Universidad Autónoma de Chiapas

Facultad de ingeniería

Programación por computadoras

Catedrático: Dr. Rubén Herrera

Galicia

Tema: “Pandeo”

Alumno: Balcázar Rincón

Grado y grupo: 2° “A”

BALCAZAR RINCON

FIC/UNACH

PROGRAMACION

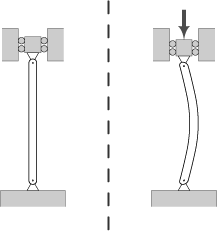
PROYECTO INDIVIDUAL

**INTRODUCCIÓN**

A continuación se mostrara una pequeña investigación sobre lo que es el pandeo y los tipos del mismo y de manera adjunta ira el programa que se realizó para poder calcular el pandeo en una columna dependiendo del material elegido, este programa pretende facilitar todo lo que es el proceso de cálculo. Se mostrara la programación que se llevó a cado utilizando el programa C++ Builder utilizando el lenguaje de programación C++.

**Pandeo**

El pandeo es un fenómeno de inestabilidad elástica que puede darse en elementos comprimidos esbeltos, y que se manifiesta por la aparición de desplazamientos importantes transversales a la dirección principal de compresión.

La aparición de deflexión por pandeo limita severamente la resistencia en compresión de un pilar o cualquier tipo de pieza esbelta.

Eventualmente, a partir de cierto valor de la carga axial de compresión, denominada carga crítica de pandeo, puede producirse una situación de inestabilidad elástica y entonces fácilmente la deformación aumentara produciendo tenciones adicionales que superarán la tensión de rotura, provocando la ruina del elemento estructural. Además del pandeo flexional ordinario existe el pandeo torsional o inestabilidad elástica provocando por un momento torsor excesivo.

Existen diferentes maneras o modos de fallo por pandeo. Para un elemento estructural frecuentemente hay que verificar varios de ellos y garantizar que las cargas están lejos de las cargas críticas asociadas a cada modo o manera de pandear.

**Los modos típicos son:**

* **PANDEO FLEXIONAL.** Modo de pandeo en el cual un elemento en compresión se flecta lateralmente sin giro ni cambios en su sección transversal.
* **PANDEO TORSIONAL.** Modo de pandeo en el cual un elemento en compresión gira alrededor de su centro de corte.
* **PANDEO FLEXO-TORSIONAL**. Modo de pandeo en el cual un elemento en compresión se flecta y se gira simultáneamente sin cambios en su sección transversal.
* **PANDEO LATERAL-TORSIONAL.** Modo de pandeo de un elemento a flexión que involucra deflexión normal al plano de flexión, y de, manera simultánea, giro alrededor de su centro de corte.

**PANDEO FLEXIONAL.**

Los pilare y barras comprimidas de celosías pueden presentar diversos modos de fallo en función de su esbeltez mecánica:

* Los pilares muy esbeltos pueden fallar por pandeo elástico y son sensibles tanto al pandeo local como al pandeo global de la estructura completa.
* En los pilares de esbeltez media las imperfecciones constructivas como las heterogeneidades son particularmente importantes pudiéndose presentar pandeo anelástico.
* Los pilares de muy bajo esbeltez fallan por exceso de compresión, antes de que los efectos del pandeo resulten importantes.

**PANDEO LOCAL.**

El pandeo local es el que aparece en piezas o elementos aislados que estructuralmente pueden considerarse aislados. En este caso la magnitud de la carga critica viene dada según el caso por la fórmula de Leonhard Euler o la de Engesser. La carga critica de Euler depende de la longitud de la pieza, del material, de su sección transversal y de las conexiones de unión, vinculación o sujeción en los extremos. Para una pieza que puede considerarse biarticulada en sus extremos, la carga crítica de Euler viene dada por:

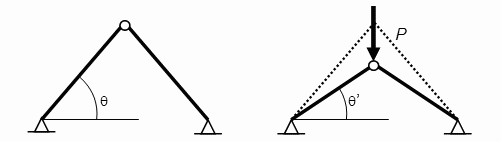
Siendo:

Cuando la sujeción de los extremos es diferentes, la carga crítica de Euler viene dada por una ecuación del tipo:

* Al producto se le llama longitud de pandeo.

**PANDEO GLOBAL.**

En una estructura compleja formada por barras y otros elementos enlazados, puede aparecer modo de deformación en los que los desplazamientos no sean proporcionales a las cargas, y la estructura puede pandear globalmente sin que ninguna de las barras o elementos estructurales alcance su propia carga de pandeo. Debido a este factor, la carga crítica global de cierto tipo de estructura (por ejemplo en entramados de cúpulas o monocapa) es mucho menor que la carga crítica (local) de cada uno de sus elementos.

El tipo de estructura que presenta pandeo global para carga crítica diferente de las de sus elementos está formado por dos barras articuladas entre sí y la cimentación, como se muestra en la figura.

Las ecuaciones que gobiernan el comportamiento de la estructura son:

* Ecuación de equilibrio:
* Relación elástica entre acortamiento y esfuerzo axial:
* Relación geométrica de las configuraciones deformada y no deformada:

Donde

* N = Esfuerzo axial de cada una de las barras.
* Acortamiento sufrido por las barras para adoptar la configuración deformada.
* , Es la diferencia de ángulos mostrados en la figura.
* Módulo de Young del material de las barras.
* Area transversal de cada una de las barras.
* Longitud inicial de cada una de las barras.

Sustituyendo la segunda de las ecuaciones en la primera, despejando de la tercera y sustityendo su valor también en la primera se llega a:

El valor para que se alcance el máximo es precisamente la carga crítica global. Las cargas de pandeo global y local vienen dadas por:

**PLANO DE PANDEO:**

El plano de pandeo se refiere al plano que contendrá el inicio de la deformada de una pieza sometida con presión dominante, el plano de pandeo contiene el eje baricentro de la viga y sobre el deflexión por pandeo es máxima. Para una pieza sometida solo a compresión sin momentos apreciables adicionales, el plano de pandeo coincidirá con el plano perpendicular, sea paralela al eje de menor inercia de la sección.

**PANDEO TORCIONAL:**

En viga de alas anchas o de escasa rigidez torsional el pandeo flexional convencional puede ir acompañado de la aparición de una torsión de la sección, resultado un modo de fallo mixto conocido como pandeo torsional o pandeo lateral. El momento torsor crítico para el cual aparecería este tipo de fallo viene dado por:

Donde las nuevas magnitudes son:

* Momento de inercia mínimo en flexión.
* Módulo de alabeo.
* Módulo de torsión.
* Módulo de elasticidad transversal.

El resto de magnitudes tiene el mismo significado que para el de pandeo flexional puro. En piezas donde el momento de alabeo es despreciable puede usarse la expresión aproximada:

**FORM1**



**Programacion**

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

TForm1 \*Form2;

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

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

: TForm(Owner)

{

}

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

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

{

Close();

}

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

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

{

Form2->Show();

Form1->Visible=false;

}

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

**Form2**



**Programación**

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

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

double a,ax,an,aux,d,ma,me,mi,l,c,e,crit,H2,arc,t,p,kj,jk,f,u,v,gf,tu,eq,an2,l2;

double da,rs,op,res,mt,mot,M,Ma,o,P,s,b,Jc,Jo,r,j,k,m,ar;

TForm2 \*Form2;

TForm2 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

if(RadioButton4->Checked==true)

{

Edit1->Text="29.0";

}

if(RadioButton5->Checked==true)

{

Edit1->Text="22.1";

}

if(RadioButton6->Checked==true)

{

Edit1->Text="8";

}

if(RadioButton7->Checked==true)

{

Edit1->Text="200";

}

if(RadioButton8->Checked==true)

{

Edit1->Text="23";

}

if(RadioButton9->Checked==true)

{

Edit1->Text="50";

}

}

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

void \_\_fastcall TForm2::Button12Click(TObject \*Sender)

{

Close();

}

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

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

{

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

p=Edit8->Text.ToDouble();

an2=Edit10->Text.ToDouble();

if (RadioButton2->Checked==true)

{

ax=p/ar;

da=an-an2;

rs=an-da;

u=rs\*(M\_PI/180);

tu=sin(u);

eq=2\*ax\*tu;

Edit11->Text=AnsiString(eq);

}

}

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

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

{

a=Edit3->Text.ToDouble();

b=Edit16->Text.ToDouble();

f=a\*b;

Edit5->Text=AnsiString(f);

}

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

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

{

a=Edit3->Text.ToDouble();

b=Edit16->Text.ToDouble();

r=Edit6->Text.ToDouble();

if(RadioButton1->Checked==true)

{

o=a\*b;

p=pow(a,2);

P=pow(b,2);

l=p+P;

s=o\*l;

Jc=s/12;

Edit2->Text=AnsiString(Jc);

}

if(RadioButton3->Checked==true)

{

o=pow(r,4);

e=(M\_PI)\*o;

Jo=(e/2);

Edit2->Text=AnsiString(Jo);

}

}

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

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

{

me=Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

if(RadioButton1->Checked==true)

{

crit=(pow((M\_PI),2)\*me\*mi)/(l\*l);

Edit7->Text=AnsiString(crit);

}

if(RadioButton2->Checked==true)

{

crit=0;

t=2;

f=3;

d=t/f;

u=an\*(M\_PI/180);

c=cos(u);

a=2\*me\*ar;

gf=(pow(c,d));

e=1-gf;crit=a\*e;

Edit7->Text=AnsiString(crit);

}

if(RadioButton3->Checked==true)

{

crit=0;

jk=me\*mi\*(pow((M\_PI),2));

kj=pow((2\*l),2);

crit=jk/kj;

Edit7->Text=AnsiString(crit);

}

}

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

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

{

ar=Edit5->Text.ToDouble();

p=Edit8->Text.ToDouble();

if (RadioButton1->Checked==true)

{

ax=p/ar;

Edit9->Text=AnsiString(ax);

}

if (RadioButton2->Checked==true)

{

ax=p/ar;

Edit9->Text=AnsiString(ax);

}

}

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

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

{

l= Edit3->Text.ToDouble();

mi= Edit2->Text.ToDouble();

l= Edit3->Text.ToDouble();

H2=pow(1,2);

Ma=(H2\*mi)/4;

Edit12->Text=AnsiString(Ma);

}

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

void \_\_fastcall TForm2::Button8Click(TObject \*Sender)

{

a=Edit5->Text.ToDouble();

k=38;

p=Edit8->Text.ToDouble();

l=pow(a,4);

m=k\*p;

j=l/m;

Edit14->Text=AnsiString(j);

}

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

void \_\_fastcall TForm2::Button9Click(TObject \*Sender)

{

me=Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

ma=Edit12->Text.ToDouble();

mt=Edit14->Text.ToDouble();

if( RadioButton3->Checked==true)

{

a=((M\_PI)\*me\*mi)/l;

ax=pow(a,2);

c=ax\*ma\*mi;

t=c+(me\*mi\*me\*mt);

p=pow(t,0.5);

aux=(M\_PI)/l;

mot=aux\*p;

Edit13->Text=AnsiString(mot);

}

}

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

void \_\_fastcall TForm2::Button10Click(TObject \*Sender)

{

me= Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

if(RadioButton1->Checked==true)

{

op= (M\_PI)\*me\*mi;

res=op/(l\*l\*ar);

Edit15->Text=AnsiString(res);

}

if(RadioButton2->Checked==true)

{

t=2;

f=3;

d=t/f;

u=an\*(M\_PI/180);

c=cos(u);

a=2\*me;

gf=(pow(c,d));

e=l-gf;

crit=a\*e;

Edit15->Text=AnsiString(res);

}

if(RadioButton3->Checked==true)

{

jk=me\*mi\*pow((M\_PI),2);

kj=pow((2\*l),2);

res=kj/jk\*ar;

Edit15->Text=AnsiString(res);

}

}

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

void \_\_fastcall TForm2::Button11Click(TObject \*Sender)

{

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

Edit9->Text="";

Edit10->Text="";

Edit11->Text="";

Edit12->Text="";

Edit13->Text="";

Edit14->Text="";

Edit15->Text="";

Edit16->Text="";

}

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

**PROGRAMACIÓN PASO POR PASO**

1. Cargar C++
2. Se guarda el archivo “pandeo.bpr”.
3. En el form1 se pone lo siguiente:

* Button 1
* Button 2

1. Se programa button1:

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

{

Form2->Show();

Form1->Visible=false;

}

1. Porgrama button2:

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

{

Close();

}

**PROGRAMACION PASO POR PASO FORM2**

1. Se colocan:

* 9 RadioButton
* 16 Edit
* 11 Button

1. Se programa button1:

if(RadioButton4->Checked==true)

{

Edit1->Text="29.0";

}

if(RadioButton5->Checked==true)

{

Edit1->Text="22.1";

}

if(RadioButton6->Checked==true)

{

Edit1->Text="8";

}

if(RadioButton7->Checked==true)

{

Edit1->Text="200";

}

if(RadioButton8->Checked==true)

{

Edit1->Text="23";

}

if(RadioButton9->Checked==true)

{

Edit1->Text="50";

}

}

1. Programa button2:

{

a=Edit3->Text.ToDouble();

b=Edit16->Text.ToDouble();

f=a\*b;

Edit5->Text=AnsiString(f);

}

1. Programa button3:

a=Edit3->Text.ToDouble();

b=Edit16->Text.ToDouble();

r=Edit6->Text.ToDouble();

if(RadioButton1->Checked==true)

{

o=a\*b;

p=pow(a,2);

P=pow(b,2);

l=p+P;

s=o\*l;

Jc=s/12;

Edit2->Text=AnsiString(Jc);

}

if(RadioButton3->Checked==true)

{

o=pow(r,4);

e=(M\_PI)\*o;

Jo=(e/2);

Edit2->Text=AnsiString(Jo);

}

1. Programacion button4:

me=Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

if(RadioButton1->Checked==true)

{

crit=(pow((M\_PI),2)\*me\*mi)/(l\*l);

Edit7->Text=AnsiString(crit);

}

if(RadioButton2->Checked==true)

{

crit=0;

t=2;

f=3;

d=t/f;

u=an\*(M\_PI/180);

c=cos(u);

a=2\*me\*ar;

gf=(pow(c,d));

e=1-gf;crit=a\*e;

Edit7->Text=AnsiString(crit);

}

if(RadioButton3->Checked==true)

{

crit=0;

jk=me\*mi\*(pow((M\_PI),2));

kj=pow((2\*l),2);

crit=jk/kj;

Edit7->Text=AnsiString(crit);

}

1. Programacion button5:

ar=Edit5->Text.ToDouble();

p=Edit8->Text.ToDouble();

if (RadioButton1->Checked==true)

{

ax=p/ar;

Edit9->Text=AnsiString(ax);

}

if (RadioButton2->Checked==true)

{

ax=p/ar;

Edit9->Text=AnsiString(ax);

}

1. Programación button6:

{

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

p=Edit8->Text.ToDouble();

an2=Edit10->Text.ToDouble();

if (RadioButton2->Checked==true)

{

ax=p/ar;

da=an-an2;

rs=an-da;

u=rs\*(M\_PI/180);

tu=sin(u);

eq=2\*ax\*tu;

Edit11->Text=AnsiString(eq);

}

1. Programación button7:

l= Edit3->Text.ToDouble();

mi= Edit2->Text.ToDouble();

l= Edit3->Text.ToDouble();

H2=pow(1,2);

Ma=(H2\*mi)/4;

Edit12->Text=AnsiString(Ma);

1. Programacion button8:

a=Edit5->Text.ToDouble();

k=38;

p=Edit8->Text.ToDouble();

l=pow(a,4);

m=k\*p;

j=l/m;

Edit14->Text=AnsiString(j);

1. Programacion button9:

me=Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

ma=Edit12->Text.ToDouble();

mt=Edit14->Text.ToDouble();

if( RadioButton3->Checked==true)

{

a=((M\_PI)\*me\*mi)/l;

ax=pow(a,2);

c=ax\*ma\*mi;

t=c+(me\*mi\*me\*mt);

p=pow(t,0.5);

aux=(M\_PI)/l;

mot=aux\*p;

Edit13->Text=AnsiString(mot);

}

1. Programacion button10:

me= Edit1->Text.ToDouble();

mi=Edit2->Text.ToDouble();

l=Edit3->Text.ToDouble();

an=Edit4->Text.ToDouble();

ar=Edit5->Text.ToDouble();

if(RadioButton1->Checked==true)

{

op= (M\_PI)\*me\*mi;

res=op/(l\*l\*ar);

Edit15->Text=AnsiString(res);

}

if(RadioButton2->Checked==true)

{

t=2;

f=3;

d=t/f;

u=an\*(M\_PI/180);

c=cos(u);

a=2\*me;

gf=(pow(c,d));

e=l-gf;

crit=a\*e;

Edit15->Text=AnsiString(res);

}

if(RadioButton3->Checked==true)

{

jk=me\*mi\*pow((M\_PI),2);

kj=pow((2\*l),2);

res=kj/jk\*ar;

Edit15->Text=AnsiString(res);

}

1. Programacion button11:

Edit1->Text="";

Edit2->Text="";

Edit3->Text="";

Edit4->Text="";

Edit5->Text="";

Edit6->Text="";

Edit7->Text="";

Edit8->Text="";

Edit9->Text="";

Edit10->Text="";

Edit11->Text="";

Edit12->Text="";

Edit13->Text="";

Edit14->Text="";

Edit15->Text="";

Edit16->Text="";

1. Programacion button12:

Close();

1. A continuación se declaran las variables:

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

#include <math.h>

double a,ax,an,aux,d,ma,me,mi,l,c,e,crit,H2,arc,t,p,kj,jk,f,u,v,gf,tu,eq,an2,l2;

double da,rs,op,res,mt,mot,M,Ma,o,P,s,b,Jc,Jo,r,j,k,m,ar;

TForm2 \*Form2;

TForm2 \*Form1;