
El método de Galerkin con funciones base de soporte local

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.

Adaptado por José Luis Gómez Muñoz

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

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]=

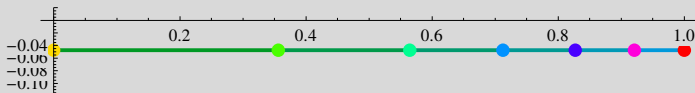
```
{0., 0.356207, 0.564575, 0.712414, 0.827087, 0.920782, 1.}
```

Representación geométrica de la partición (mallado) del dominio $a \leq x \leq b$

In[15]=

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

Out[17]=

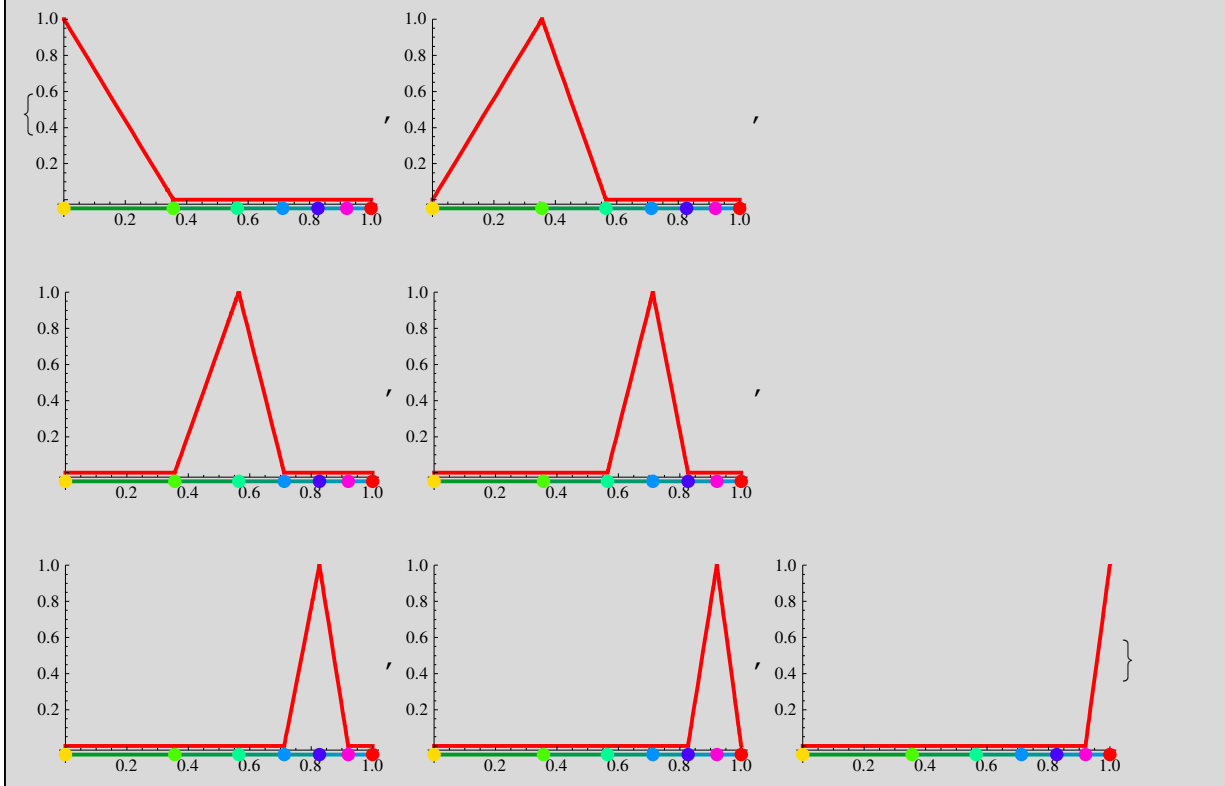


La solución aproximada $\Phi(x)$ será una combinación lineal de ciertas funciones linealmente independientes, llamadas funciones base: $\Phi(x) = \sum_{j=1}^n c_j \phi_j(x)$. Las funciones base $\phi_j(x)$ usadas en el método de Elemento Finito tienen soporte local, lo que quiere decir que cada una es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

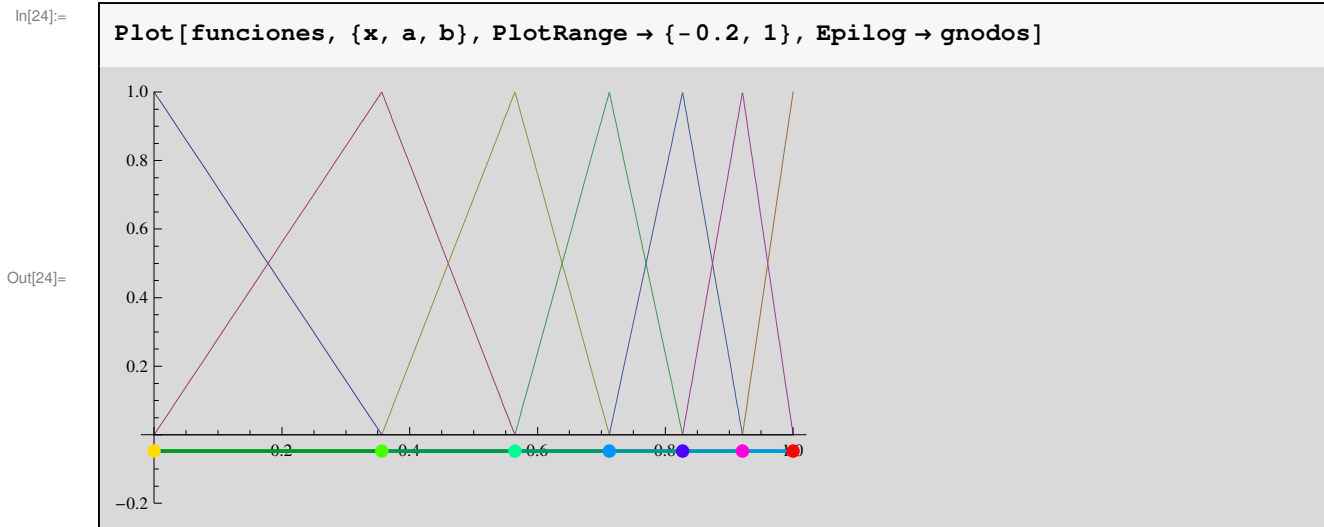
In[18]=

```
Clear[ $\phi$ ,  $\mathbf{x}$ ];
 $\phi_1[\mathbf{x}_-]$  := {  $\frac{\text{coords}[[2]]-\mathbf{x}}{\text{coords}[[2]]-a}$   $a < \mathbf{x} \leq \text{coords}[[2]]$  ;
0 True
}
 $\phi_n[\mathbf{x}_-]$  := {  $\frac{\mathbf{x}-\text{coords}[[n-1]]}{b-\text{coords}[[n-1]]}$   $\text{coords}[[n-1]] < \mathbf{x} \leq b$  ;
0 True
}
 $\phi_j[\mathbf{x}_-]$  := {  $\frac{\mathbf{x}-\text{coords}[[j-1]]}{\text{coords}[[j]]-\text{coords}[[j-1]]}$   $\text{coords}[[j-1]] < \mathbf{x} \leq \text{coords}[[j]]$ 
 $\frac{\text{coords}[[j+1]]-\mathbf{x}}{\text{coords}[[j+1]]-\text{coords}[[j]]}$   $\text{coords}[[j]] < \mathbf{x} \leq \text{coords}[[j+1]]$  ;
0 True
}
funciones = Table[ $\phi_j[\mathbf{x}]$ , {j, 1, n}];
Table[Plot[funciones[[j]], { $\mathbf{x}$ , a, b},
PlotRange -> {-4 distancia, 1}, PlotStyle -> {Red, Thick},
Epilog -> gnodos, AxesOrigin -> {0, -distancia}], {j, 1, n}]
```

Out[23]=



Abajo se muestran todas las funciones base, junto con los nodos (mallado):



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[25]:=

```
Clear[Φ, c];

Φ[x_] := Sum[c_j * φ_j[x], {j, 1, n}];

Print["Forma de la aproximación a la solución:"];
Print["Φ[x]=", TraditionalForm[Φ[x]]]
```

Forma de la aproximación a la solución:

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

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, $r(x)$. En otras palabras, si el residuo $r(x)$, mostrado abajo, fuera cero, en ese caso la solución aproximada sería la solución exacta:

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

El método de 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 $\{\phi_j\}_{j=2}^{n-1}$ (sin incluir ϕ_1 ni ϕ_n , porque serán usadas para cumplir las condiciones de frontera). Esto significa que el producto punto del residuo por cada función base debe ser cero:

$$\int_a^b \phi_j(x) r(x) dx = 0 \quad \text{para } j = 2 \dots n-1$$

es decir

$$-\int_a^b \phi_j(x) \frac{d}{dx} \left(p(x) \frac{d}{dx} \Phi(x) \right) dx + \int_a^b \phi_j(x) (q(x) \Phi(x) - f(x)) dx = 0 \quad \text{para } j = 2 \dots n-1$$

Sin embargo la aproximación $\Phi(x)$ que estamos utilizando en este ejemplo no tiene segunda derivada. Usando integración por partes se obtiene una expresión que sólo involucra primeras derivadas:

$$-\left[\phi_j(x) p(x) \frac{d\Phi(x)}{dx} \right]_a^b + \int_a^b p(x) \frac{d\Phi(x)}{dx} \frac{d\phi_j(x)}{dx} dx + \int_a^b \phi_j(x) (q(x) \Phi(x) - f(x)) dx = 0 \quad \text{para } j = 2 \dots n-1,$$

pero recordemos que debido a su soporte local, $\phi_j(b) = \phi_j(a) = 0$ para $j = 2 \dots n-1$, entonces cada producto punto igualado a cero queda:

$$\int_a^b p(x) \frac{d\Phi(x)}{dx} \frac{d\phi_j(x)}{dx} dx + \int_a^b \phi_j(x) (q(x) \Phi(x) - f(x)) dx = 0 \quad \text{para } j = 2 \dots n-1,$$

que, junto con las condiciones de frontera $c_1 = y_a$, $c_n = y_b$, forman un sistema de n ecuaciones para los n parámetros c_j , como se muestra abajo.

Nota: Las integraciones deberían ser integraciones numéricas, con el comando `NIntegrate[]`, pero se usan las integrales simbólicas, que son más lentas, para efectos de claridad en este documento didáctico:

In[29]:=

```

ecuaciones =
  Table [ $\int_a^b p[x] * \bar{\varphi}'[x] * \phi_j'[x] dx + \int_a^b \phi_j[x] * (q[x] * \bar{\varphi}[x] - f[x]) dx == 0,$ 
    {j, 2, n - 1}];

ecuaciones = Join [{c1 == ya}, Expand[N[ecuaciones]], {cn == yb}}];

TraditionalForm[TableForm[ecuaciones]]

```

Out[31]/TraditionalForm=

```

c1 = 0.
1.6623 c1 + 6.79909 c2 + 0.00723589 c3 + 1.21298 = 0.
0.00723589 c2 + 7.36655 c3 - 2.03068 c4 + 1.35743 = 0.
-2.03068 c3 + 10.5716 c4 - 4.6032 c5 + 1.28881 = 0.
-4.6032 c4 + 15.4196 c5 - 7.69093 c6 + 1.19617 = 0.
-7.69093 c5 + 21.5384 c6 - 11.2537 c7 + 1.10866 = 0.
c7 = 0.5

```

A continuación se muestra el sistema de ecuaciones en forma matricial:

In[32]:=

```

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

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

Print [MatrixForm[m], ".",
  MatrixForm[parametros], "==",
  MatrixForm[-v]
]

```

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

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

In[35]:=

```
LinearSolve[m, -v]
```

Out[35]=

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

Sin embargo el comando `NSolve` proporciona la misma respuesta en un formato más útil para ser usado después en este documento:

In[36]:=

```
soluciones = NSolve[ecuaciones, parametros]
```

Out[36]=

```
{ {c1 → 1.33332 × 10-16, 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[37]:=

```
g[x_] := ReplaceAll[Φ[x], soluciones[[1]]];
Print["Solución aproximada por el método de Galerkin: "];
Print["g[x]=", TraditionalForm[g[x]]]
```

Solución aproximada por el método de Galerkin:

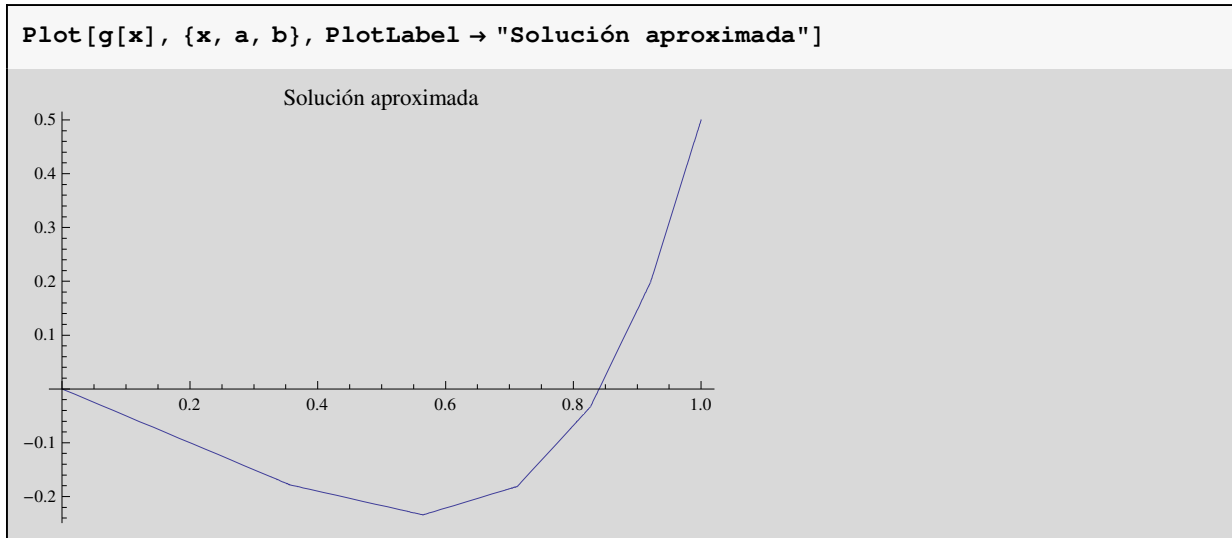
$$\begin{aligned}
 g[x] = & 1.33332 \times 10^{-16} \left(\begin{array}{ll} 2.80735 (0.356207 - x) & 0. < x \leq 0.356207 \\ 0 & \text{True} \end{array} \right) + \\
 & 0.198029 \left(\begin{array}{ll} 10.673 (x - 0.827087) & 0.827087 < x \leq 0.920782 \\ 12.6234 (1. - x) & 0.920782 < x \leq 1. \end{array} \right) - \\
 & 0.0328933 \left(\begin{array}{ll} 8.72044 (x - 0.712414) & 0.712414 < x \leq 0.827087 \\ 10.673 (0.920782 - x) & 0.827087 < x \leq 0.920782 \end{array} \right) - \\
 & 0.181193 \left(\begin{array}{ll} 6.7641 (x - 0.564575) & 0.564575 < x \leq 0.712414 \\ 8.72044 (0.827087 - x) & 0.712414 < x \leq 0.827087 \end{array} \right) - \\
 & 0.234043 \left(\begin{array}{ll} 4.7992 (x - 0.356207) & 0.356207 < x \leq 0.564575 \\ 6.7641 (0.712414 - x) & 0.564575 < x \leq 0.712414 \end{array} \right) - \\
 & 0.178155 \left(\begin{array}{ll} 2.80735 (x + 0.) & 0. < x \leq 0.356207 \\ 4.7992 (0.564575 - x) & 0.356207 < x \leq 0.564575 \end{array} \right) + \\
 & 0.5 \left(\begin{array}{ll} 12.6234 (x - 0.920782) & 0.920782 < x \leq 1. \\ 0 & \text{True} \end{array} \right)
 \end{aligned}$$

Ésta es la gráfica de la solución aproximada con el método de Bubnov-Galerkin:

In[40]=

```
Plot[g[x], {x, a, b}, PlotLabel -> "Solución aproximada"]
```

Out[40]=

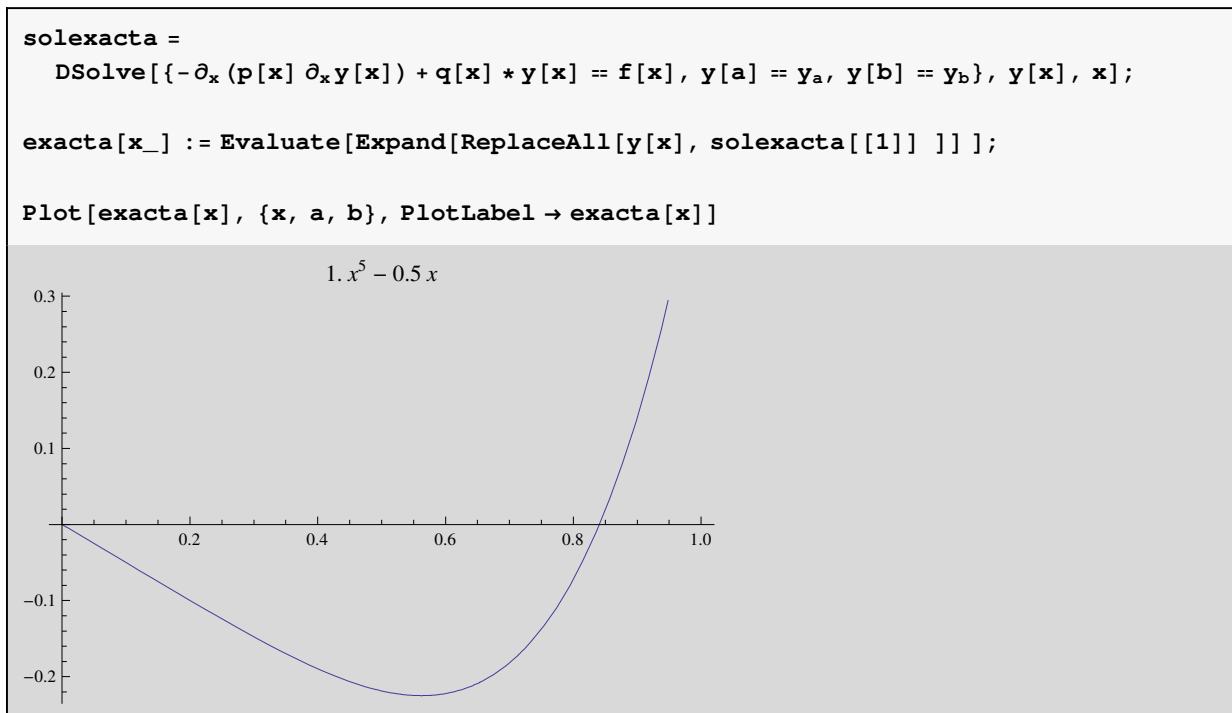


A continuación se muestra la gráfica de la solución analítica exacta, obtenida con el comando DSolve:

In[41]=

```
solexacta =
  DSolve[{-∂x (p[x] ∂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[43]=

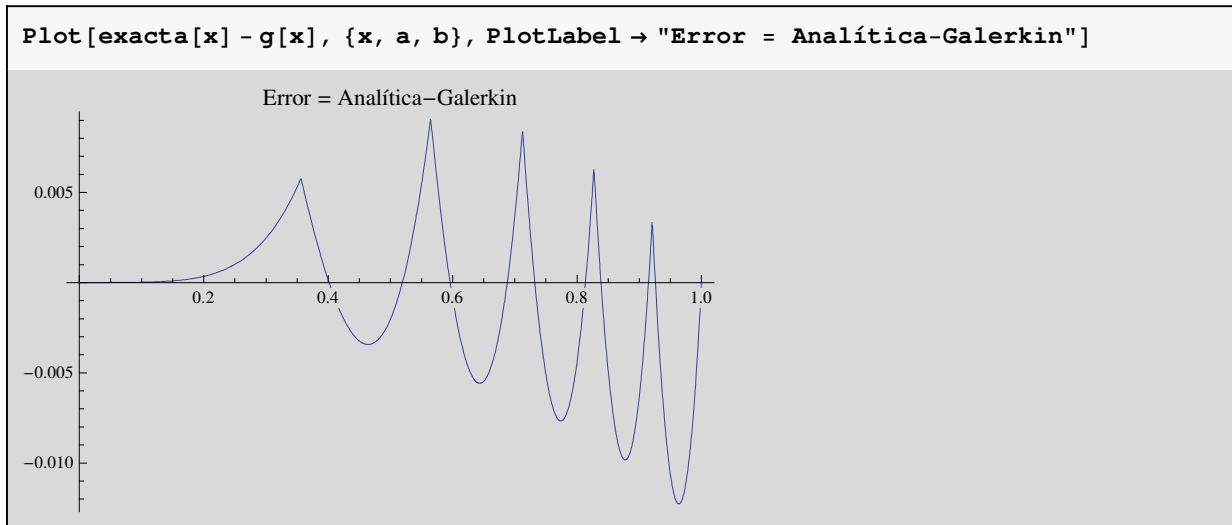


Podemos graficar juntas la solución analítica y la aproximación con el método de Galerkin:

In[44]:=

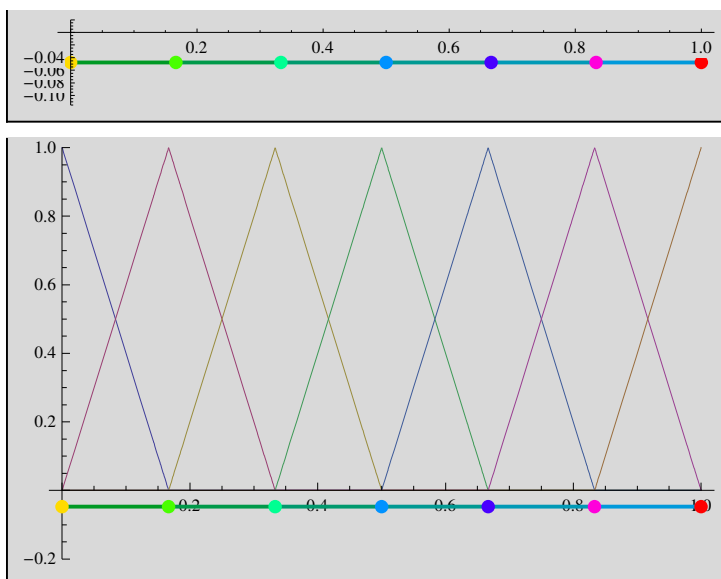
```
Plot[exacta[x] - g[x], {x, a, b}, PlotLabel -> "Error = Analítica-Galerkin"]
```

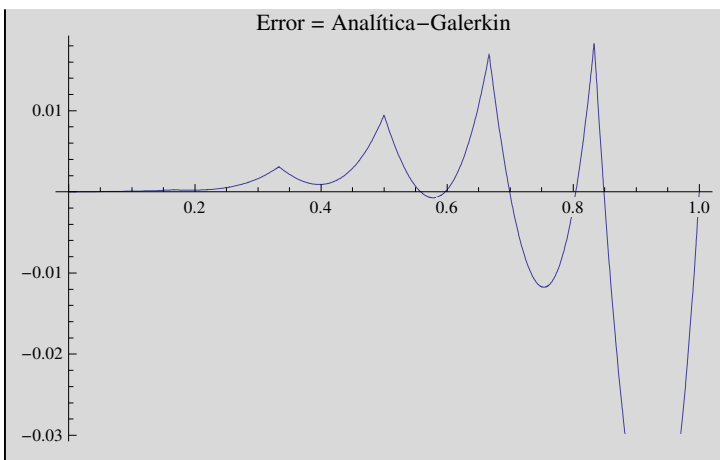
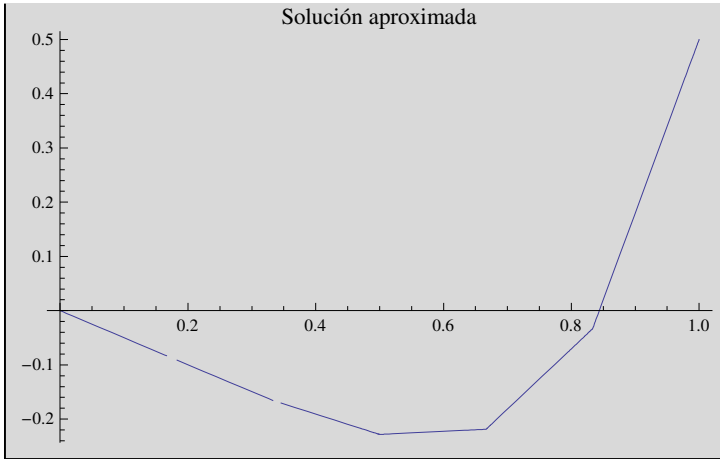
Out[44]=



Ejercicios

I. Modifica éste documento para obtener una solución aproximada al mismo problema, pero que los nodos del mallado estén igualmente espaciados, debes obtener las gráficas que se muestran abajo para las funciones base, la solución aproximada y el error:





2. Modifica éste documento para obtener una solución aproximada a este otro problema:

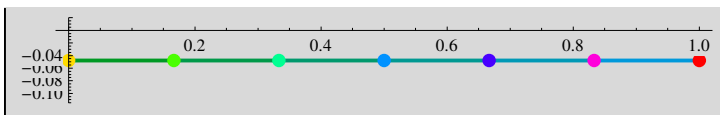
$$-\frac{d}{dx} \left(p(x) \frac{d}{dx} y(x) \right) + q(x) y(x) = f(x),$$

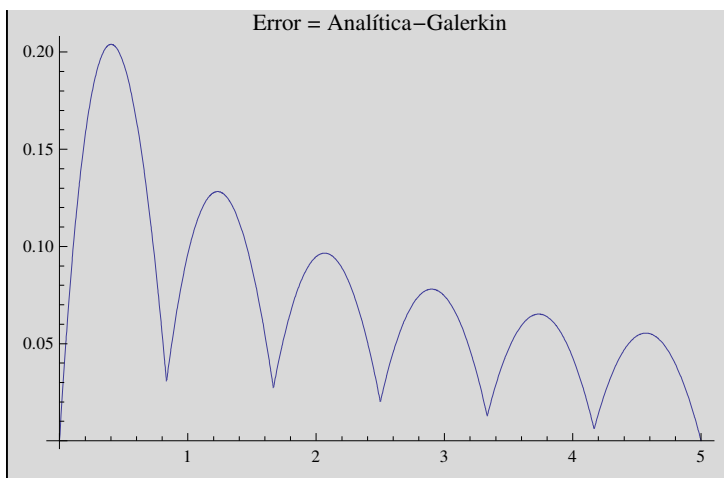
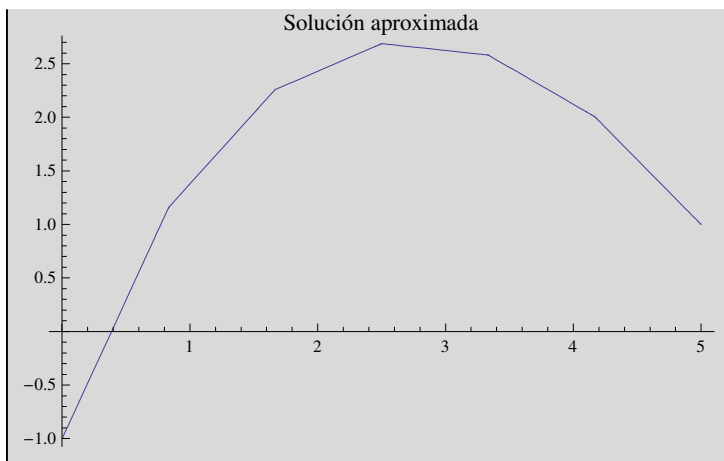
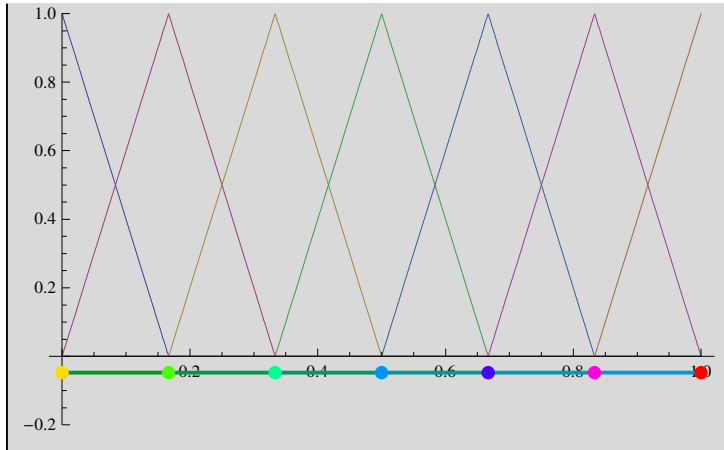
con $p(x) = 1 + x$, $q(x) = 0$, $f(x) = x$,

$y(0) = -1$

$y(5) = +1$

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 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[45]=

\$Version

Out[45]=

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

In[46]=

DateString[]

Out[46]=

Wed 2 Apr 2014 16:08:49