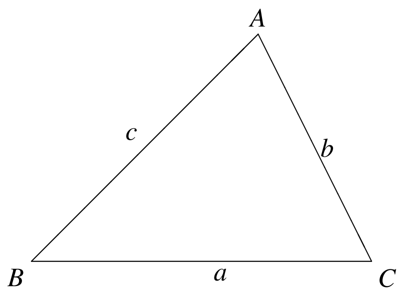
Revista digital Matemática, Educación e Internet (www.cidse.itcr.ac.cr/revistamate/). Vol. 1, Nº 2. Agosto – Diciembre 2001.

Resolver triángulos en Visual Basic. Parte 1/3

Luis Acuña P.
lacuna@itcr.ac.cr
Escuela de Matemática
Instituto Tecnológico de Costa Rica

Introducción

Uno de los últimos temas de trigonometría que se estudian en secundaria es la resolución de triángulos usando las leyes de senos y cosenos. En un triángulo cualquiera, las medidas más importantes son las longitudes de sus lados y las medidas de sus ángulos. En su forma más general, el problema de resolver un triángulo consiste en determinar las tres medidas desconocidas cuando se conocen tres.
En la notación usual, las letras a, b y c denotan los lados, y las mayúsculas A, B y C denotan los respectivos ángulos opuestos:



No todos los casos tienen solución. Por ejemplo, conocer las medidas de los tres ángulos no da ninguna pista acerca de las longitudes de los lados. Pero por el contrario, conocer los tres lados permite encontrar los ángulos sin problema.
Vamos a desarrollar un programa en Visual Basic que permita al usuario indicar tres datos cualesquiera, determine si es posible calcular las otras tres medidas, y muestre gráficamente la o las soluciones (dibujando un triángulo con los ángulos correctos y los lados en proporción a sus medidas).
En esta columna vamos a abordar parte del problema matemático. En la siguiente terminaremos con ese problema y desarrollaremos la interfaz con el usuario, pero por ahora hay varios detalles de programación que resolver.

1.1 Las leyes de senos y de cosenos

La clave para resolver el problema de determinar las tres medidas faltantes en un triángulo está en usar apropiadamente las siguientes fórmulas trigonométricas (todas en la notación usual que describimos arriba).

La ley de senos: En palabras, dice que los lados son proporcionales a los senos de los ángulos opuestos. La fórmula es

$$\frac{a}{\operatorname{sen} A} = \frac{b}{\operatorname{sen} B} = \frac{c}{\operatorname{sen} C}.$$

La ley de cosenos: Es una extensión del Teorema de Pitágoras a triángulos no rectángulos. Puede verse en tres formas distintas pero equivalentes:

$$\begin{aligned} a^2 &= b^2+c^2-2bc\cos A \\ b^2 &= a^2+c^2-2ac\cos B \\ c^2 &= a^2+b^2-2ab\cos C. \end{aligned}$$

En ninguna de las fórmulas está despejado un ángulo. Si se quiere encontrar el valor de un ángulo deberá despejarse de la fórmula apropiada (dependiendo de los datos que se conocen) y aplicar seno inverso o coseno inverso.
La función coseno tiene inversa en el dominio que nos interesa para triángulos, que es $[0,\pi]=[0^\circ,180^\circ]$. Pero la función seno no tiene inversa allí, sino en el dominio $[-\pi/2,\pi/2]=[-90^\circ,90^\circ]$. Por lo anterior, la ley de senos no es recomendable para encontrar ángulos obtusos.
En un triángulo, solamente el ángulo mayor podría ser obtuso. Entonces se recomienda que, de ser posible, no se use la ley de senos para encontrar el ángulo mayor. Los casos en los que esto es inevitable comúnmente llevan a dos soluciones: un ángulo obtuso y otro agudo.

1.2 Los distintos casos por resolver

Si los triángulos tienen seis medidas (tres ángulos y tres lados), son muchas las combinaciones de tres datos conocidos: pueden conocerse los tres ángulos, o dos de los ángulos y el lado entre ellos, o dos de los ángulos y un lado no entre ellos, etc. En total son ocho posibilidades, que por simetría se reducen a seis. Vamos a denotarlas con un código de tres letras, en el que las letras A y L denotan ángulo conocido y lado conocido, respectivamente. Los tres casos que mencionamos arriba se denotarán AAA (se conocen tres ángulos), ALA (dos ángulos y el lado entre ellos) y AAL o LAA (dos ángulos y un lado no entre ellos).
Las ocho posibilidades que mencionamos pueden agruparse de la siguiente manera:

1. El caso AAA: Este es el más fácil en el sentido de que no hay nada que hacer. Como ya mencionamos, no pueden encontrarse los lados si sólo se conocen los ángulos.

2. Los casos AAL (o LAA) y ALA: Entre los casos factibles, estos son los más sencillos. Conociendo dos ángulos y un lado, puede calcularse primero el tercer ángulo sabiendo que la suma de los tres es 180°, y luego usarse la ley de senos para cada uno de los lados faltantes.

3. El caso LAL: Conociendo dos lados y el ángulo entre ellos puede usarse la ley de cosenos para calcular el tercer lado, luego la ley de senos para encontrar el ángulo más pequeño entre los que faltan (recuérdese no usar la ley de senos para calcular el ángulo más grande, siempre que pueda evitarse), y finalmente determinar el tercer ángulo sabiendo que la suma de los tres es 180°.

4. El caso LLL: Si se tienen las longitudes de los tres lados, puede calcularse cada uno de los ángulos con la ley de cosenos, pero esto es más trabajo del necesario, porque la ley de cosenos es más complicada que la de senos. Para resolver el problema a mano podría empezarse usando la ley de cosenos para encontrar el ángulo mayor, luego la ley de senos para cualquiera de los otros dos ángulos (a fin de no usar la ley de senos para el ángulo mayor), y por último calcular el tercero restando de 180° las medidas de los otros dos.

5. El caso LLA (o ALL): Al conocer dos lados y un ángulo no entre ellos, debe empezarse por calcular el ángulo desconocido opuesto a un lado conocido, con la ley de senos. Pero éste puede ser obtuso o agudo (dos soluciones), recto (una solución) o puede no existir. En caso de haber solución, se encuentra el tercer ángulo restando de 180°, y por último el lado faltante por ley de senos.

1.3 Las subrutinas principales

Nuestro proyecto en Visual Basic tendrá un módulo estándar con los procedimientos que resolverán los distintos casos. La interfaz visual, como dijimos, queda para la próxima columna. Por ahora vamos a escribir un procedimiento para cada uno de los casos 2–4.
Para crear un módulo estándar en un proyecto se usa la opción **Project** | **Add Module**. En la ventana de propiedades definiremos que el nombre del módulo es `Funciones`.
Las subrutinas que resuelven los casos tendrán un nombre de la forma `CasoXXX`, donde en vez de `XXX` usaremos las tres letras que describen el caso. Cada subrutina recibirá seis parámetros: los tres lados y los tres ángulos, pero no en ese orden. Tres de los parámetros serán datos, y los escribiremos primero. Los otros tres serán los resultados, y serán los últimos.
Cada una de las subrutinas recibirá sus tres datos como parámetros por valor. En Visual Basic esto se indica con la palabra `ByVal` antes del nombre de cada parámetro. Por otra parte, las variables donde se retornarán los resultados serán parámetros por referencia, indicados con la palabra `ByRef`. (Si no se indica ninguna de esas dos palabras, Visual Basic usa parámetros por referencia. Aquí vamos a usar las dos palabras explícitamente para dejar bien clara la intención.)
La notación que vamos a usar será la siguiente: Los lados se denotarán `a`, `b` y `c`, y los ángulos serán `angA`, `angB` y `angC`. Como las funciones trigonométricas en Visual Basic trabajan en radianes, supondremos aquí que los ángulos están indicados en radianes. Las variables reales serán de tipo `Double` para aprovechar toda la precisión que ofrece Visual Basic.
Si uno de los datos fuera inválido (como una longitud negativa o un ángulo mayor que 180°), las subrutinas retornarán el valor 0 en los tres resultados.
El encabezado del módulo `Funciones` contiene las siguientes dos líneas:

```
Option Explicit
Const pi = 3.141592653589793
```

Los casos AAL (o LAA) y ALA

Aquí conocemos dos de los ángulos y un lado. Dentro del procedimiento denotaremos `angA` y `angB` los ángulos, y `c` el lado conocido. El método que usamos aquí funciona independientemente de que el lado conocido esté o no entre los dos ángulos.
Como dijimos, empezamos por calcular el tercer ángulo, `angC`, restando los otros dos de π (en realidad dijimos que restaríamos de 180°, pero vamos a trabajar en radianes). Luego usaremos la ley de senos dos veces: una para cada lado faltante.
En notación matemática, las fórmulas son:

- $C = \pi - A - B$
- $\dfrac{a}{\operatorname{sen} A} = \dfrac{c}{\operatorname{sen} C} \Rightarrow a = \dfrac{c \operatorname{sen} A}{\operatorname{sen} C}$
- $\dfrac{b}{\operatorname{sen} B} = \dfrac{c}{\operatorname{sen} C} \Rightarrow b = \dfrac{c \operatorname{sen} B}{\operatorname{sen} C}$

Los datos serán válidos mientras $A + B < 180°$ y $c > 0$.
En Visual Basic, el procedimiento es así:

```
Public Sub CasoAAL(ByVal angA As Double, ByVal angB As Double, ByVal c As Double, _
    ByRef a As Double, ByRef b As Double, ByRef angC As Double)
'   Resuelve el caso de ngulo-ngulo-lado conocidos.
'   Calcula primero el ngulo desconocido y luego
'   los dos lados por la ley de senos.
'   Recibe los datos angA, angB y c, y calcula
```

```
'   los resultados a, b y angC.

    ' validar los datos: debe ser angA+angB<pi y c>0
If angA + angB >= pi Or c <= 0 Then
    a = 0: b = 0: angC = 0
    Exit Sub
End If

    ' calcular el ngulo C por diferencia a pi
angC = pi - angA - angB
    ' calcular el lado a por ley de senos
a = c * Sin(angA) / Sin(angC)
    ' calcular el lado b por ley de senos
b = c * Sin(angB) / Sin(angC)
End Sub
```

El caso LAL

Conociendo dos lados y el ángulo entre ellos, se empieza por calcular el tercer lado por la ley de cosenos. En segundo lugar, se calcula el ángulo más pequeño (entre los dos restantes) por la ley de senos. Y por último, el tercer ángulo se calcula restando los dos primeros de π.
Denotemos los datos a, B, c (lado, ángulo, lado). Suponiendo que el ángulo menor entre A y C es A (lo cual se reconoce porque $a < c$), calculamos:

- $b^2 = a^2 + c^2 - 2ac\cos B \Rightarrow b = \sqrt{a^2 + c^2 - 2ac\cos B}$
- $\dfrac{\operatorname{sen} A}{a} = \dfrac{\operatorname{sen} B}{b} \Rightarrow A = \operatorname{arcsen}\left(\dfrac{a \operatorname{sen} B}{b}\right)$
- $C = \pi - A - B$

Los datos serán válidos si $a, c > 0$ y $0 < B < 180°$.
En Visual Basic, denotando los datos a, angB y c, y los resultados angA, b y angC, escribimos:

```
Public Sub CasoLAL(ByVal a As Double, ByVal angB As Double, ByVal c As Double, _
    ByRef angA As Double, ByRef b As Double, ByRef angC As Double)
'   Resuelve el caso de lado-ngulo-lado conocidos.
'   Calcula primero el lado desconocido, luego el
'   ngulo menor por la ley de senos.
'   Recibe los datos a, angB y c y calcula
'   los resultados angA, b y angC.

    ' validar los datos: debe ser a,c>0 y 0<angB<pi
If a <= 0 Or c <= 0 Or angB <= 0 Or angB >= pi Then
    angA = 0: b = 0: angC = 0
    Exit Sub
End If

    ' calcular el lado b por ley de cosenos
b = Sqr(a ^ 2 + c ^ 2 - 2 * a * c * Cos(angB))

    ' calcular el lado menor por ley de senos
    ' y el otro por diferencia a pi
If a < c Then
    angA = ArcSen(a * Sin(angB) / b)
```

```
    angC = pi - angA - angB
Else
    angC = ArcSen(c * Sin(angB) / b)
    angA = pi - angB - angC
End If
End Sub
```

Hay una trampita en el código anterior: se hace referencia inocentemente a una función `ArcSen`, que no es parte del lenguaje Visual Basic. Luego, en la sección “Funciones auxiliares”, definiremos las funciones arcsen y arccos.

El caso LLL

Cuando se conocen los tres lados, dijimos arriba que usar la ley de cosenos para encontrar cada ángulo era más trabajo del necesario, porque la ley de cosenos es más complicada que otros métodos. Cuando un problema se resuelve manualmente, es recomendable usar la ley de cosenos para encontrar el ángulo mayor, luego la de senos para cualquiera de los otros dos ángulos, y por último calcular el tercero restando de π las medidas de los otros dos. Recuérdese no usar, siempre que pueda evitarse, la ley de senos para encontrar el ángulo mayor.
Pero para una computadora no es mucho trabajo usar la ley de cosenos tres veces. Más bien sería mucho el trabajo del programador si nos ocupáramos de que el primer ángulo sea el mayor. Preferimos la siguiente solución, manteniendo la recomendación del párrafo anterior cuando el problema se resuelve manualmente.

- $a^2 = b^2 + c^2 - 2bc\cos A \Rightarrow \cos A = \dfrac{b^2+c^2-a^2}{2bc} \Rightarrow A = \arccos\left(\dfrac{b^2+c^2-a^2}{2bc}\right)$
- $B = \arccos\left(\dfrac{a^2+c^2-b^2}{2ac}\right)$
- $C = \arccos\left(\dfrac{a^2+b^2-c^2}{2ab}\right)$

Para que los datos sean válidos es necesario que cada lado sea menor que la suma de los otros dos.

```
Public Sub CasoLLL(ByVal a As Double, ByVal b As Double, ByVal c As Double, _
    ByRef angA As Double, ByRef angB As Double, ByRef angC As Double)
'   Resuelve el caso de lado-lado-lado conocidos.
'   Calcula cada ngulo por la ley de cosenos.
'   Recibe los datos a, b y c y calcula
'   los resultados angA, angB y angC.

    ' validar los datos: cada lado menor que la suma de los otros
If a >= b + c Or b >= a + c Or c >= a + b Then
    angA = 0: angB = 0: angC = 0
    Exit Sub
End If

    ' calcular cada ngulo por ley de cosenos
angA = ArcCos((b ^ 2 + c ^ 2 - a ^ 2) / (2 * b * c))
angB = ArcCos((a ^ 2 + c ^ 2 - b ^ 2) / (2 * a * c))
angC = ArcCos((a ^ 2 + b ^ 2 - c ^ 2) / (2 * a * b))
End Sub
```

La función arccos no existe en Visual Basic. Luego la definiremos.

El caso LLA (o ALL)

Este es el caso más complejo, ya que dos lados y un ángulo no entre ellos no determinan completamente un triángulo. Vamos a dejar este caso para la próxima columna.

1.4 Funciones auxiliares

Visual Basic no define las funciones arcsen ni arccos. Lo más cercano es la función `Atn(x As Double) As Double`, que calcula el arcotangente de `x`, en radianes. Las otras dos funciones pueden definirse a partir de ésta.
Para arcsen, empecemos por plantear $y = \operatorname{sen} x$. El objetivo es escribir x en términos de y y de la función arctan. Nótese que $\cos^2 x = 1 - y^2$, por lo que $\tan^2 x = y^2/(1-y^2)$. Entonces $\tan x = y/\sqrt{1-y^2}$ para $x \in]-\pi/2, \pi/2[$ y $y \in]-1, 1[$. Finalmente, $x = \arctan(y/\sqrt{1-y^2})$; es decir,

$$\operatorname{arcsen} y = \arctan \frac{y}{\sqrt{1-y^2}}.$$

Las dos excepciones, $y = \pm 1$, deben tratarse por aparte: $\operatorname{arcsen} \pm 1 = \pm\pi/2$. La función `Sgn` en Visual Basic retorna el signo del argumento: 1 si el argumento es positivo, -1 si es negativo, ó 0 si es 0. Entonces:

```
Public Function ArcSen(y As Double) As Double
'   Calcula el seno inverso de y

If y = 1 Or y = -1 Then
    ArcSen = pi / 2 * Sgn(y)
Else
    ArcSen = Atn(y / Sqr(1 - y ^ 2))
End If
End Function
```

El arccos es un poco más complicado porque su dominio no es igual al de arctan, como sí era el de arcsen. Si $y = \cos x$ entonces $\operatorname{sen}^2 x = 1 - y^2$ y de aquí $\tan^2 x = (1-y^2)/y^2$. Entonces $\tan x = \pm\sqrt{1-y^2}/y$. Si $y > 0$ es porque $x \in [0, \pi/2[$, y entonces $x = \arctan(\sqrt{1-y^2}/y)$. Pero si $y < 0$ debe ser $x \in]\pi/2, \pi]$, por lo que $x = \pi + \arctan(\sqrt{1-y^2}/y)$. Entonces

$$\arccos y = \begin{cases} \arctan \dfrac{\sqrt{1-y^2}}{y} & \text{si } y > 0 \\ \pi + \arctan \dfrac{\sqrt{1-y^2}}{y} & \text{si } y < 0. \end{cases}$$

El caso $y = 0$ no cabe dentro de lo anterior. Para ese caso, $\arccos 0 = \pi/2$. En resumen:

```
Public Function ArcCos(y As Double) As Double
'   Calcula el coseno inverso de y
Dim a As Double

If y = 0 Then
    ArcCos = pi / 2
Else
    a = Atn(Sqr(1 - y ^ 2) / y)
    If y > 0 Then ArcCos = a Else ArcCos = pi + a
End If
End Function
```

1.5 Lo que falta

Como dijimos, falta resolver el caso LLA, que puede tener cero, una o dos soluciones. Y también falta algo importantísimo: ‰la interfaz con el usuario! Eso queda para la próxima columna, pero ya avanzamos mucho en el desarrollo del “motor” matemático del programa.