \frac{1-i}{1+i}$$

```
> evalf(%);
```

$$-12{,}00978384360737301913772$$

Veamos algunos ejemplos más con números complejos:

```
> Sum((2 + k*I)/(2 - k*I), k=0..10);
```

$$\sum_{k=0}^{10} \frac{2+k\,I}{2-k\,I}$$

```
> evalc(%);
```

$$\left(\sum_{k=0}^{10} \left(\frac{4}{4+k^2} - \frac{k^2}{4+k^2}\right)\right) + \left(\sum_{k=0}^{10} \left(\frac{4\,k}{4+k^2}\right)\right) I$$

```
> evalf(%);
```

$$-4{,}469122136618022415412289 + 6{,}619767013957377155356412\,I$$

```
> sum((234 + sin(t)^2*I), t=1..10);
```

$$2340 + \sin(2)^2\,I + \sin(9)^2\,I + \sin(4)^2\,I + \sin(6)^2\,I + \sin(8)^2\,I + \sin(10)^2\,I + \sin(1)^2\,I$$
$$+ \sin(3)^2\,I + \sin(5)^2\,I + \sin(7)^2\,I$$

```
> evalf(%);
```

$$2340. + 5{,}001430633485520535668588\,I$$

Otra forma de calcular sumas es por medio de la instrucción **add**, la cual nos permite obtener la suma de una sucesión de valores; su sintaxis es similar a la de **sum**. Por ejemplo:

```
> add(sin(i)^2, i=1..20);
```

$$\sin(1)^2 + \sin(2)^2 + \sin(3)^2 + \sin(4)^2 + \sin(5)^2 + \sin(6)^2 + \sin(7)^2 + \sin(8)^2 + \sin(9)^2$$
$$+ \sin(10)^2 + \sin(11)^2 + \sin(12)^2 + \sin(13)^2 + \sin(14)^2 + \sin(15)^2 + \sin(16)^2$$
$$+ \sin(17)^2 + \sin(18)^2 + \sin(19)^2 + \sin(20)^2$$

```
> evalf(%);
```

$$10{,}29712660081824500630002$$

A diferencia de **sum**, la función **add** no tiene una forma inerte y no puede realizar sumas simbólicas, solo puede sumar datos numéricos explícitos.

6. Productos

Otro tipo de operaciones válidas son los productos, tanto de números reales como de complejos. Una forma de calcularlos es por medio de la función **product**, su sintaxis es la siguiente:

product(expr(x), x=a..b);

En donde **expr(x)** es una expresión (real o compleja) en términos de **x**, y **a**, **b** son números enteros tales que **a**<**b**. Esta función también tiene una forma inerte. Consultese su ***página de ayuda***.Veamos algunos ejemplos:

```
> Product((j + 2)/(j + 3), j=1..10) = product((j + 2)/(j + 3), j=1..10);
```

$$\prod_{j=1}^{10} \frac{j+2}{j+3} = \frac{3}{13}$$

```
> product(d!*sin(d), d=1..20);
```

$$12744203123816107641872326695912457280940535764803996814646074786\backslash$$
$$210860158451928733118473415711895225119341900569911702323200\backslash$$
$$000000000000000000000000000000000 \sin(1)\sin(2)\sin(3)\sin(4)\sin(5)\sin(6)$$
$$\sin(7)\sin(8)\sin(9)\sin(10)\sin(11)\sin(12)\sin(13)\sin(14)\sin(15)\sin(16)\sin(17)$$
$$\sin(18)\sin(19)\sin(20)$$

```
> evalf(%);
```

$$-0{,}1679685154707858352623685\,10^{152}$$

Esta función también nos permite manipular productos indefinidos:

```
> product((k + 3)/k^2, k=1..n);
```

$$\frac{1}{6}\,\frac{\Gamma(n+4)}{\Gamma(n+1)^2}$$

```
> subs(n = 10,%);
```

$$\frac{1}{6}\,\frac{\Gamma(14)}{\Gamma(11)^2}$$

```
> evalf(%);
```

$$0{,}0000788139329805996472663139 3$$

Además, también se pueden calcular productos de complejos:

```
> product(I/(j + 3*I), j=1..10);
```

$$\frac{-7309}{458967717000} - \frac{5401}{1376903151000}\,I$$

```
> evalf(%);
```

$$-0{,}1592486733\,10^{-7} - 0{,}3922570731\,10^{-8}\,I$$

```
> Product(n*I/(4 + 2*I), n=6..15) = evalf(product(n*I/(4 + 2*I),
> n=6..15));
```

$$\prod_{n=6}^{15} \left(\frac{1}{10} + \frac{1}{5}\,I\right) n = 258{,}2656877 - 3395{,}594442\,I$$

Otra función que nos permite calcular productos es **mul**, la cual calcula el producto de una secuencia explícita de valores. A diferencia de **pruduct**, esta función no tiene una forma inerte. Su tintaxis es similar a la de **product**. Por ejemplo:

```
> datos:=[2, Pi, 4, 5*Pi, 8, 10];
```

$$datos := [2, \pi, 4, 5\,\pi, 8, 10]$$

```
> mul(x - i, i=datos);
```

$$(x-2)\,(x-\pi)\,(x-4)\,(x-5\,\pi)\,(x-8)\,(x-10)$$

Podemos usar la función **expand** para obtener el resultado del producto

```
> expand(%);
```

$$x^6 - 24\,x^5 + 196\,x^4 - 6\,x^5\,\pi + 144\,x^4\,\pi - 1176\,x^3\,\pi - 624\,x^3 + 3744\,x^2\,\pi + 5\,x^4\,\pi^2 - 120\,x^3\,\pi^2 + 980\,x^2\,\pi^2 - 3120\,x\,\pi^2 + 640\,x^2 - 3840\,x\,\pi + 3200\,\pi^2$$