
Elemento Finito con cuadratura de Gauss y matrices dispersas

Adaptado por José Luis Gómez Muñoz
<http://homepage.cem.itesm.mx/jose.luis.gomez>

Basado en el trabajo de John H. Mathews
<http://math.fullerton.edu/mathews/n2003/GalerkinMod.html>
también en la sección 11.5 “El método de Rayleigh-Ritz” del libro de Richard L. Burden y J. Douglas Faires, “Análisis numérico” 9a. edición, editorial Cengage, págs 696-711
y finalmente también en la sección 6.2 “Galerkin’s method with piecewise polynomials” del libro de Kenneth Eriksson, D. Estep, P. Hansbo & C. Johnson, “Computational Differential Equations” Volumen 1, Cambridge University Press, 1997.

También revisar el capítulo 2 del libro:
J.N. Reddy
An Introduction to the Finite Element Method, 3rd Edition
McGraw-Hill

Se va a calcular una solución aproximada a una ecuación diferencial ordinaria, lineal e inhomogénea, $-\frac{d}{dx} \left(p(x) \frac{d}{dx} y(x) \right) + q(x) y(x) = f(x)$, con las condiciones de frontera $y(a) = y_a$, $y(b) = y_b$, como se muestra a continuación :

In[1]:=

```
Clear[x, y, p, q, f, a, b];  
  
p[x_] := x^2;  
q[x_] := 30;  
f[x_] := -14 x;  
a = 0.0;  
b = 1.0;  
y_a = 0.0;  
y_b = 0.5;  
  
Print[TraditionalForm[-Dx (p[x] Dx y[x]) + q[x] * y[x] == f[x]]];  
Print[TraditionalForm[y[a] == y_a]];  
Print[TraditionalForm[y[b] == y_b]];
```

$$x^2 (-y''(x)) - 2x y'(x) + 30 y(x) = -14x$$

$$y(0.) = 0.$$

$$y(1.) = 0.5$$

Lista de nodos (partición del dominio $a \leq x \leq b$)

In[12]=

```
Clear[n, j, coords];
n = 7;

coords = Table[a + (b - a)  $\frac{\text{Log}[j]}{\text{Log}[n]}$ , {j, 1, n}]
```

Out[14]=

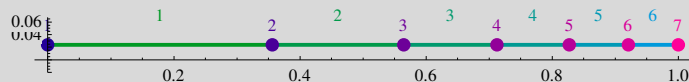
```
{0., 0.356207, 0.564575, 0.712414, 0.827087, 0.920782, 1.}
```

Representación geométrica del mallado del dominio $a \leq x \leq b$, con números para identificar los nodos y los elementos:

In[15]=

```
distancia = (b - a) / (6 * n);
gnodos = {PointSize[Large], Thick,
  Table[{RGBColor[0, 0.6, j / n], Line[
    {{coords[[j]], distancia}, {coords[[j + 1]], distancia}}]}, {j, 1, n - 1}],
  Table[{RGBColor[0, 0.6, j / n], Text[j, { $\frac{\text{coords}[[j]] + \text{coords}[[j + 1]]}{2}$ ,
    distancia}], {0, -3}}]}, {j, 1, n - 1}],
  Table[{RGBColor[j / n, 0, 0.6], Point[{coords[[j]], distancia}]}], {j, 1, n}],
  Table[{RGBColor[j / n, 0, 0.6],
    Text[j, {coords[[j]], distancia}, {0, -2}]}], {j, 1, n]
};
Graphics[gnodos, Axes -> True]
```

Out[17]=



Los nodos “incidentes” de cada elemento:

In[18]=

```
incidentes = Table[{j, j + 1}, {j, 1, n - 1}]
```

Out[18]=

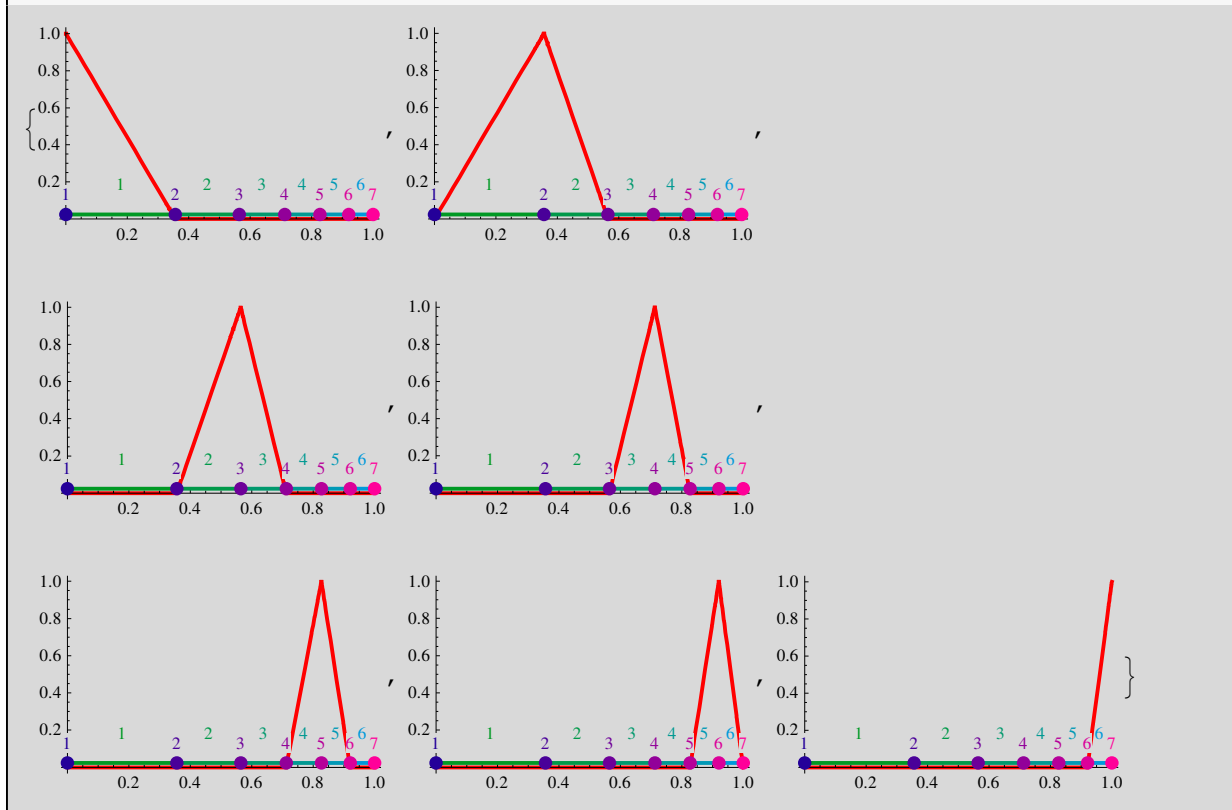
```
{{1, 2}, {2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 7}}
```

La solución aproximada $\Phi(x)$ será una combinación lineal de ciertas funciones linealmente independientes, llamadas funciones base: $\Phi(x) = \sum_{j=1}^n c_j \phi_j(x)$. Las funciones base $\phi_j(x)$ usadas en el método de Elemento Finito tienen soporte local, lo que quiere decir que cada una es distinta de cero sólo en una pequeña parte del dominio, entre tres nodos consecutivos:

In[19]=

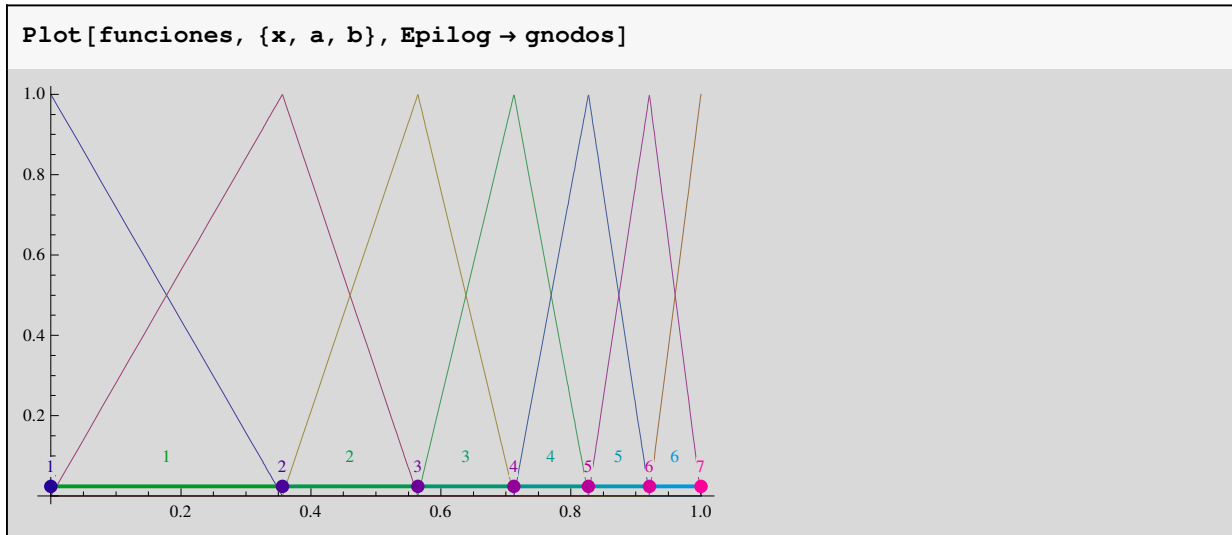
```
Clear[ $\phi$ ,  $x$ ];
 $\phi_1[x_] := \begin{cases} \frac{\text{coords}[[2]]-x}{\text{coords}[[2]]-a} & a < x \leq \text{coords}[[2]]; \\ 0 & \text{True} \end{cases}$ ;
 $\phi_n[x_] := \begin{cases} \frac{x-\text{coords}[[n-1]]}{b-\text{coords}[[n-1]]} & \text{coords}[[n-1]] < x \leq b; \\ 0 & \text{True} \end{cases}$ ;
 $\phi_j[x_] := \begin{cases} \frac{x-\text{coords}[[j-1]]}{\text{coords}[[j]]-\text{coords}[[j-1]]} & \text{coords}[[j-1]] < x \leq \text{coords}[[j]] \\ \frac{\text{coords}[[j+1]]-x}{\text{coords}[[j+1]]-\text{coords}[[j]]} & \text{coords}[[j]] < x \leq \text{coords}[[j+1]]; \\ 0 & \text{True} \end{cases}$ ;
funciones = Table[ $\phi_j[x]$ , {j, 1, n}];
Table[Plot[funciones[[j]], {x, a, b}, PlotRange -> All,
PlotStyle -> {Red, Thick}, Epilog -> gnodes], {j, 1, n}]
```

Out[24]=



A continuación se muestran juntas todas las funciones base:

In[25]=



La solución aproximada tendrá la forma $\Phi(x) = \sum_{j=1}^n c_j \phi_j(x)$. El objetivo será encontrar los coeficientes c_j que den la mejor aproximación a la solución de la ecuación diferencial:

In[26]=

```
Clear[ϕ, c];
```

$$\phi[x_] := \sum_{j=1}^n c_j * \phi_j[x];$$

```
Print["Forma de la aproximación a la solución:"];
```

```
Print["ϕ[x]=", TraditionalForm[ϕ[x]]];
```

Forma de la aproximación a la solución:

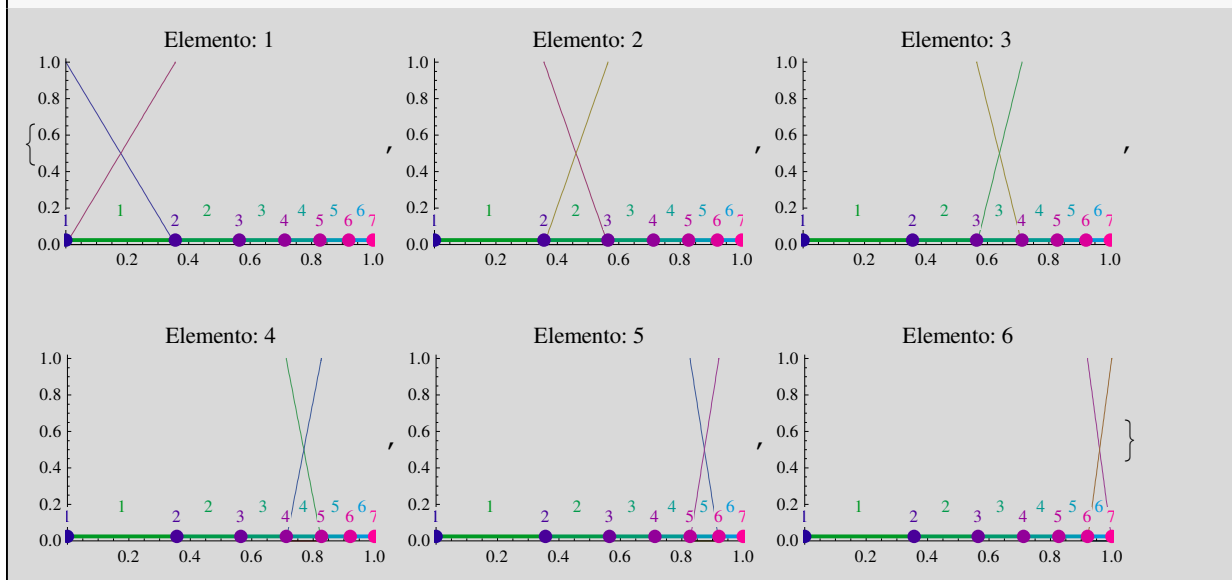
$$\begin{aligned} \phi[x] = & c_1 \left(\begin{cases} 2.80735 (0.356207 - x) & 0. < x \leq 0.356207 \\ 0 & \text{True} \end{cases} \right) + \\ & c_2 \left(\begin{cases} 2.80735 (x + 0.) & 0. < x \leq 0.356207 \\ 4.7992 (0.564575 - x) & 0.356207 < x \leq 0.564575 \end{cases} \right) + \\ & c_3 \left(\begin{cases} 4.7992 (x - 0.356207) & 0.356207 < x \leq 0.564575 \\ 6.7641 (0.712414 - x) & 0.564575 < x \leq 0.712414 \end{cases} \right) + \\ & c_4 \left(\begin{cases} 6.7641 (x - 0.564575) & 0.564575 < x \leq 0.712414 \\ 8.72044 (0.827087 - x) & 0.712414 < x \leq 0.827087 \end{cases} \right) + \\ & c_5 \left(\begin{cases} 8.72044 (x - 0.712414) & 0.712414 < x \leq 0.827087 \\ 10.673 (0.920782 - x) & 0.827087 < x \leq 0.920782 \end{cases} \right) + \\ & c_6 \left(\begin{cases} 10.673 (x - 0.827087) & 0.827087 < x \leq 0.920782 \\ 12.6234 (1. - x) & 0.920782 < x \leq 1. \end{cases} \right) + \\ & c_7 \left(\begin{cases} 12.6234 (x - 0.920782) & 0.920782 < x \leq 1. \\ 0 & \text{True} \end{cases} \right) \end{aligned}$$

En el Método de Elemento Finito (MEF, o FEM por sus siglas en Inglés) el dominio es dividido en elementos. Cada elemento contiene partes de las funciones base que no valen cero dentro del elemento:

In[30]:=

```
Table[
  Plot[funciones,
    {x, coords[[First[incidentes[[j]] ] ]], coords[[Last[incidentes[[j]] ] ]]},
    PlotLabel -> "Elemento: " <> ToString[j],
    PlotRange -> {{a, b}, All}, Epilog -> gnodos],
  {j, 1, n - 1}
]
```

Out[30]=



La ecuación diferencial puede escribirse de la forma:

$$-\frac{d}{dx} \left(p(x) \frac{d}{dx} y(x) \right) + q(x) y(x) - f(x) = 0$$

Si reemplazamos la solución aproximada $\Phi(x)$ en lugar de la solución exacta $y(x)$ en el lado izquierdo de la ecuación anterior, el resultado ya no será exactamente cero. En su lugar obtendremos una función conocida como residuo, $r(x)$:

$$-\frac{d}{dx} \left(p(x) \frac{d}{dx} \Phi(x) \right) + q(x) \Phi(x) - f(x) = r(x)$$

El método de Bubnov-Galerkin consiste en encontrar la combinación lineal

$$\Phi(x) = \sum_{j=1}^n c_j \phi_j(x)$$

cuyo residuo $r(x)$ tenga componente igual a cero en el

subespacio generado por las funciones base. Esto significa que el producto interno del residuo por cada función base debe ser cero. En el Método de Elemento Finito

este cálculo se hace elemento por elemento:

$$\int_{\text{elemento}} \phi_j(\mathbf{x}) r(\mathbf{x}) d\mathbf{x} = 0$$

reemplazando la expresión para $r(\mathbf{x})$ en el integrando obtenemos:

$$-\int_{\text{elemento}} \phi_j(\mathbf{x}) \frac{d}{d\mathbf{x}} \left(p(\mathbf{x}) \frac{d}{d\mathbf{x}} \Phi(\mathbf{x}) \right) d\mathbf{x} + \int_{\text{elemento}} \phi_j(\mathbf{x}) (q(\mathbf{x}) \Phi(\mathbf{x}) - f(\mathbf{x})) d\mathbf{x} = 0$$

Sin embargo la forma de la aproximación $\Phi(\mathbf{x}) = \sum_{j=1}^n c_j \phi_j(\mathbf{x})$ que estamos utilizando en este ejemplo no tiene segunda derivada. Usando integración por partes se obtiene una expresión equivalente a la anterior, pero que sólo involucra primeras derivadas:

$$\int_{\text{elemento}} p(\mathbf{x}) \frac{d\Phi(\mathbf{x})}{d\mathbf{x}} \frac{d\phi_j(\mathbf{x})}{d\mathbf{x}} d\mathbf{x} + \int_{\text{elemento}} \phi_j(\mathbf{x}) (q(\mathbf{x}) \Phi(\mathbf{x}) - f(\mathbf{x})) d\mathbf{x} = 0$$

al evaluar esas integrales para cada función base, ϕ_j , se obtiene un sistema de ecuaciones para los parámetros c_j que corresponden a las funciones que son diferentes de cero dentro del elemento. Estas ecuaciones todavía no toman en cuenta las condiciones de frontera. En este documento, las integrales se calcularán usando la cuadratura de Gauss. Abajo se muestra este procedimiento (cálculo de las integrales con cuadratura de Gauss) para uno de los elementos:

In[31]:=

```

integrationweights = N[{5/9, 8/9, 5/9}];

integrationpoints = N[{-sqrt(3/5), 0, sqrt(3/5)}];

elemento = 2;

nodoini = First[incidentes[[elemento]]];
nodofin = Last[incidentes[[elemento]]];

xini = coords[[nodoini]];
xfin = coords[[nodofin]];

(* Below, xi is a local variable inside the element, -1<=xi<=1 *)
integrationvar[xi_] := (1 - xi)/2 xini + (1 + xi)/2 xfin;

integrationfunction[u_, k_] := p[u] * D[u] * phi_k'[u] + phi_k[u] * (q[u] * D[u] - f[u]);

ecuaciones =
  Table[
    0.5 * (xfin - xini) * Sum[integrationweights[[j]] *
      integrationfunction[integrationvar[integrationpoints[[j]]], k],
    {j, 1, Length[integrationpoints]}] == 0,
  {k, 1, n}];

ecuaciones = Expand[N[ecuaciones]];

Print["Ecuaciones sin condiciones de frontera"];
Print["Elemento:", elemento];
Print[TraditionalForm[TableForm[ecuaciones]]]

```

Ecuaciones sin condiciones de frontera

Elemento:2

True

3.11828 c₂ + 0.00723589 c₃ + 0.620862 = 0.0.00723589 c₂ + 3.11828 c₃ + 0.722168 = 0.

True

True

True

True

La matriz y el vector del sistema de ecuaciones son guardados en memoria como arreglos dispersos (`SparseArray`), es decir, sólo se guardan en memoria los elementos diferentes de cero:

In[45]=

```
ecuaciones = ReplaceAll[ecuaciones, True → 0];  
parametros = Table[cj, {j, 1, n}];  
{v, m} = N[ CoefficientArrays[ecuaciones, parametros] ];  
{ArrayRules[m], ArrayRules[-v]}
```

Out[48]=

```
{{{2, 2} → 3.11828, {2, 3} → 0.00723589, {3, 2} → 0.00723589,  
  {3, 3} → 3.11828, {_, _} → 0.}, {{2} → -0.620862, {3} → -0.722168, {_} → 0.}}
```


A continuación se calculan las matrices y vectores de carga para todos los elementos en este ejemplo:

In[49]=

```

parametros = Table[cj, {j, 1, n}];
matrices = {};
cargas = {};

integrationweights = N[{5/9, 8/9, 5/9}];

integrationpoints = N[{-√(3/5), 0, √(3/5)}];

Do[
  nodoini = First[incidentes[[elemento]];
  nodofin = Last[incidentes[[elemento]];

  xini = coords[[nodoini]];
  xfin = coords[[nodofin]];

  (* Below, ξ is a local variable inside the element, -1 ≤ ξ ≤ 1 *)
  integrationvar[ξ_] := (1 - ξ)/2 xini + (1 + ξ)/2 xfin;

  integrationfunction[u_, k_] := p[u] * Φ'[u] * φk'[u] + φk[u] * (q[u] * Φ[u] - f[u]);

  ecuaciones =
  Table[
    0.5 * (xfin - xini) * Sum[integrationweights[[j]] *
      integrationfunction[integrationvar[integrationpoints[[j]]], k],
    {j, 1, Length[integrationpoints]}] == 0,
    {k, 1, n}];

  ecuaciones = ReplaceAll[ecuaciones, True → 0];

  {v, m} = N[ CoefficientArrays[ecuaciones, parametros] ];

  AppendTo[matrices, m];
  AppendTo[cargas, -v];

  Print["Matriz dispersa y vector disperso para elemento:", elemento];
  Print[ArrayRules[m], " ", ArrayRules[-v]],
  {elemento, 1, n - 1}]

```

Matriz dispersa y vector disperso para elemento:1

```
{ {1, 1} → 3.68081, {1, 2} → 1.6623, {2, 1} → 1.6623, {2, 2} → 3.68081, {_, _} → 0.}
  { {1} → -0.296062, {2} → -0.592123, {_} → 0.}
```

Matriz dispersa y vector disperso para elemento:2

```
{ {2, 2} → 3.11828, {2, 3} → 0.00723589, {3, 2} → 0.00723589, {3, 3} → 3.11828, {_, _} → 0.}
  { {2} → -0.620862, {3} → -0.722168, {_} → 0.}
```

Matriz dispersa y vector disperso para elemento:3

```
{ {3, 3} → 4.24827, {3, 4} → -2.03068, {4, 3} → -2.03068, {4, 4} → 4.24827, {_, _} → 0.}
  { {3} → -0.635263, {4} → -0.686262, {_} → 0.}
```

Matriz dispersa y vector disperso para elemento:4

```
{ {4, 4} → 6.32329, {4, 5} → -4.6032, {5, 4} → -4.6032, {5, 5} → 6.32329, {_, _} → 0.}
  { {4} → -0.602547, {5} → -0.63323, {_} → 0.}
```

Matriz dispersa y vector disperso para elemento:5

```
{ {5, 5} → 9.09636, {5, 6} → -7.69093, {6, 5} → -7.69093, {6, 6} → 9.09636, {_, _} → 0.}
  { {5} → -0.56294, {6} → -0.583424, {_} → 0.}
```

Matriz dispersa y vector disperso para elemento:6

```
{ {6, 6} → 12.442, {6, 7} → -11.2537, {7, 6} → -11.2537, {7, 7} → 12.442, {_, _} → 0.}
  { {6} → -0.525239, {7} → -0.539882, {_} → 0.}
```

A continuación las ecuaciones de todos los elementos se suman para obtener un único sistema de ecuaciones para todos los parámetros c_j . Este procedimiento se conoce como ensamblado de la matriz global. El sistema de ecuaciones que se obtiene todavía no incluye las condiciones de frontera, por eso este sistema no tiene solución; la matriz es singular:

In[55]=

```
matrizGlobal = Total[matrices];

cargaGlobal = Total[cargas];

Print[
  "Matriz dispersa y vector disperso globales, sin condiciones de frontera"];
Print[ArrayRules[matrizGlobal], "\n", ArrayRules[cargaGlobal]]
Print["Ecuaciones globales sin condiciones de frontera"];
Print[MatrixForm[matrizGlobal], ".",
  MatrixForm[parametros], "=",
  MatrixForm[cargaGlobal]
]
```

Matriz dispersa y vector disperso globales, sin condiciones de frontera

```
{ {1, 1} → 3.68081, {1, 2} → 1.6623, {2, 2} → 6.79909, {2, 3} → 0.00723589, {2, 1} → 1.6623,
  {3, 3} → 7.36655, {3, 4} → -2.03068, {3, 2} → 0.00723589, {4, 4} → 10.5716, {4, 5} → -4.6032,
  {4, 3} → -2.03068, {5, 5} → 15.4196, {5, 6} → -7.69093, {5, 4} → -4.6032, {6, 6} → 21.5384,
  {6, 7} → -11.2537, {6, 5} → -7.69093, {7, 6} → -11.2537, {7, 7} → 12.442, {_, _} → 0.}
{ {1} → -0.296062, {2} → -1.21298, {3} → -1.35743, {4} → -1.28881,
  {5} → -1.19617, {6} → -1.10866, {7} → -0.539882, {_} → 0.}
```

Ecuaciones globales sin condiciones de frontera

$$\begin{pmatrix} 3.68081 & 1.6623 & 0. & 0. & 0. & 0. & 0. \\ 1.6623 & 6.79909 & 0.00723589 & 0. & 0. & 0. & 0. \\ 0. & 0.00723589 & 7.36655 & -2.03068 & 0. & 0. & 0. \\ 0. & 0. & -2.03068 & 10.5716 & -4.6032 & 0. & 0. \\ 0. & 0. & 0. & -4.6032 & 15.4196 & -7.69093 & 0. \\ 0. & 0. & 0. & 0. & -7.69093 & 21.5384 & -11.2537 \\ 0. & 0. & 0. & 0. & 0. & -11.2537 & 12.442 \end{pmatrix}$$

$$\cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} -0.296062 \\ -1.21298 \\ -1.35743 \\ -1.28881 \\ -1.19617 \\ -1.10866 \\ -0.539882 \end{pmatrix}$$

A continuación se insertan las condiciones de frontera, reemplazando a las ecuaciones correspondientes:

In[61]:=

```
matrizGlobal[[1]] = Prepend[Table[0, {j, 2, n}], 1];
cargaGlobal[[1]] = ya;

matrizGlobal[[n]] = Append[Table[0, {j, 2, n}], 1];
cargaGlobal[[n]] = yb;

Print[
  "Matriz dispersa y vector disperso globales, CON condiciones de frontera"];
Print[ArrayRules[matrizGlobal], "\n", ArrayRules[cargaGlobal]]
Print["Ecuaciones CON condiciones de frontera"];
Print[MatrixForm[matrizGlobal], ".",
  MatrixForm[parametros], "=",
  MatrixForm[cargaGlobal]
]
```

Matriz dispersa y vector disperso globales, CON condiciones de frontera

```
{ {1, 1} → 1, {2, 1} → 1.6623, {2, 2} → 6.79909, {2, 3} → 0.00723589, {3, 2} → 0.00723589,
  {3, 3} → 7.36655, {3, 4} → -2.03068, {4, 3} → -2.03068, {4, 4} → 10.5716,
  {4, 5} → -4.6032, {5, 4} → -4.6032, {5, 5} → 15.4196, {5, 6} → -7.69093,
  {6, 5} → -7.69093, {6, 6} → 21.5384, {6, 7} → -11.2537, {7, 7} → 1, {_, _} → 0.}
{ {2} → -1.21298, {3} → -1.35743, {4} → -1.28881,
  {5} → -1.19617, {6} → -1.10866, {7} → 0.5, {_, _} → 0.}
```

Ecuaciones CON condiciones de frontera

$$\begin{pmatrix} 1 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1.6623 & 6.79909 & 0.00723589 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.00723589 & 7.36655 & -2.03068 & 0. & 0. & 0. & 0. \\ 0. & 0. & -2.03068 & 10.5716 & -4.6032 & 0. & 0. & 0. \\ 0. & 0. & 0. & -4.6032 & 15.4196 & -7.69093 & 0. & 0. \\ 0. & 0. & 0. & 0. & -7.69093 & 21.5384 & -11.2537 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{pmatrix} = \begin{pmatrix} 0. \\ -1.21298 \\ -1.35743 \\ -1.28881 \\ -1.19617 \\ -1.10866 \\ 0.5 \end{pmatrix}$$

Como ya tenemos la matriz y el vector del sistema lineal, podemos obtener la solución usando el comando `LinearSolve[]` de *Mathematica*:

In[69]=

```
solvevector = LinearSolve[matrizGlobal, cargaGlobal]
```

Out[69]=

```
{0., -0.178155, -0.234043, -0.181193, -0.0328933, 0.198029, 0.5}
```

Ponemos la solución en un formato adecuado para reemplazar en la expresión de $\Phi(x)$:

In[70]=

```
solu = Table[parametros[[j]] → solvevector[[j]], {j, 1, n}]
```

Out[70]=

```
{c1 → 0., c2 → -0.178155, c3 → -0.234043,
  c4 → -0.181193, c5 → -0.0328933, c6 → 0.198029, c7 → 0.5}
```

La solución aproximada se obtiene reemplazando los coeficientes c_j en la expresión $\Phi(x) = \sum_{j=1}^n c_j \phi_j(x)$. Abajo se le llama $g(x)$ a la solución aproximada que se obtiene por este procedimiento:

In[71]=

```
g[x_] := ReplaceAll[ϕ[x], solu];

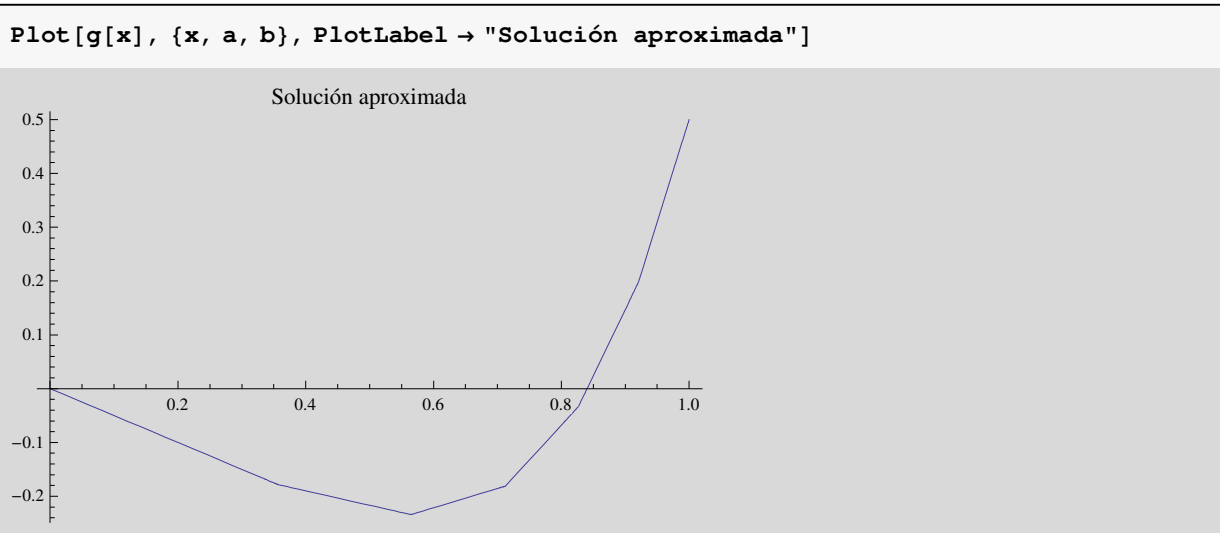
Print["Solución aproximada por el método de Galerkin: "];
Print["g[x]=", TraditionalForm[g[x]]]
```

Solución aproximada por el método de Galerkin:

$$\begin{aligned}
 g[x] = & 0.198029 \left(\begin{cases} 10.673 (x - 0.827087) & 0.827087 < x \leq 0.920782 \\ 12.6234 (1. - x) & 0.920782 < x \leq 1. \end{cases} \right) - \\
 & 0.0328933 \left(\begin{cases} 8.72044 (x - 0.712414) & 0.712414 < x \leq 0.827087 \\ 10.673 (0.920782 - x) & 0.827087 < x \leq 0.920782 \end{cases} \right) - \\
 & 0.181193 \left(\begin{cases} 6.7641 (x - 0.564575) & 0.564575 < x \leq 0.712414 \\ 8.72044 (0.827087 - x) & 0.712414 < x \leq 0.827087 \end{cases} \right) - \\
 & 0.234043 \left(\begin{cases} 4.7992 (x - 0.356207) & 0.356207 < x \leq 0.564575 \\ 6.7641 (0.712414 - x) & 0.564575 < x \leq 0.712414 \end{cases} \right) - \\
 & 0.178155 \left(\begin{cases} 2.80735 (x + 0.) & 0. < x \leq 0.356207 \\ 4.7992 (0.564575 - x) & 0.356207 < x \leq 0.564575 \end{cases} \right) + \\
 & 0.5 \left(\begin{cases} 12.6234 (x - 0.920782) & 0.920782 < x \leq 1. \\ 0 & \text{True} \end{cases} \right) + 0.
 \end{aligned}$$

Ésta es la gráfica de la solución aproximada con el Método de Elemento Finito:

In[74]=



A continuación se muestra la gráfica de la solución analítica exacta, obtenida con el comando DSolve:

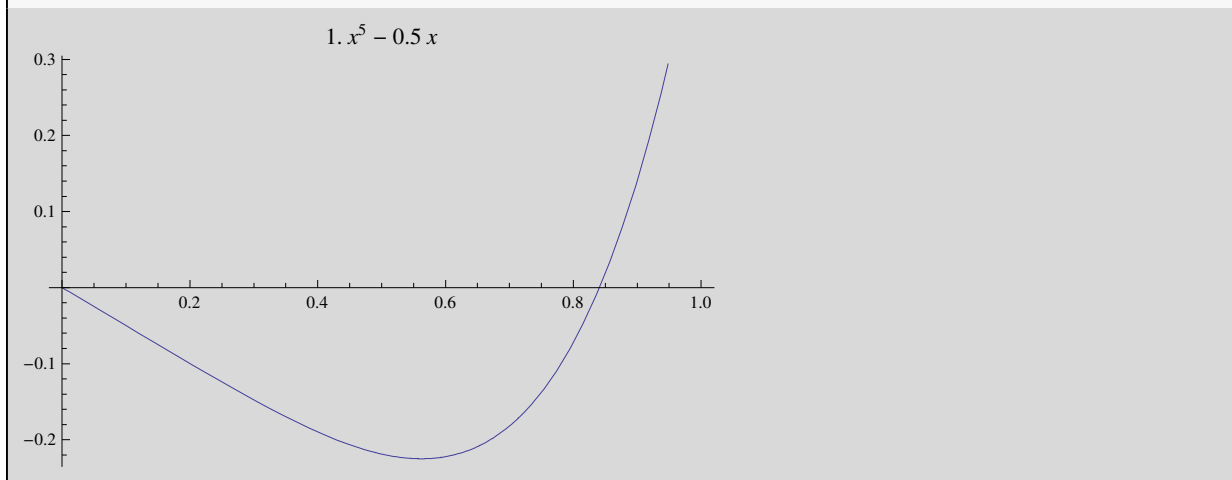
In[75]=

```
solexacta =
  DSolve[{-D[x (p[x] D[x] y[x]) + q[x] * y[x] == f[x], y[a] == ya, y[b] == yb}, y[x], x];

exacta[x_] := Evaluate[Expand[ReplaceAll[y[x], solexacta[[1]] ]]];

Plot[exacta[x], {x, a, b}, PlotLabel -> exacta[x]]
```

Out[77]=

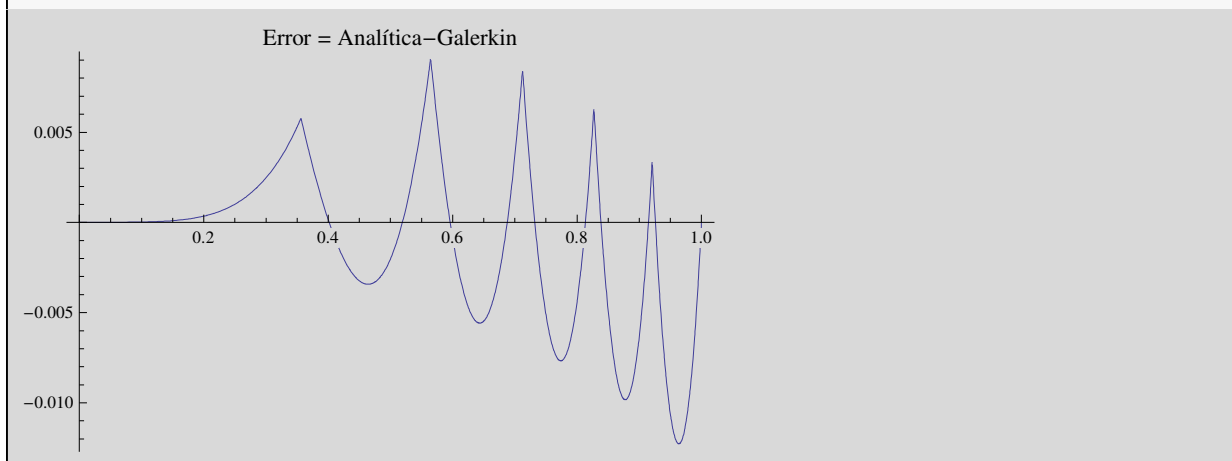


Podemos graficar el error (diferencia) entre la solución analítica y la aproximación con el Método de Elemento Finito:

In[78]=

```
Plot[exacta[x] - g[x], {x, a, b}, PlotLabel -> "Error = Analítica-Galerkin"]
```

Out[78]=



Ejercicios

I. Modifica éste documento para obtener una solución aproximada a este otro

problema:

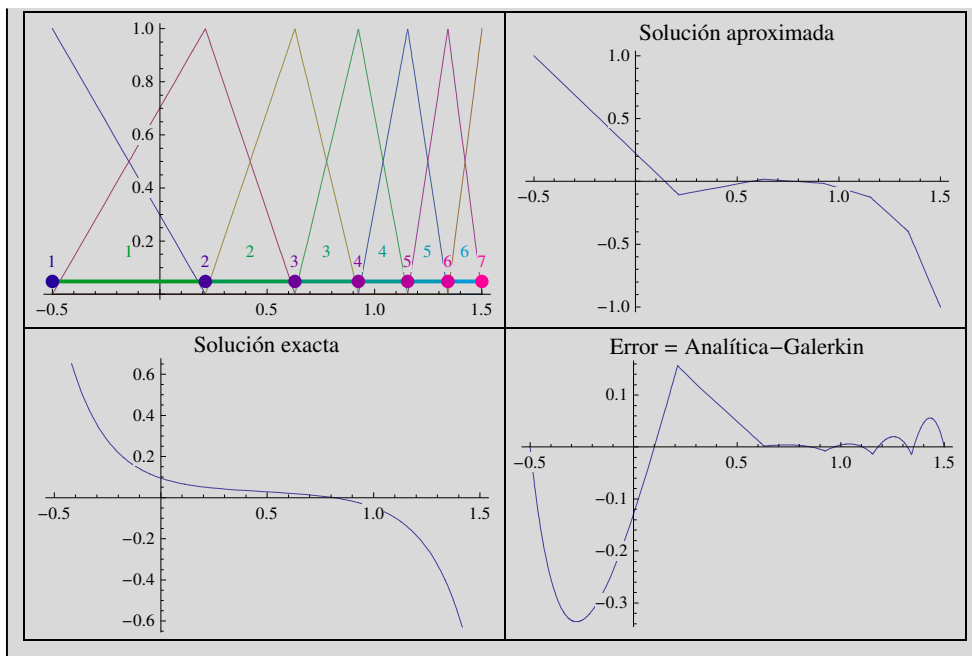
$$-\frac{d}{dx} \left(p(x) \frac{d}{dx} y(x) \right) + q(x) y(x) = f(x),$$

con $p(x) = 1$, $q(x) = 30$, $f(x) = \text{Cos}[x]$,

$$y(-0.5) = 1.0$$

$$y(1.5) = -1.0$$

debes obtener las gráficas que se muestran abajo para las funciones base, la solución aproximada y el error:



Referencias

Adaptado por José Luis Gómez Muñoz
<http://homepage.cem.itesm.mx/jose.luis.gomez>

Basado en el trabajo de John H. Mathews
<http://math.fullerton.edu/mathews/n2003/GalerkinMod.html>
 también en la sección 11.5 "El método de Rayleigh-Ritz" del libro de Richard L. Burden y J. Douglas Faires, "Análisis numérico" 9a. edición, editorial Cengage, págs 696-711
 y finalmente también en la sección 6.2 "Galerkin's method with piecewise polynomials" del libro de Kenneth Eriksson, D. Estep, P. Hansbo & C. Johnson, "Computational Differential Equations" Volumen 1, Cambridge University Press, 1997.

También revisar el capítulo 2 del libro:
 J.N. Reddy
 An Introduction to the Finite Element Method, 3rd Edition

McGraw-Hill

Autor José Luis Gómez Muñoz

<http://homepage.cem.itesm.mx/jose.luis.gomez>

In[79]:=

```
$Version
```

Out[79]=

```
9.0 for Microsoft Windows (64-bit) (January 25, 2013)
```

In[80]:=

```
DateString[]
```

Out[80]=

```
Wed 2 Apr 2014 16:00:33
```