# Coordenadas Baricéntricas

A continuación incluimos instrucciones de *Mathematica* para trabajar con coordenadas baricéntricas.

Francisco Javier García Capitán, 2005.

## ■ Constantes

### *Números*

Las siguientes constantes se usan en varios de los apartados que siguen

$$s = \frac{a+b+c}{2};\ SA = \frac{b^2+c^2-a^2}{2};\ SB = \frac{c^2+a^2-b^2}{2};\ SC = \frac{a^2+b^2-c^2}{2};$$

### *Puntos y rectas*

Los vértices y lados del triángulo de referencia *ABC* tienen las expresiones más simples. Podemos hacer asignaciones múltiples, teniendo en cuenta que *Mathematica* devuelve el valor de una asignación:

```
rtBC = ptA = {1, 0, 0};
rtCA = ptB = {0, 1, 0};
rtAB = ptC = {0, 0, 1};
ptI = {a, b, c};
ptIa = {-a, b, c};
ptIb = {a, -b, c};
ptIc = {a, b, -c};
```

## ■ Operaciones abstractas

### *Permutación de coordenadas*

Esta función será util cuando hemos efectuado un cálculo que toma como base un lado o ángulo del triángulo de referencia ABC y queremos calcular el cálculo correspondiente a los otros lados o ángulos:

```
PermutarTerna[{x_, y_, z_}] := {z, x, y} /.
   {a → b, b → c, c → a, u → v, v → w, w → u};
Permutar[f_] := f /.
   {a → b, b → c, c → a, u → v, v → w, w → u};
```

### *Sustituir un punto*

Se supone que *f* es una expresión en *u*, *v*, *w* y queremos sustituir (*u*, *v*, *w*) por (*x*, *y*, *z*):

```
Sustituir[f_, {x_, y_, z_}] := f /.
   {u → x, v → y, w → z};
```

### Simplificación de coordenadas

Las coordenadas baricéntricas homogéneas de un punto y los coeficientes de la ecuación de una recta los introducimos en *Mathematica* como ternas de puntos. Como dos de estas ternas se consideran equivalentes cuando son proporcionales, usamos la función **Simplificar** para conseguir la expresión más sencilla:

```
Simplificar::"usage" =
"Simplificar devuelve una terna dividida por su Máximo Común Divisor";
Clear[Simplificar];
Simplificar[p_] := Simplify[p / PolynomialGCD[p[[1]], p[[2]], p[[3]]]]
```

## ■ Cálculo de puntos y rectas

### Coordenadas de un punto y ecuación de una recta

Aprovechando la dualidad entre coordenadas de puntos y ecuaciones de rectas, podemos usar para ambos la función **Cross** para hallar el punto de intersección de dos rectas y la recta que une dos puntos.

```
Recta::"usage" =
  "Recta[P,Q] halla los coeficientes de la ecuación de la recta PQ";
Recta[P_, Q_] := Simplificar[Cross[P, Q]];
Punto::"usage" = "Punto[r,s] halla las
    coordenadas de la intersección de las rectas r y s";
Punto[r_, s_] := Simplificar[Cross[r, s]];
```

### Punto del infinito de una recta

El punto del infinito de la recta $p\,x + q\,y + r\,z = 0$ tiene coordenadas $(q - r : r - p : p - q)$

```
PuntoInfinito[{p_, q_, r_}] := Simplificar[{q - r, r - p, p - q}];
```

### Rectas paralelas

Calculamos la recta paralela a una recta *QR* pasando por un punto *P*:

```
Paralela[P_, Q_, R_] := Recta[P, PuntoInfinito[Recta[Q, R]]];
```

### Punto del infinito de rectas perpendiculares

Si una recta tiene a $(f : g : h)$ como punto del infinito, entonces el punto del infinito de cualquier recta perpendicular a ella es el punto del infinito de la recta $(\mathrm{SA}\,f)\,x + (\mathrm{SB}\,g)\,y + (\mathrm{SC}\,h)\,z = 0$.

```
PuntoInfinitoPerpendicular[{f_, g_, h_}] :=
  PuntoInfinito[{SA f, SB g, SC h}];
```

### Rectas perpendiculares

Hallamos la recta que pasa por *P* perpendicular a la recta *QR*:

```
Perpendicular[P_, Q_, R_] := Recta[P,
   PuntoInfinitoPerpendicular[PuntoInfinito[Recta[Q, R]]]];
```

### División de un segmento en una razón dada

La siguiente función calcula el punto *X* que divide al segmetno *PQ* en la razón m:n, es decir, *PX*:*XQ*=*m*:*n*.

```
DividirRazon[P_, Q_, m_, n_] := Simplificar[n Tr[Q] P + m Tr[P] Q]
```

## *Algunos casos particulares de puntos y rectas*

El punto medio de un segmento:

```
Medio[P_, Q_] := DividirRazon[P, Q, 1, 1]
```

La mediana de un triángulo *PQR* pasando por *P*:

```
Mediana[P_, Q_, R_] := Recta[P, Medio[Q, R]];
```

La mediatriz de un segmento:

```
Mediatriz[P_, Q_] := Perpendicular[Medio[P, Q], P, Q];
```

La altura de un triángulo *PQR* pasando por *P*:

```
Altura[P_, Q_, R_] := Perpendicular[P, Q, R];
```

La proyección de un punto *P* sobre una recta *QR*:

```
Pie[P_, Q_, R_] := Punto[
   Perpendicular[P, Q, R],
   Recta[Q, R]];
```

La proyección de un punto *P* sobre una recta *QR*:

```
SimetriaAxial[P_, Q_, R_] := Module[{H},
   H = Pie[P, Q, R];
   DividirRazon[P, H, 2, -1]];
```

## *Algunos puntos concretos:*

El baricentro como intersección de dos medianas:

```
ptG = Punto[
  Mediana[ptA, ptB, ptC],
  Mediana[ptB, ptC, ptA]]
```

$\{1, 1, 1\}$

El circuncentro como intersección de dos mediatrices:

```
ptO = Punto[
  Mediatriz[ptA, ptB],
  Mediatriz[ptA, ptC]]
```

$\{a^2 (a^2 - b^2 - c^2), b^2 (-a^2 + b^2 - c^2), c^2 (-a^2 - b^2 + c^2)\}$

El ortocentro como intersección de dos alturas:

```
ptH = Punto[
  Altura[ptA, ptB, ptC],
  Altura[ptB, ptC, ptA]]
```

$\{a^4 - (b^2 - c^2)^2, -a^4 + b^4 + 2 a^2 c^2 - c^4, -a^4 + 2 a^2 b^2 - b^4 + c^4\}$

# ■ Concurrencia y alineación

### *Determinar si los puntos de una lista están alineados*

Hacemos los determinantes de los dos primeros elementos de la lista y cada uno de los demás. Todos los determinantes deben ser cero.

```
EstanAlineados[lista_] := Module[{temp},
  temp = True;
  For[i = 3, i ≤ Length[lista], i++,
   If[! SameQ[Simplify[Det[{lista[[1]], lista[[2]], lista[[i]]}]], 0],
    temp = False]];
  temp]
```

### *Determinar si las rectas de una lista son concurrentes*

Aprovechamos la dualidad de puntos y rectas:

```
SonConcurrentes[lista_] := EstanAlineados[lista];
```

# ■ Distancias y áreas

### *Area de un trángulo*

La fórmula siguiente calcula el área del triángulo *ABC*, considerando como unidad el área del triángulo de referencia *ABC*:

$$\texttt{AreaTriangulo[P\_, Q\_, R\_]} := \frac{\texttt{Det[\{P, Q, R\}]}}{\texttt{Tr[P] Tr[Q] Tr[R]}}$$

### *Distancia entre dos puntos*

Introducimos la fórmula de la distancia de dos puntos:

```
CuadradoDistancia[{u_, v_, w_}, {x_, y_, z_}] := Factor[
   Divide[SA ((v + w) x - u (y + z))^2 +
     SB ((w + u) y - v (z + x))^2 + SC ((u + v) z - w (x + y))^2,
    (u + v + w)^2 (x + y + z)^2]];
```

### *Razón simple*

Cuadrado de la razón simple de tres puntos:

$$\texttt{CuadradoRazonSimple[P\_, Q\_, R\_]} := \texttt{Factor}\left[\frac{\texttt{CuadradoDistancia[P, R]}}{\texttt{CuadradoDistancia[R, Q]}}\right];$$

### *Razón doble*

Cuadrado de la razón doble de cuatro puntos:

$$\texttt{CuadradoRazonDoble[P\_, Q\_, R\_, S\_]} := \texttt{Factor}\left[\texttt{Divide}\left[\frac{\texttt{CuadradoDistancia[P, R]}}{\texttt{CuadradoDistancia[R, Q]}}, \frac{\texttt{CuadradoDistancia[P, S]}}{\texttt{CuadradoDistancia[S, Q]}}\right]\right];$$

# ■ Conversión de coordenadas

### *Conversión de coordenadas baricéntricas en coordenadas cartesianas*

La siguiente fórmula transforma la expresión *f* en función de x:y:z en otra expresión con coordenadas cartesianas:

```
BarCar[f_, cA_, cB_, cC_] := f /.
  {x → Det[( cB[[1]] - x  cC[[1]] - x
             cB[[2]] - y  cC[[2]] - y )],
   y → Det[( cC[[1]] - x  cA[[1]] - x
             cC[[2]] - y  cA[[2]] - y )],
   z → Det[( cA[[1]] - x  cB[[1]] - x
             cA[[2]] - y  cB[[2]] - y )],
   a → Norm[cB - cC], b → Norm[cC - cA], c → Norm[cA - cB]};
```

### *Coordenadas cartesianas de un punto dados sus baricéntricas y el triángulo de referencia:*

```
PuntoBarCar[{u_, v_, w_}, cA_, cB_, cC_] :=
  (u cA + v cB + w cC)/(u + v + w) /.
   {a → Norm[cB - cC], b → Norm[cC - cA], c → Norm[cA - cB]};
```

# ■ Enciclopedia de Kimberling

### *Coordenadas del triángulo de Kimberling*

Kimberling considera un triángulo con lados 6, 9 y 13:

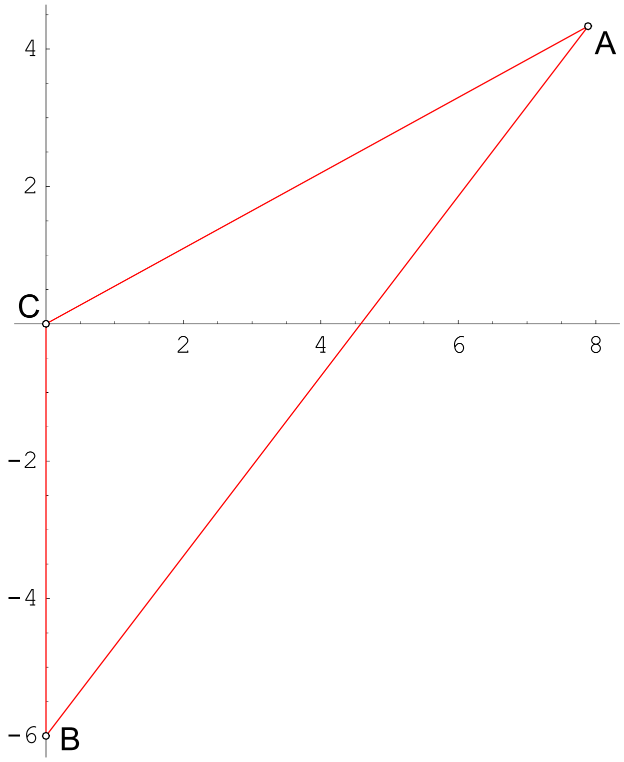
$$\texttt{kiA} = \left\{\frac{4\sqrt{35}}{3}, \frac{13}{3}\right\};\ \texttt{kiB} = \{0, -6\};\ \texttt{kiC} = \{0, 0\};$$

### *Primera coordenada de un punto en el triángulo de Kimberling:*

```
Kimberling[{u_, v_, w_}] :=
  N[(u kiA + v kiB + w kiC)/(u + v + w), 12] [[1]] /. {a → 6, b → 9, c → 13};
```

### *Gráfico del triángulo de Kimberling*

```
EscribirTexto[texto_, {x_, y_}, {dx_, dy_}] :=
  Text[texto, {x + dx, y + dy},
   TextStyle → {FontFamily -> "Arial", FontSize → 12 }];
circulito[{x_, y_}] := {
   RGBColor[1, 1, 1],
   Disk[{x, y}, 0.05],
   RGBColor[0, 0, 0],
   Circle[{x, y}, 0.05]};
puntos = Graphics[{circulito[kiA], circulito[kiB], circulito[kiC]}];
rotulos = Graphics[{
    EscribirTexto["A", kiA, {+0.25, - 0.25}],
    EscribirTexto["B", kiB, {+0.35, -.05}],
    EscribirTexto["C", kiC, {-0.25, 0.25}]
   }
  ];
triangulo = Graphics[{
    RGBColor[1, 0, 0],
    Line[{kiA, kiB, kiC, kiA}]}];
Show[{triangulo, puntos, rotulos},
 Axes → Automatic, AspectRatio → Automatic]
```




- Graphics -