APLICACIONES DE LA ESTATICA EN EL DISEÑO DE PUENTES PEATONALES EN EL MEDIO RURAL

**INTRODUCCION**

El puente peatonal como construcción permite el paso de peatones sobre cuerpos de agua, vías de tráfico o valles en las montañas. Se pueden construir en diferentes tipos de materiales. Los hay estáticos y móviles (se pliegan, gira o elevan), los tamaños son muy diversos. Debido a la poca carga para la que están concebidos y a la limitada longitud que han de atravesar, el diseños de los mismos pueden ser muy diverso.

Desde el punto de vista de planificación de transporte la gran ventaja de estas estructuras es que no dificultan el tráfico. Este tipo de estructuras alargan el camino con respecto a un paso de cebra o con un semáforo.

Para el diseño de puentes peatonales en el medio rural, frecuentemente se utilizan cables, ya que estos poseen la pecualiaridad de ser utilizados en obras de ingeniería como estructuras. Los cables están hechos de materiales metalicos, fibras vegetales o fibras sinteticas siendo el uso mas ferecuente, los cables de metal.

Entre algunas propiedades mecanicas que tienen los cables, es que son flexibles e inextensibles, osea que no son capaces de resistir flexion y no se alargan, en el estudio del equilibrio de un cable se consideran los principios de la estatica.

En el medio rural la mayor parte de los puentes peatonales construidos son puentes colgantes debido a los costos de los materiales y que para el diseño de estos se tomara en cuenta según el lugar especifico de construcion, las condiciones climatológicas, condiciones especiales, además se tendrá presente la magnitud de las cargas uniformemente repartidas.

CABLES PARABOLICOS Y ELL PUENTE COLGANTE.

Un puente colgante es aquella cuya calzada esta sostenida por muchos suspensores verticales. Los suspensores a su vez, se encuentra suspendidos mendiante cables gruesos flexibles de acero que se anclan en cada uno de los extremos del puente. Si la calzada horizontal se sujeta a una carga uniformemente distribuida debida a la gravedad (peso), también se supone que la carga sobre los cables esta uniformemente distribuida en sentido horizontal.

El espaciamiento entre los suspensores suelen ser pequeños en comparación con la distancia entre los apoyos. Asi mismo el peso de la calzada es muchas veces mayor que el peso de los cables y los suspensores que transmiten las cargas hacia aquellos. Por tanto en un análisis inicial de ese tipo de aplicaciones, se desprecian los pesos de las cables y los suspensores.

El análisis de los cables se restringirá a aquellas que se encuentran suspendidos entre dos apoyos tembien llamados anclajes. La distancia entre los apoyos se llaman claro y la vertical desde un apoyo hasta el punto mas bajo del cable se llama flecha o deflexión, en general los apoyos puenden encontrarse a elevaciones diferentes y por tanto la fecha en un cable se puede medir con relación al apoyo izquierdo o al derecho.

APOYOS A DIFERENTES NIVELES

Considere un cable con apoyos A y B y con el claro horizontal a, el cable soporta una carga de la calzada uniformemente distribuida de w (fuerza por la unidad de la calzada). Se desprecia el alargamiento y el peso de cable. Con el fin de determinar las expresión para la tensión T en cualquier punto del cable.

Por lo tanto:

O bien,

Asi que:

Asi mismo ya que las fuerzas son concurrentes, por consideraciones geométricas.

La ecuación representa una parábola cuyo vértice esta en C y cuyo eje es vertical. Se eleva al cuadrado y se suman las ecuaciones (2) y se toma la raíz cuadrada del resultado se obtiene:

Formula para la tensión minima:

Formula de la tensión el cualquier punto del cable:

Formula para la tensión máxima:

APOYOS AL MISMO NIVEL:

Si los apoyos están al mismo nivel d es la flecha del punto medio C del cable. Del mismo modo donde a es el claro, entonces la ecuación para la tensión minima en el cable es:

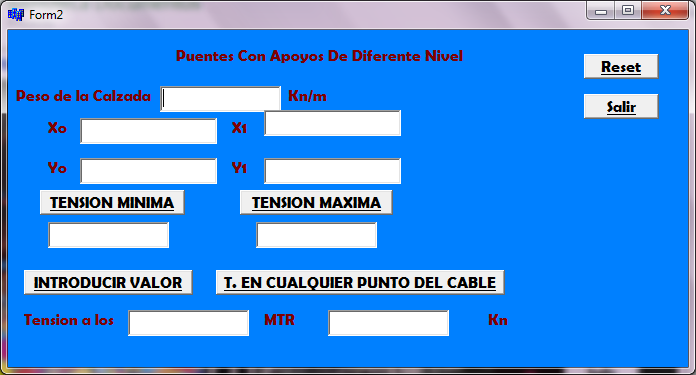
Por las ecuaciones anteriores se deducen que la tensión T en cualquier punto del cable, en x, esta dada por:

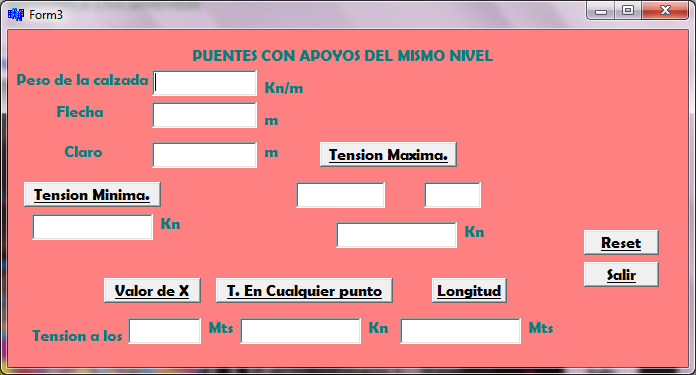
Formula para calcular la tensión máxima:

La ecuación para calcular la longitud es:

Esta ecuación solo es válida para cables cuyos apoyos están en el mismo nivel.

PROGRAMA





//Botones Del Primer Form1

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

#include "Unit3.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

Form2->Show();

}

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

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Form3->Show();

}

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

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

Close();

//Form2

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

double w,xo,yo,x1,y1,h,t,s,q,m,n;

int x=1;

TForm2 \*Form2;

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

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

: TForm(Owner)

{

}

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

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

{

w=Edit1->Text.ToDouble();

xo=Edit2->Text.ToDouble();

yo=Edit4->Text.ToDouble();

h=w\*(xo\*xo)/(2\*yo);

Edit6->Text=AnsiString(h);

}

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

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

{

Close();

}

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

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

{

w=Edit1->Text.ToDouble();

xo=Edit2->Text.ToDouble();

yo=Edit4->Text.ToDouble();

x1=Edit3->Text.ToDouble();

y1=Edit5->Text.ToDouble();

s=(1+(x1\*x1)/(4\*y1\*y1));

t=w\*x1\*pow(s,0.5);

Edit7->Text=AnsiString(t);

}

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

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

{

w=Edit1->Text.ToDouble();

x1=Edit3->Text.ToDouble();

y1=Edit5->Text.ToDouble();

q=Edit8->Text.ToDouble();

m=(q\*q)+(x1\*x1\*x1\*x1)/(4\*y1\*y1);

n=w\*pow(m,0.5);

Edit9->Text=AnsiString(n);

}

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

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

{

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

Edit9->Text="";

Edit1->SetFocus();

}

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

//Form 3

#include <vcl.h>

#pragma hdrstop

#include "Unit3.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

double m,d,a,n,s,p,q,lon,w,c,b,v,z,f,g,h,i,l,j,k,di,su,e;

int x;

TForm3 \*Form3;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm3::Button7Click(TObject \*Sender)

{

Close();

}

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

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

{

w=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

a=Edit3->Text.ToDouble();

h=(w\*a\*a)/(8\*d);

Edit4->Text=AnsiString(h);

}

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

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

{

w=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

a=Edit3->Text.ToDouble();

di=Edit5->Text.ToDouble();

su=Edit6->Text.ToDouble();

e=a\*a;

f=d\*d;

g=e/f;

h=di\*g;

i=su+h;

l=pow(i,0.5);

j=w\*a/2;

k=j\*l;

Edit7->Text=AnsiString(k);

}

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

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

{

w=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

a=Edit3->Text.ToDouble();

z=Edit8->Text.ToDouble();

v=(a\*a\*a\*a)/(64\*d\*d);

b=(z\*z)+v;

c=w\*pow(b,0.5);

Edit9->Text=AnsiString(c);

}

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

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

{

w=Edit1->Text.ToDouble();

d=Edit2->Text.ToDouble();

a=Edit3->Text.ToDouble();

m=(d/a)\*(d/a);

n=(d/a)\*(d/a)\*(d/a)\*(d/a);

s=(8/3);

p=(32/5);

q=1+s\*m-p\*n;

lon=a\*q;

Edit10->Text=AnsiString(lon);

}

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

void \_\_fastcall TForm3::Button6Click(TObject \*Sender)

{

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

Edit9->Text="";

Edit10->Text="";

Edit1->SetFocus();

}

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