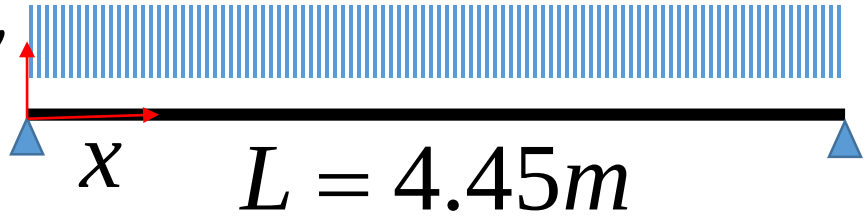
Deflección de una viga

Problema matemático

Se quiere calcular la deflección máxima de una viga simplemente apoyado con un estado de carga constante



$$q = 1960 N / m$$

La ecuación diferencial de la deflección (y) es:

$$\frac{d^2y}{dx^2} + \frac{q}{2EI}\left(x^2 - Lx\right) = 0$$

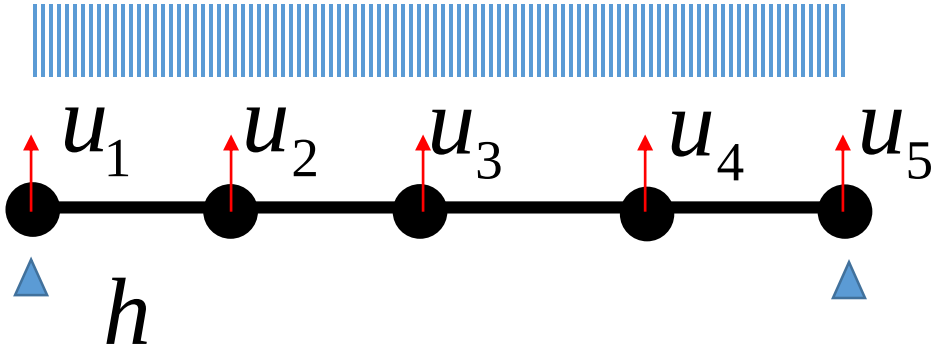
Acero $E = 200GPa$

Perfil IPN 140 $I = 573cm^4$

$$\alpha = \frac{q}{2EI}$$

Problema numérico

Se verá en la última unidad de la materia que se puede tranformar el problema diferencial en un Sistema de Ecuaciones Lineales aplicando el Método de las Diferencias Finitas



Cantidad de nodos

$$n = 5$$

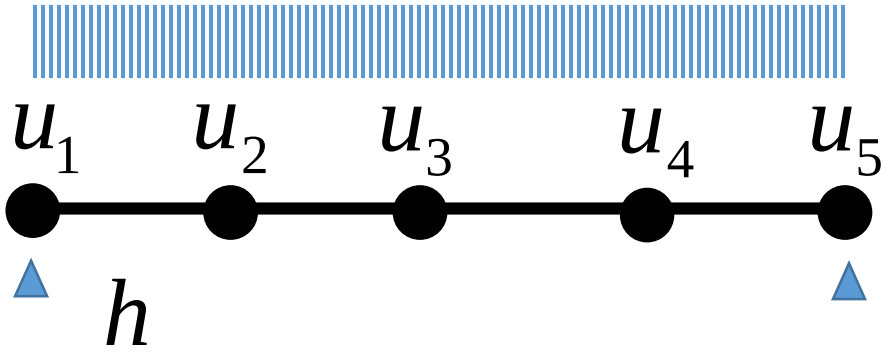
Paso (distancia entre nodos)

$$h = L/(n-1)$$

Solo se obtiene la solución en los nodos

Para indicar que es una solución discrete la llamamos u en lugar de y

Problema numérico



Cantidad de nodos

$$n = 5$$

Paso (distancia entre nodos)

$$h = L/(n-1)$$

Ecuación para el primer nodo

$$u_1 = 0$$

Ecuación para los nodos intermedios: i = 1, 2, 3, 4

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \alpha\left(\left(i-1\right)\cdot h\right)\cdot\left(\left(i-1\right)\cdot h - L\right) = 0$$

Ecuación para el ultimo nodo

$$u_n = 0$$

Armado de la matriz

viga.m

```
1  # Parámetros del problema matemático
2  L = 4.45;
3  alpha = 1960 / 2 / 200e9 / (573 / 100**4);
4
5  # Parámetros del problema numérico
6  n = 5;
7  h = L / (n - 1);
8
9  # Armado de la matriz y el vector independiente
10 A = zeros(n, n);
11 b = zeros(n, 1);
12
13 # Armado de primera y última fila
14 A(1,1) = 1;
15 A(n,n) = 1;
16 # Armado de filas intermedias
17 for i = 2:n-1
18     A(i,i-1) = 1/h**2;
19     A(i,i) = -2/h**2;
20     A(i,i+1) = 1/h**2;
21     b(i) = -alpha * ((i-1) * h) * ((i-1) * h - L);
22 endfor
23
24 disp("A es:"); disp(A);
25 disp("B es:"); disp(b);
```

Armado de la matriz

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
>> viga

A es:
   1.00000   0.00000   0.00000   0.00000   0.00000
   0.80798  -1.61596   0.80798   0.00000   0.00000
   0.00000   0.80798  -1.61596   0.80798   0.00000
   0.00000   0.00000   0.80798  -1.61596   0.80798
   0.00000   0.00000   0.00000   0.00000   1.00000
B es:
   0.0000000
   0.0031751
   0.0042335
   0.0031751
   0.0000000
```

Resolución por Jacobi

Resolveremos ahora el Sistema de Ecuaciones Lineales por el método de Jacobi

1- Adopto una condición inicial

$$X^0 \qquad x_i^0$$

2- Repito varias veces el cálculo iterativo

$$x_i^{k+1} = {}^{1}\!/_{a_{i,i}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{i,j} x_j^k \right)$$

Resolución del sistema

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
28  # Resolución por método de Jacobi
29  # Creo el vector Xk donde guardaré X^k
30  Xk = zeros(n, 1);
31  # Creo el vector Xkmas1 donde guardaré X^k+1
32  Xkmas1 = zeros(n, 1);
33  tolerancia = 1e-5; # Centésima de milímetro
34  errorMax = 10000; # Inicializo el error en un valor grande
35  it = 1;
36
37  # Condición inicial se guarda en X^k
38  # Por ahora probemos dejar todos ceros
39
40  while errorMax > tolerancia
41
42      # Calculo X^k+1
43      for i = 1:n
44          Xkmas1(i) = b(i) / A(i, i);
45          for j = 1:n
46              if j == i
47                  continue
48              endif
49              Xkmas1(i) = Xkmas1(i) - A(i, j) * Xk(j) / A(i, i);
50          endfor
51      endfor
52
53      # Cálculo del error
54      errorMax = 0;
55      for i = 1:n
56          error = abs(Xkmas1(i) - Xk(i));
57          if error > errorMax
58              errorMax = error;
59          endif
60      endfor
61
62      printf("Iteracion %d, error %f\n", it, errorMax);
63      it = it + 1;
64
65      # Sobrescribo el Xk con el nuevo valor
66      Xk = Xkmas1;
67  endwhile
68
69  disp("La solucion final es:"); disp(Xkmas1);
70  disp("La deflección máxima es:"); disp(max(abs(Xkmas1)));
71
```

Criterio de corte en función del error absoluto

Resolución del sistema

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
Iteracion 1, error 0.002620
Iteracion 2, error 0.001965
Iteracion 3, error 0.001310
Iteracion 4, error 0.000982
Iteracion 5, error 0.000655
Iteracion 6, error 0.000491
Iteracion 7, error 0.000327
Iteracion 8, error 0.000246
Iteracion 9, error 0.000164
Iteracion 10, error 0.000123
Iteracion 11, error 0.000082
Iteracion 12, error 0.000061
Iteracion 13, error 0.000041
Iteracion 14, error 0.000031
Iteracion 15, error 0.000020
Iteracion 16, error 0.000015
Iteracion 17, error 0.000010
Iteracion 18, error 0.000008
La solucion final es:
   0.0000000
  -0.0065368
  -0.0091515
  -0.0065368
   0.0000000
La deflexion mxima es:
 0.0091515
```

19 iteraciones

= 9.2 mm

Convergencia

- ¿Este problema hubiera convergido para cualquier condición inicial?
- Dos formas de garantizarlo:
  - Si la matriz es estrictamente diagonal dominante (no lo es)
  - Radio espectral de la matriz de iteración menor que 1

Radio espectral


```
72  # Cálculo de la matriz de iteración
73  L = zeros(n,n);
74  U = zeros(n,n);
75  D = zeros(n,n);
76  for i = 1:n
77      for j = 1:n
78          if i == j
79              D(i, j) = A(i, j);
80          elseif i > j
81              U(i, j) = -A(i, j);
82          else
83              L(i, j) = -A(i, j);
84          endif
85      endfor
86  endfor
87  T = inv(D)*(L+U);
88
89  # Cálculo de los autovalores
90  autovalores = eig(T);
91  disp("Autovalores:"); disp(autovalores);
92  # Cálculo del radio espectral
93  radio = max(abs(autovalores));
94  disp("Radio Espectral:"); disp(radio);
```

Radio espectral

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
Autovalores:
  -0.70711
   0.00000
   0.70711
   0.00000
   0.00000
Radio Espectral:
  0.70711
```

< 1

Entonces efectivamente converge para cualquier X0

Resolución por Gauss-Seidel

Resolveremos ahora el Sistema de Ecuaciones Lineales por el método de Gauss-Seidel

1- Adopto una condición inicial

$$X^0 \qquad x_i^0$$

2- Repito varias veces el cálculo iterativo

$$x_i^{k+1} = {}^{1}\!/_{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} - \sum_{j=i+1}^{n} a_{i,j} x_j^k \right)$$

Resolución del sistema

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
28  # Resolución por método de Jacobi
29  # Creo el vector Xk donde guardaré X^k
30  Xk = zeros(n, 1);
31  # Creo el vector Xkmas1 donde guardaré X^k+1
32  Xkmas1 = zeros(n, 1);
33  tolerancia = 1e-5; # Centésima de milímetro
34  errorMax = 10000; # Inicializo el error en un valor grande
35  it = 1;
36
37  # Condición inicial se guarda en X^k
38  # Por ahora probemos dejar todos ceros
39
40  while errorMax > tolerancia
41
42      # Calculo X^k+1
43      for i = 1:n
44          Xkmas1(i) = b(i) / A(i, i);
45          for j = 1:i-1
46              Xkmas1(i) = Xkmas1(i) - A(i, j) * Xkmas1(j) / A(i, i);
47          endfor
48          for j = i+1:n
49              Xkmas1(i) = Xkmas1(i) - A(i, j) * Xk(j) / A(i, i);
50          endfor
51      endfor
52
53      # Cálculo del error
54      errorMax = 0;
55      for i = 1:n
56          error = abs(Xkmas1(i) - Xk(i));
57          if error > errorMax
58              errorMax = error;
59          endif
60      endfor
61
62      printf("Iteracion %d, error %f\n", it, errorMax);
63      it = it + 1;
64
65      # Sobrescribo el Xk con el nuevo valor
66      Xk = Xkmas1;
67  endwhile
68
69  disp("La solucion final es:"); disp(Xkmas1);
70  disp("La defleccion máxima es:"); disp(max(abs(Xkmas1)));
```

Solo se modifica esto

Resolución del sistema

```
Iteracion 1, error 0.003766
Iteracion 2, error 0.002784
Iteracion 3, error 0.001392
Iteracion 4, error 0.000696
Iteracion 5, error 0.000348
Iteracion 6, error 0.000174
Iteracion 7, error 0.000087
Iteracion 8, error 0.000043
Iteracion 9, error 0.000022
Iteracion 10, error 0.000011
Iteracion 11, error 0.000005
La solucion final es:
   0.0000000
  -0.0065441
  -0.0091639
  -0.0065468
   0.0000000
La deflexion mxima es:
  0.0091639
```

11 iteraciones

= 9.2 mm

Es decir, llegamos al mismo resultado con cerca de la mitad de operaciones

Radio espectral


```
72  # Cálculo de la matriz de iteración
73  L = zeros(n,n);
74  U = zeros(n,n);
75  D = zeros(n,n);
76  for i = 1:n
77      for j = 1:n
78          if i == j
79              D(i, j) = A(i, j);
80          elseif i > j
81              U(i, j) = -A(i, j);
82          else
83              L(i, j) = -A(i, j);
84          endif
85      endfor
86  endfor
87  T = inv(D-L)*(U);
88
89  # Cálculo de los autovalores
90  autovalores = eig(T);
91  disp("Autovalores:"); disp(autovalores);
92  # Cálculo del radio espectral
93  radio = max(abs(autovalores));
94  disp("Radio Espectral:"); disp(radio);
```

Radio espectral

FACULTAD DE INGENIERIA
Universidad de Buenos Aires

```
Autovalores:
   0.00000
   0.50000
   0.00000
   0.00000
   0.00000
Radio Espectral:
 0.50000
```

< 1

Entonces también converge para cualquier X0