MARROQUIN CANO

 FIC/UNACH

 PROGRAMACION

 PROYECTO INDIVIDUAL



**CALCULO DE CATENARIAS**

La palabra Catenaria deriva del [latín](http://es.wikipedia.org/wiki/Lat%C3%ADn) *catenarĭus* (*propio de la cadena*). Por extensión, en [matemáticas](http://es.wikipedia.org/wiki/Matem%C3%A1ticas) se denomina *catenaria* a la curva que adopta una cadena, [cuerda](http://es.wikipedia.org/wiki/Cuerda) o [cable](http://es.wikipedia.org/wiki/Cable_mec%C3%A1nico) ideal perfectamente flexible, con masa distribuida uniformemente por unidad de longitud, suspendida por sus extremos y sometida a la acción de un campo gravitatorio uniforme, gráficamente esta curva se puede confundir con una parábola como lo hizo Galileo Galilei, pero fue [Christiaan Huygens](http://es.wikipedia.org/wiki/Christiaan_Huygens) quien demostró a los 17 años que no era una parábola, pero no supo obtener la ecuación de la catenaria. La ecuación finalmente fue encontrada por el propio Huygens, Johann Bernoulli y Gottfried Leibniz en el año 1691. La ecuación de la catenaria viene definida por:



En general la ecuación de la catenaria se refiere a cadenas o cuerdas infinitamente flexibles e inextensibles. El requisito de flexibilidad infinita se refiere a que la [rigidez flexional](http://es.wikipedia.org/wiki/Rigidez) sea nula y el requisito de inextensibilidad se refiere a que la longitud de cada tramo de la misma no varíe a pesar de estar sometido a fuerzas. Obviamente en las cuerdas reales estos requisitos se cumplen sólo de forma aproximada. Para cuerdas de gran longitud, la elasticidad de la cuerda las aleja del comportamiento perfectamente inextensible. Si bien la catenaria de una cuerda inextensible es siempre una curva plana, para cables gruesos de pequeña longitud la rigidez flexional finita hace que su deformada no necesariamente esté contenida en un plano.

FORMULACION DISCRETA

Sea una cadena de bolitas metálicas como las que se utilizan para sujetar los tapones de los fregaderos. Supondremos que hay *N* bolitas igualmente repartidas sobre un hilo de longitud *L*y de masa despreciable.



Cada bolita estará, por tanto, sometida a tres fuerzas: su propio peso, la fuerza que ejerce el hilo a su izquierda y a su derecha.

La condición de equilibrio para la bolita *i* de masa *m* se expresa



Todas las componentes horizontales de la tensión del hilo son iguales, y la denominaremos *Tx*.

*Tx=T*cos*0= T*cos*i= T*cos*i+1 =T*cos*N+1*



A la cantidad constante cociente entre el peso de cada bolita *mg* y la componente horizontal *Tx* de la tensión del hilo, le denominaremos parámetro  . La relación de recurrencia se escribe para cada bolita *i*=1*... N.*

tan*1=*tan*0-*tan*2=*tan*1-*tan*3=*tan*2-
...............*tan*i=*tan*i-1-
.............*tan*N-1=*tan*N-2-*tan*N=*tan*N-1-*

Sumando miembro a miembro obtenemos el ángulo *N* en función del ángulo inicial *0*.

tan*N=*tan*0-N*

Si los extremos del hilo están a la misma altura, por razón de simetría tendremos que

tan*0=-* tan*N*

Por tanto,

tan*0*=*N /2*

Sumando miembro a miembro la relación de recurrencia hasta el término *i*, obtenemos el ángulo *i* en función del ángulo inicial *0*.

tan*i=*tan*0- i=*(*N-*2*i*)*·*/2

El ángulo *i*que forma el hilo con la horizontal en la posición de cada una de las bolitas, el ángulo inicial *0* y el final *N* se calculan mediante la siguiente fórmula



Las coordenadas (*xi, yi*) de la bolita *i* se obtendrán sumando las proyecciones *d·*cos* j* y *d·*sen* j, j=*0*...i-1,* sobre el eje X y sobre el eje Y respectivamente, siendo *d* la distancia entre dos bolitas consecutivas *d=L/*(*N+*1)

|  |  |
| --- | --- |
| catenaria2_1.gif (2209 bytes) | *http://www.sc.ehu.es/sbweb/fisica/solido/din_rotacion/catenaria/Image1422.gif* |

DEDUCCION MATEMATICA

Aplicando el equilibrio de fuerzas a las fuerzas horizontales y verticales fuerzas a una porción infinitesimal de catenaria, como se mencionó anteriormente, se tiene que:



Donde :

 es el ángulo formado por la catenaria y la horizontal.

 es la tensión total del cable para cada punto.

 es el peso por unidad de longitud.

La primera de las ecuaciones implica que  mientras que la segunda de ellas puede escribirse escogiendo adecuadamente el origen de la longitud de arco como:



Introduciendo la relación entre la tangente del ángulo de la pendiente y la longitud de arco:



Derivando la última relación se obtiene: 

Donde:

 es el peso por unidad de longitud.

* la tensión horizontal que aparecerá en los extremos del cable.

Siendo esta ecuación quien relaciona las tensiones en los extremos de un tramo y el peso del mismo

La solución general viene dada por:



La solución para de la ecuación anterior para un cable suspendido de dos puntos a la misma altura y cuyo punto mínimo es, tomando su mínimo en el punto (0,a) resulta ser:



Es la componente horizontal de la tensión, que es constante, es el peso por unidad de longitud del hilo y  es la función [coseno hiperbólico](http://es.wikipedia.org/wiki/Coseno_hiperb%C3%B3lico). Si se desarrolla en [series de Taylor](http://es.wikipedia.org/wiki/Serie_de_Taylor) la ecuación de la catenaria se obtiene una curva cercana a una parábola:

![y(x) \approx a\left[ \left(1+\frac{C_2}{a}\right) + \frac{1}{2}\frac{(x-C_1)^2}{a^2} \right] + O(x^4)]()

Esto corresponde a la ecuación de una parábola más un término de cuarto orden. Es por este motivo que las gráficas son tan parecidas en el entorno de cero.

**Relaciones importantes**

La longitud del arco, con el origen de arco en el mínimo es:



La tensión total del hilo es:





***PROGRAMACION***

***UNIT 1***

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit3.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

Double d, h, v, l, g, t, m;

TForm3 \*Form3;

//---------------------------------------------------------------------------

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

 : TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button1Click(TObject \*Sender)

{

h=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

v=(pow(2.7182818281828182,d/h)-(pow(2.7182818281828182,-d/h)));

l=v\*h;

Edit3->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button2Click(TObject \*Sender)

{

g=Edit4->Text.ToDouble();

t=g\*l;

Edit5->Text=AnsiString(t);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::RadioButton1Click(TObject \*Sender)

{

if(RadioButton1->Checked==true){

 l=l\*1000;

 }

Edit3->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::RadioButton2Click(TObject \*Sender)

{

if(RadioButton2->Checked==true){

 l=l;

 }

Edit3->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button3Click(TObject \*Sender)

{

h=Edit1->Text.ToDouble();

m=h/2;

Edit6->Text=AnsiString(m);

Edit7->Text=AnsiString(h);

Edit8->Text=AnsiString(h);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button4Click(TObject \*Sender)

{

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button5Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------

***UNIT 2***

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit3.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

Double a, b, c, d, v, l, t, m, h, w, g;

TForm2 \*Form2;

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

 : TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button1Click(TObject \*Sender)

{

a=Edit1->Text.ToDouble();

b=Edit2->Text.ToDouble();

c=Edit3->Text.ToDouble();

d=Edit4->Text.ToDouble();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button2Click(TObject \*Sender)

{

w=pow(((c-a)\*(c-a))+((d-b)\*(d-b)),0.5);

Edit5->Text=AnsiString(w);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button3Click(TObject \*Sender)

{

h=Edit6->Text.ToDouble();

v=(pow(2.7182818281828182,d/h)-(pow(2.7182818281828182,-d/h)));

l=v\*h;

Edit7->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button4Click(TObject \*Sender)

{

g=Edit8->Text.ToDouble();

t=g\*l;

Edit9->Text=AnsiString(t);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button5Click(TObject \*Sender)

{

h=Edit6->Text.ToDouble();

m=h/2;

Edit10->Text=AnsiString(m);

Edit11->Text=AnsiString(h);

Edit12->Text=AnsiString(h);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button6Click(TObject \*Sender)

{

Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::Button7Click(TObject \*Sender)

{

Form3->Visible=true;

}

//-------------------------------------------------------------------------------

***UNIT 3***

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit3.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

Double d, h, v, l, g, t, m;

TForm3 \*Form3;

//---------------------------------------------------------------------------

\_\_fastcall TForm3::TForm3(TComponent\* Owner)

 : TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button1Click(TObject \*Sender)

{

h=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

v=(pow(2.7182818281828182,d/h)-(pow(2.7182818281828182,-d/h)));

l=v\*h;

Edit3->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button2Click(TObject \*Sender)

{

g=Edit4->Text.ToDouble();

t=g\*l;

Edit5->Text=AnsiString(t);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::RadioButton1Click(TObject \*Sender)

{

if(RadioButton1->Checked==true){

 l=l\*1000;

 }

Edit3->Text=AnsiString(l);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::RadioButton2Click(TObject \*Sender)

{

if(RadioButton2->Checked==true){

 l=l;

 }

Edit3->Text=AnsiString(l); }

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button3Click(TObject \*Sender)

{

h=Edit1->Text.ToDouble();

m=h/2;

Edit6->Text=AnsiString(m);

Edit7->Text=AnsiString(h);

Edit8->Text=AnsiString(h);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button4Click(TObject \*Sender)

{

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm3::Button5Click(TObject \*Sender)

{Close(); }

