

Examen 4 Análisis Numérico 2

Pablo Buitrago Jaramillo

Junio 2023

1

Los elementos finitos de tipo \mathbb{P}_2 (Lagrange) usan el espacio discreto $V_h := \left\{ v \in C([0, 1]) : v|_{[x_j, x_{j+1}]} \in \mathbb{P}_2, 0 \leq j \leq n \right\}$ y su subespacio $V_{0h} := \{v \in V_h : v(0) = v(1) = 0\}$. A partir de lo anterior, se definen las funciones de referencia $\phi(x), \psi(x)$ como sigue:

$$\phi(x) := \begin{cases} (1+x)(1+2x) & \text{si } -1 \leq x \leq 0 \\ (1-x)(1-2x) & \text{si } 0 \leq x \leq 1 \\ 0 & \text{si } |x| > 1, \end{cases}$$
$$\psi(x) := \begin{cases} 1-4x^2 & \text{si } |x| \leq 1/2 \\ 0 & \text{si } |x| > 1/2 \end{cases}$$

1.1

Sea $\Omega = (a, b) = (0, 1)$. Considere el problema de Poisson siguiente: Dada $f \in L^2(\Omega)$, determinar la función $u \in H_0^1(\Omega)$ que satisface (en el sentido débil):

$$\begin{cases} -u_{xx}(x) = f(x) \\ u(0) = 0, u(1) = 0 \end{cases} \quad x \in \Omega \quad (1)$$

Para una malla uniforme sobre $[0, 1]$, las funciones base son:

$$\phi_j(x) := \phi\left(\frac{x-x_j}{h}\right) \text{ para } j = 0, \dots, n+1, \psi_{j+1/2}(x) = \psi\left(\frac{x-x_{j+1/2}}{h}\right) \text{ para } j = 0, \dots, n$$

Además, es sabido que la solución aproximada u_h es de la forma

$$u_h(x) = \sum_{j=0}^{n+1} u_h(x_j) \phi_j(x) + \sum_{j=0}^n u_h(x_{j+1/2}) \psi_{j+1/2}(x) \quad (2)$$

- Considere una solución exacta no nula de la ecuación 1, la cual denotaremos por $u_{ex}(x)$ (cada uno de ustedes la propone).
- Encuentre la formulación variacional del problema (1) y describa tal formulación como un sistema matricial determinando de forma explícita las matrices respectivas.
- Implemente el sistema matricial obtenido en el inciso anterior y obtener la solución aproximada $u_h(x)$.
- Posteriormente, construya una tabla de error en función de la cantidad de nodos, es decir, puede considerar $n = 5, 10, 20, 50, 100$ y construir la tabla asociada a $\|u_{ex} - u_h\|$.

Nota: La norma del error puede ser $L^2(\Omega)$ o $L^\infty(\Omega)$.

1.1.1 Solución exacta

Una solución exacta no nula del problema es $u(x) = \sin(4\pi x)\cos(4\pi x)$ con $x \in [0, 1]$. Esta cumple con las condiciones de frontera $u(x=0) = u(x=1) = 0$ y es por lo menos dos veces derivable

- $u_x(x) = 4\pi\cos(8\pi x)$
- $u_{xx}(x) = -32\pi^2\sin(8\pi x)$

1.1.2 Formulación variacional y sistema matricial

Sea $v(x)$ una función de prueba

$$\begin{aligned} -u_{xx}(x)v(x) &= f(x)v(x) \\ -\int_0^1 u_{xx}(x)v(x)dx &= \int_0^1 f(x)v(x)dx \end{aligned}$$

Integrando por partes con:

$$\begin{aligned} u &= v(x) & dv &= u_{xx}(x)dx \\ du &= v_x(x)dx & v &= u_x(x) \end{aligned}$$

$$\begin{aligned} \int_0^1 u_x(x)v_x(x)dx - \left[u_x(x)v(x) \right]_{x=0}^{x=1} &= \int_0^1 f(x)v(x)dx \\ \int_0^1 u_x(x)v_x(x)dx &= \int_0^1 f(x)v(x)dx \end{aligned} \quad (3)$$

Donde (3) corresponde a la formulación variacional para este problema. Las funciones base de este problema pueden ser reemplazada por la siguiente:

$$\lambda_j(x) := \begin{cases} \phi_j(x) & \text{si } j \% 2 = 0 \\ \psi_j(x) & \text{si } j \% 2 \neq 0 \end{cases} \quad (4)$$

De esta manera, dado un numero n de nodos, realmente estaríamos considerando una malla de $2n+1$ nodos en el mismo intervalo. Este cambio nos permite entonces definir la solución aproximada como

$$u_h(x) = \sum_{j=0}^{2n+1} \alpha_j \lambda_j(x) \quad (5)$$

Que al reemplazar en (3) tenemos:

$$\begin{aligned} \int_0^1 \frac{d}{dx} \left(\sum_{j=0}^{2n+1} \alpha_j \lambda_j(x) \right) \frac{d}{dx} (\lambda_i(x)) dx &= \int_0^1 f(x) \lambda_i(x) dx \\ \sum_{j=0}^{2n+1} \alpha_j \int_0^1 \lambda_j'(x) \lambda_i'(x) dx &= \int_0^1 f(x) \lambda_i(x) dx \end{aligned}$$

La ecuacion anterior puede ser vista como un problema matricial de la forma

$$\alpha \mathbf{K} = \mathbf{b} \quad (6)$$

Donde \mathbf{K} es la matriz de rigidez cuyos elementos están dados por:

$$K_{ij} = \int_0^1 \lambda_j'(x) \lambda_i'(x) dx$$

y \mathbf{b} es el vector de carga cuyos elementos están dados por:

$$b_i = \int_0^1 f(x)\lambda_i(x)dx$$

Y el vector de coeficientes $\boldsymbol{\alpha}$ esta dado por:

$$\boldsymbol{\alpha} = \mathbf{b}\mathbf{K}^{-1}$$

Bastaría entonces con encontrar el vector de coeficientes $\boldsymbol{\alpha}$ para poder calcular la solución aproximada.

1.1.3 Implementación del sistema matricial y solución aproximada

A continuación se mostraran los resultados obtenidos de la implementación para resolver el sistema matricial y encontrar la solución aproximada.

Se utilizo el software **Julia** con los paquetes *SymPy* para hacer cálculos simbólicos y *LinearSolve* para resolver los sistemas lineales

Como puede verse en la figura (4), la solución aproximada u_h mejora su ajuste con la solución exacta $u(x)$ a medida que se incrementa el número de nodos (n) para resolver el sistema matricial y resolver la formulación variacional del problema, pero a medida que el número de nodos crece, el tiempo de cómputo también lo hace.

n	Tiempo de computo
5	00:09
10	00:35
50	14:26
100	57:40

Table 1: Tiempo de computo para $n \in [5,10,20,50]$

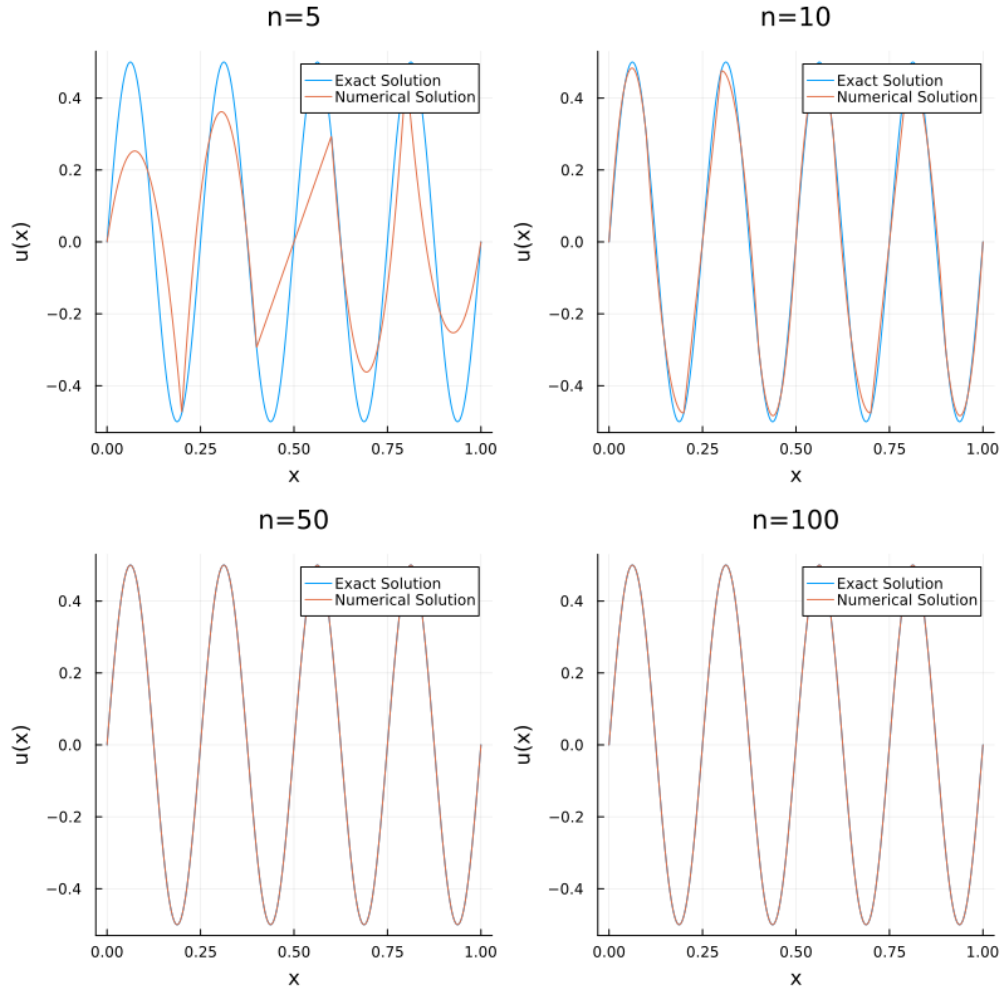


Figure 1: Comparación entre $u(x)$ y $u_h(x)$ para diferentes valores de n

A continuación se encuentran los coeficientes del sistema matricial correspondiente a $n = 5$.

$$K = \begin{bmatrix} 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 \end{bmatrix}$$

$$b = \begin{bmatrix} 12.259899186394588 \\ -19.33423583505869 \\ 19.836933582233584 \\ -11.949214892572499 \\ 5.684341886080802 \times 10^{-14} \\ 11.949214892572172 \\ -19.836933582233655 \\ 19.33423583505885 \\ -12.259899186394364 \end{bmatrix}$$

$$\alpha = \begin{bmatrix} 0.22198209041602027 \\ -0.47552825814755273 \\ 0.35917456718689444 \\ -0.293892626146188 \\ 6.779894115048364e - 14 \\ 0.2938926261463204 \\ -0.35917456718679086 \\ 0.47552825814767685 \\ -0.22198209041591366 \end{bmatrix}$$

1.1.4 Error L^∞

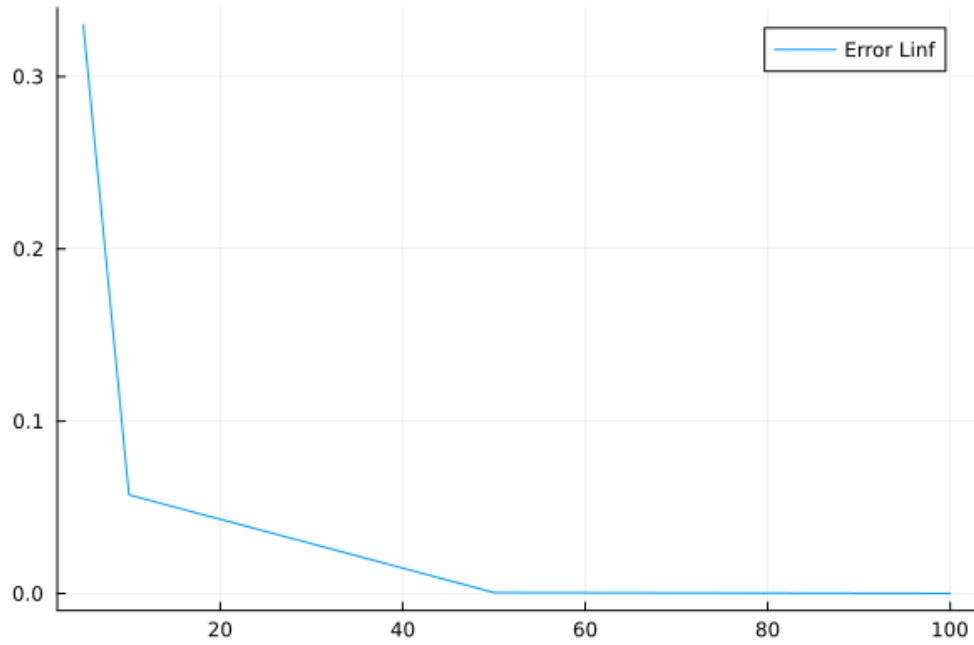


Figure 2: Gráfica del error con norma L^∞ para los valores de n evaluados

n	Error L^∞
5	0.330024
10	5.72439e-2
50	5.07059e-4
100	6.36111e-5

Table 2: Error L^∞ para $n \in [5, 10, 20, 50]$

Al aumentar el número de nodos en la malla (n), se observa una clara disminución en el error L^∞ , esto indica que el método de elementos finitos se está acercando a una solución más precisa a medida que se refinan las mallas. Se puede observar que el error disminuye en varios órdenes de magnitud a medida que aumenta n , esto sugiere que el método de elementos finitos utilizado tiene una convergencia de al menos segundo orden en este problema en particular. El valor más bajo del error L^∞ se obtiene con $n = 100$, lo que indica que esta malla refinada proporciona la solución más precisa en comparación con las otras mallas utilizadas en el estudio. Aunque se logra una mayor precisión al aumentar el número de nodos en la malla, también se incrementa el costo computacional del cálculo como indica la tabla (1). Por lo tanto, se debe encontrar un equilibrio entre la precisión requerida y la eficiencia computacional, eligiendo un número óptimo de nodos en la malla.

1.2

Sean $\Omega = (0, 1)$ y $t \in [0, T]$. Considere el problema de conductividad térmica: dada $u_0 \in L^2(\Omega)$ y $f(t, x)$ en un apropiado espacio de Sobolev, determinar la función $u(t, x)$ en un apropiado espacio de Sobolev (a saber, H^{-1} para el tiempo y H_0^1 para el espacio) que satisface (en el sentido débil):

$$\begin{cases} u_t(t, x) - u_{xx}(t, x) + u(t, x) = f(t, x) & (t, x) \in (0, T) \times \Omega \\ u(t, 0) = 0, u(t, 1) = 0 & t \in (0, T) \\ u(t = 0, x) = u_0(x) \end{cases} \quad (7)$$

- Considere una solución exacta no nula de la ecuación (7), la cual denotaremos por $u_{ex}(t, x)$ (cada uno de ustedes la propone).
- Encuentre la formulación variacional del problema (7) y describa tal formulación como un sistema matricial determinando de forma explícita las matrices respectivas.
- Use el inciso a) de este ejercicio para discretizar la parte espacial de (7) y simultáneamente use diferencias finitas para la parte temporal. Determinar el sistema matricial asociado a la formulación variacional y obtener la solución aproximada $u_h(t, x)$
- Posteriormente, fijar la cantidad de nodos espaciales (por ejemplo $n = 37$) y construir una tabla de error en función del tiempo que permita ver el cambio $\|u_{ex} - u_h\|$ a lo largo de los pasos temporales.

2) 1.5 puntos. Los elementos finitos de tipo \mathbb{P}_3 (Hermite) usan el espacio discreto siguiente:

1.2.1 Solución exacta

Una solución exacta no nula del problema es $u(t, x) = \sin(4\pi x)e^{-4\pi^2 t}$ con $x \in [0, 1]$. Esta cumple con las condiciones de frontera $u(t, x = 0) = u(t, x = 1) = 0$, es derivable con respecto a t y por lo menos dos veces derivable con respecto a x

- $u_t(t, x) = -4\pi^2 e^{-4\pi^2 t} \sin(4\pi x)$
- $u_x(t, x) = e^{-4\pi^2 t} \cos(4\pi x) \cdot 4\pi$
- $u_{xx}(t, x) = -16\pi^2 e^{-4\pi^2 t} \sin(4\pi x)$

1.2.2 Formulación variacional y sistema matricial

Sea $v(x)$ una función de prueba

$$u_t(t, x)v(x) - u_{xx}(t, x)v(x) + u(t, x)v(x) = f(t, x)v(x)$$

$$\int_0^1 u_t(t, x)v(x)dx - \int_0^1 u_{xx}(t, x)v(x)dx + \int_0^1 u(t, x)v(x)dx = \int_0^1 f(t, x)v(x)dx$$

Integrando por partes con:

$$\begin{aligned} u &= v(x) & dv &= u_{xx}(t, x)dx \\ du &= v_x(x)dx & v &= u_x(t, x) \end{aligned}$$

$$\begin{aligned} \int_0^1 u_t(t, x)v(x)dx - \left[u_x(t, x)v(x) \right]_0^1 - \int_0^1 u_x(t, x)v_x(x)dx + \int_0^1 u(t, x)v(x)dx &= \int_0^1 f(t, x)v(x)dx \\ \int_0^1 u_t(t, x)v(x)dx - \left[- \int_0^1 u_x(t, x)v_x(x)dx \right] + \int_0^1 u(t, x)v(x)dx &= \int_0^1 f(t, x)v(x)dx \\ \int_0^1 u_t(t, x)v(x)dx + \int_0^1 u_x(t, x)v_x(x)dx + \int_0^1 u(t, x)v(x)dx &= \int_0^1 f(t, x)v(x)dx \end{aligned} \quad (8)$$

Donde (8) corresponde a la formulación variacional para este problema. De forma similar al inciso anterior, definimos la solución aproximada como:

$$u_h(t, x) = \sum_{j=0}^{2n+1} \alpha_j(t) \lambda_j(x) \quad (9)$$

Que al reemplazar en (8) tenemos:

$$\begin{aligned} \int_0^1 \frac{d}{dt} \left(\sum_{j=0}^{2n+1} \alpha_j(t) \lambda_j(x) \right) \lambda_i(x) dx + \int_0^1 \frac{d}{dx} \left(\sum_{j=0}^{2n+1} \alpha_j(t) \lambda_j(x) \right) \frac{d}{dx} (\lambda_i(x)) dx + \int_0^1 \left(\sum_{j=0}^{2n+1} \alpha_j(t) \lambda_j(x) \right) \lambda_i(x) dx \\ = \int_0^1 f(t, x) \lambda_i(x) dx \\ \sum_{j=0}^{2n+1} \alpha_j'(t) \int_0^1 \lambda_j(x) \lambda_i(x) dx + \sum_{j=0}^{2n+1} \alpha_j(t) \int_0^1 \lambda_j'(x) \lambda_i'(x) dx + \sum_{j=0}^{2n+1} \alpha_j(t) \int_0^1 \lambda_j(x) \lambda_i(x) dx \\ = \int_0^1 f(t, x) \lambda_i(x) dx \\ \sum_{j=0}^{2n+1} \alpha_j'(t) \int_0^1 \lambda_j(x) \lambda_i(x) dx + \sum_{j=0}^{2n+1} \alpha_j(t) \left(\int_0^1 \lambda_j'(x) \lambda_i'(x) dx + \int_0^1 \lambda_j(x) \lambda_i(x) dx \right) \\ = \int_0^1 f(t, x) \lambda_i(x) dx \end{aligned}$$

La ecuación anterior puede ser escrita como un sistema matricial de la forma

$$\alpha'(t)M + \alpha(t)(K + M) = b \quad (10)$$

Donde K es la matriz cuyos elementos están dados por:

$$K_{ij} = \int_0^1 \lambda_j'(x) \lambda_i'(x) dx$$

M es la matriz cuyos elementos están dados por:

$$M_{ij} = \int_0^1 \lambda_j(x) \lambda_i(x) dx$$

b es el vector de carga cuyos elementos están dados por:

$$b_i = \int_0^1 f(t, x) \lambda_i(x) dx$$

Bastaria entonces con encontrar los vectores de coeficientes α y α' para poder calcular la solución aproximada.

1.2.3 Implementación del sistema matricial y solución aproximada

Los vectores α y α' pueden ser encontrados mediante diferencias finitas, donde $\alpha'(t+1) = \frac{\alpha(t+1) - \alpha(t)}{\Delta t}$. Veamos que para $t = t+1$

$$\begin{aligned} \alpha'(t+1)M + \alpha(t+1)(K + M) &= b \\ \left(\frac{\alpha(t+1) - \alpha(t)}{\Delta t} \right) M + \alpha(t+1)(K + M) &= b \\ \frac{1}{\Delta t} \left[\alpha(t+1)M - \alpha(t)M + \alpha(t+1)(K + M)\Delta t \right] &= b \\ \alpha(t+1)M - \alpha(t)M + \alpha(t+1)(K + M)\Delta t &= b\Delta t \\ \alpha(t+1)M + \alpha(t+1)(K + M)\Delta t &= b\Delta t + \alpha(t)M \\ \alpha(t+1)(M + K\Delta t + M\Delta t) &= b\Delta t + \alpha(t)M \end{aligned}$$

A continuación se mostraran los resultados obtenidos de la implementación para resolver el sistema matricial y encontrar la solución aproximada.

Se utilizó el software **Julia** con los paquetes *SymPy* para hacer cálculos simbólicos y *LinearSolve* para resolver los sistemas lineales. Para $n = 5$, $T = 1$ (tiempo final), $\Delta t = 0.1$ los coeficientes del sistema matricial son:

$$K = \begin{bmatrix} 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 & 1.66667 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 & -13.3333 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.66667 & -13.3333 & 23.3333 & -13.3333 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -13.3333 & 26.6667 \end{bmatrix}$$

Para la matriz M se redujeron las cifras significativas para que no se salga del margen

$$M = \begin{bmatrix} 0.107 & 0.013 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.013 & 0.053 & 0.013 & -0.007 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.013 & 0.107 & 0.013 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.007 & 0.013 & 0.053 & 0.013 & -0.007 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.013 & 0.107 & 0.013 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & -0.007 & 0.013 & 0.053 & 0.013 & -0.007 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.013 & 0.107 & 0.013 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -0.007 & 0.013 & 0.053 & 0.013 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.013 & 0.107 \end{bmatrix}$$

$$b = \begin{bmatrix} 13.2172507997187 \cdot \exp(-39.4784176043574t) \\ 4.72642428755876 \cdot \exp(-39.4784176043574t) \\ -9.04966988146695 \cdot \exp(-39.4784176043574t) \\ -8.6466053017786 \cdot \exp(-39.4784176043574t) \\ 1.42542265546108 \cdot \exp(-39.4784176043574t) \\ 9.2640769780241 \cdot \exp(-39.4784176043574t) \\ 6.74328757659694 \cdot \exp(-39.4784176043574t) \\ -6.34298612305945 \cdot \exp(-39.4784176043574t) \\ -12.3362911503095 \cdot \exp(-39.4784176043574t) \end{bmatrix}$$

La matriz de coeficientes $\alpha(t)$ es de tamaño 9x11, por lo que solo se muestran algunas columnas

$$\alpha = \begin{bmatrix} 0.93 & 0.07 & 0.00 & 0.00 & \dots & -3.83 \times 10^{-6} & -1.83 \times 10^{-6} & -8.78 \times 10^{-7} & -4.21 \times 10^{-7} \\ 0.65 & 0.04 & 0.00 & -7.38 \times 10^{-6} & \dots & -7.28 \times 10^{-6} & -3.49 \times 10^{-6} & -1.67 \times 10^{-6} & -8.01 \times 10^{-7} \\ -0.57 & -0.04 & -0.00 & -0.00 & \dots & -1.00 \times 10^{-5} & -4.80 \times 10^{-6} & -2.30 \times 10^{-6} & -1.10 \times 10^{-6} \\ -1.06 & -0.07 & -0.00 & -0.00 & \dots & -1.18 \times 10^{-5} & -5.64 \times 10^{-6} & -2.70 \times 10^{-6} & -1.30 \times 10^{-6} \\ 4.75 \times 10^{-13} & 0.00 & -0.00 & -0.00 & \dots & -1.24 \times 10^{-5} & -5.93 \times 10^{-6} & -2.84 \times 10^{-6} & -1.36 \times 10^{-6} \\ 1.06 & 0.07 & 0.00 & 2.16 \times 10^{-5} & \dots & -1.18 \times 10^{-5} & -5.64 \times 10^{-6} & -2.70 \times 10^{-6} & -1.30 \times 10^{-6} \\ 0.57 & 0.04 & 0.00 & -4.54 \times 10^{-5} & \dots & -1.00 \times 10^{-5} & -4.80 \times 10^{-6} & -2.30 \times 10^{-6} & -1.10 \times 10^{-6} \\ -0.65 & -0.04 & -0.00 & -0.00 & \dots & -7.28 \times 10^{-6} & -3.49 \times 10^{-6} & -1.67 \times 10^{-6} & -8.01 \times 10^{-7} \\ -0.93 & -0.07 & -0.00 & -0.00 & \dots & -3.83 \times 10^{-6} & -1.83 \times 10^{-6} & -8.78 \times 10^{-7} & -4.21 \times 10^{-7} \end{bmatrix}$$

1.2.4 Error L^∞

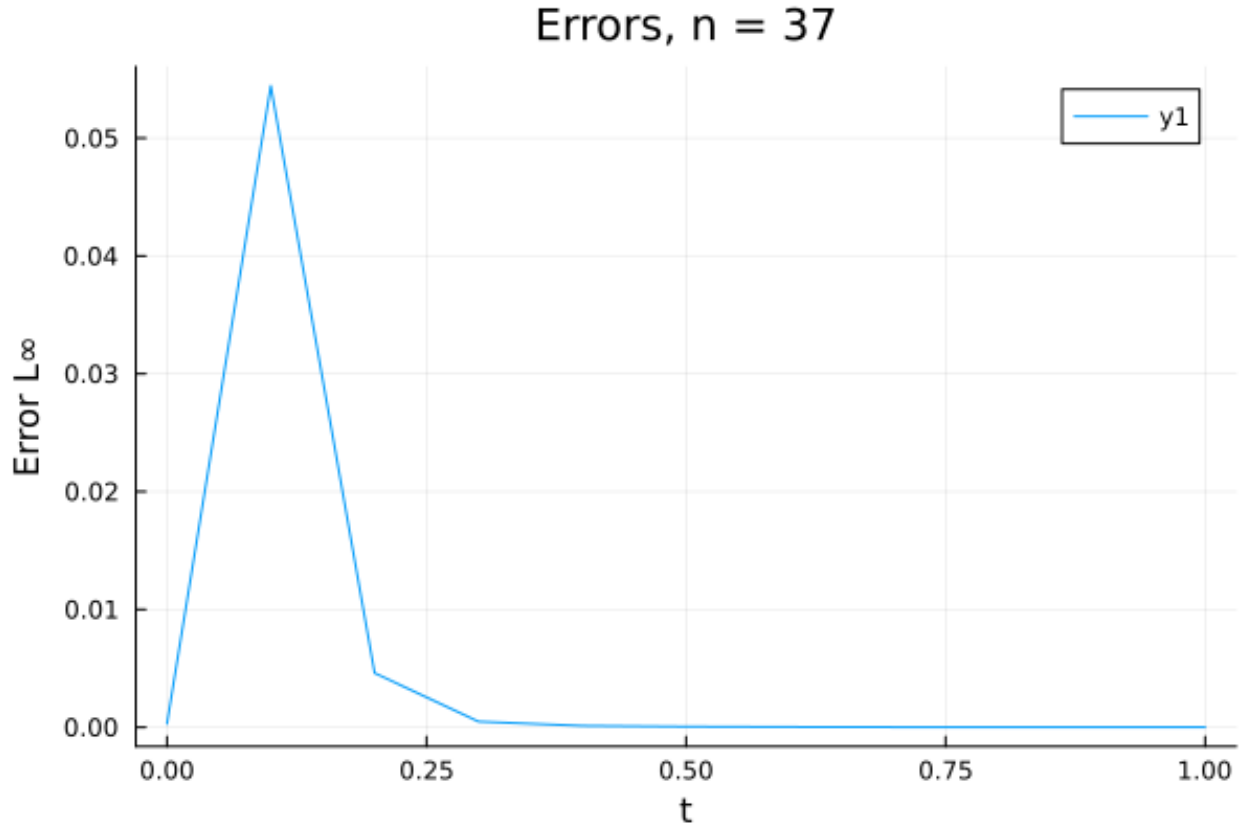


Figure 3: Comparación entre $u(x)$ y $u_h(x)$ para $t = 0.2$ con $n = 37$

Table 3: My Table

t	Error L^∞
0.0	0.000303924
0.1	0.0544387
0.2	0.00460039
0.3	0.000480634
0.4	0.000122177
0.5	5.41443e-5
0.6	2.59422e-5
0.7	1.24364e-5
0.8	5.95974e-6
0.9	2.85577e-6
1.0	1.36839e-6

El error L^∞ se calcula en cada paso temporal y muestra la norma máxima del error en todo el dominio. A medida que avanzamos en el tiempo, el error tiende a disminuir, lo que indica que la solución numérica se acerca cada vez más a la solución exacta de la ecuación en derivadas parciales (EDP). El error es relativamente pequeño en general, lo que sugiere que el método de elementos finitos está proporcionando una buena aproximación de la solución. Sin embargo, se observa un aumento en el error en algunos pasos

temporales (por ejemplo, en $t = 0.0$ y $t = 0.1$) en comparación con los pasos adyacentes. Esto puede indicar una mayor sensibilidad numérica o inestabilidad en esos puntos temporales específicos. Es posible que se necesiten técnicas de refinamiento temporal o ajustes en la discretización temporal para mejorar la precisión en esos casos.

A medida que avanzamos en el tiempo, el error disminuye significativamente y se estabiliza a un nivel bajo. Esto sugiere que la solución numérica ha alcanzado una convergencia temporal y la solución aproximada se mantiene cercana a la solución exacta en los pasos posteriores.

2

Los elementos finitos de tipo P3 (Hermite) usan el espacio discreto siguiente:

$V_h := \left\{ v \in C^1([a, b]) : v|_{[x_j, x_{j+1}]} \in \mathbb{P}_3, 0 \leq j \leq n \right\}$ y su subespacio $V_{0h} := \{v \in V_h : v(a) = v(b) = 0, v'(a) = v'(b) = 0\}$.

A partir de lo anterior, se definen las funciones de referencia $\phi(x), \psi(x)$ como sigue:

$$\phi(x) := \begin{cases} (1+x)^2(1-2x) & \text{si } -1 \leq x \leq 0 \\ (1-x)^2(1+2x) & \text{si } 0 \leq x \leq 1 \\ 0 & \text{si } |x| > 1, \end{cases} \quad \psi(x) := \begin{cases} x(1+x)^2 & \text{si } -1 \leq x \leq 0 \\ x(1-x)^2 & \text{si } 0 \leq x \leq 1 \\ 0 & \text{si } |x| > 1. \end{cases}$$

Sea $\Omega = (0, 1)$. Considere el problema de placas siguiente: Dada $f \in L^2(\Omega)$, determinar la función $u \in H_0^2(\Omega)$ que satisface (en el sentido débil):

$$\begin{cases} u_{xxx}(x) = f(x) & x \in \Omega \\ u(0) = 0, u(1) = 0 \\ u'(0) = 0, u'(1) = 0 \end{cases} \quad (11)$$

Para una malla uniforme sobre $[0, 1]$, las funciones base son:

$\phi_j(x) := \phi\left(\frac{x-x_j}{h}\right)$ para $j = 0, \dots, n+1$, $\psi_j(x) = \psi\left(\frac{x-x_{j+1/2}}{h}\right)$ para $j = 0, \dots, n+1$.

Además, es sabido que la solución aproximada u_h es de la forma

$$u_h(x) = \sum_{j=0}^{n+1} u_h(x_j) \phi_j(x) + \sum_{j=0}^{n+1} (u_h)'(x_j) \psi_j(x), \quad (u_h)' \equiv du_h/dx \quad (12)$$

- Considere una solución exacta no nula de la ecuación (11), la cual denotaremos por $u_{ex}(x)$ (cada uno de ustedes la propone).
- Encuentre la formulación variacional del problema (11) y describa tal formulación como un sistema matricial determinando de forma explícito las matrices respectivas.
- Implemente el sistema matricial obtenido en el inciso anterior y obtener la solución aproximada $u_h(x)$.
- Posteriormente, construya una tabla de error en función de la cantidad de nodos, es decir, puede considerar particiones del dominio con $n = 5, 10, 20, 50, 100$ y construir la tabla asociada a $\|u_{ex} - u_h\|$.

Nota: La norma del error puede ser $L^2(\Omega)$ o $L^\infty(\Omega)$.

2.1 Solución exacta

Una solución exacta no nula del problema es $u(x) = \sin(4\pi x)(1 - \cos(4\pi x))$ con $x \in [0, 1]$. Esta cumple con las condiciones de frontera $u(x=0) = u(x=1) = 0$ y $u'(x=0) = u'(x=1) = 0$, y es por lo menos tres veces derivable con respecto a x .

- $u_x(x) = 4\pi \cos(4\pi x) (1 - \cos(4\pi x)) + 4\pi \sin^2(4\pi x)$
- $u_{xx}(x) = -16\pi^2 \sin(4\pi x) + 64\pi^2 \sin(4\pi x) \cos(4\pi x)$
- $u_{xxx}(x) = -64\pi^3 \cos(4\pi x) + 256\pi^3 \cos(8\pi x)$

2.2 Formulación variacional y sistema matricial

Sea $v(x)$ una función de prueba

$$u_{xxx}(x)v(x) = f(x)v(x)$$

$$\int_0^1 u_{xxx}(x)v(x)dx = \int_0^1 f(x)v(x)dx$$

Integrando por partes con:

$$\begin{aligned} u &= v(x) & dv &= u_{xxx}(x)dx \\ du &= v_x(x)dx & v &= u_{xx}(x) \end{aligned}$$

$$u_x(x)v(x) \Big|_0^1 - \int_0^1 u_{xx}(x)v_x(x)dx = \int_0^1 f(x)v(x)dx$$

$$- \int_0^1 u_{xx}(x)v_x(x)dx = \int_0^1 f(x)v(x)dx \quad (13)$$

Donde (13) corresponde a la formulación variacional del problema. La ecuación (14) puede reescribirse como:

$$u_h(x) = \sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \quad (14)$$

Las funciones base dadas forman un conjunto completo en el espacio, por lo que podemos considerar $v(x) = \phi_j(x)$ y $v(x) = \psi_j(x)$

Si tomamos $v(x) = \phi_j(x)$ y reemplazando (14) en (13) obtenemos:

$$\begin{aligned} & - \int_0^1 \frac{d^2}{dx^2} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\phi_i(x)) dx = \int_0^1 f(x) \phi_i(x) dx \\ & - \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \right) \phi_i'(x) dx = \int_0^1 f(x) \phi_i(x) dx \\ & - \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) \phi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \phi_i'(x) \right) dx = \int_0^1 f(x) \phi_i(x) dx \\ & - \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j''(x) \phi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j''(x) \phi_i'(x) dx = \int_0^1 f(x) \phi_i(x) dx \end{aligned}$$

La ecuación anterior puede ser escrita como un sistema matricial de la forma

$$-\alpha_1 K_1 - \alpha_2 M_1 = b_1 \quad (15)$$

$$K_{ij} = \int_0^1 \phi_j''(x) \phi_i'(x) dx \quad M_{ij} = \int_0^1 \psi_j''(x) \phi_i'(x) dx \quad b_i = \int_0^1 f(x) \phi_i(x) dx$$

Si tomamos $v(x) = \psi_j(x)$ y reemplazando (14) en (13) obtenemos:

$$\begin{aligned}
& - \int_0^1 \frac{d^2}{dx^2} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\psi_i(x)) dx = \int_0^1 f(x) \psi_i(x) dx \\
& - \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \right) \psi_i'(x) dx = \int_0^1 f(x) \psi_i(x) dx \\
& - \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) \psi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \psi_i'(x) \right) dx = \int_0^1 f(x) \psi_i(x) dx \\
& - \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j''(x) \psi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j''(x) \psi_i'(x) dx = \int_0^1 f(x) \psi_i(x) dx
\end{aligned}$$

La ecuación anterior puede ser escrita como un sistema matricial de la forma

$$-\alpha_1 K_2 - \alpha_2 M_2 = b_2 \quad (16)$$

$$K_{ij} = \int_0^1 \phi_j''(x) \phi_i'(x) dx \quad M_{ij} = \int_0^1 \psi_j''(x) \psi_i'(x) dx \quad b_i = \int_0^1 f(x) \psi_i(x) dx$$

Luego poseemos ensamblar el siguiente sistema lineal con (15) y (16)

$$- \begin{bmatrix} K_1 & M_1 \\ K_2 & M_2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Basta entonces encontrar el vector de los α para poder encontrar la solución aproximada

2.3 Implementación del sistema matricial y solución aproximada

A continuación se mostraran los resultados obtenidos de la implementación para resolver el sistema matricial y encontrar la solución aproximada.

Se utilizo el software **Julia** con los paquetes *SymPy* para hacer cálculos simbólicos y *LinearSolve* para resolver los sistemas lineales.

Como puede verse en la figura (4), la solución aproximada u_h mejora su ajuste con la solución exacta $u(x)$ a medida que se incrementa el número de nodos (n) para resolver el sistema matricial y resolver la formulación variacional del problema, pero a medida que el número de nodos crece, el tiempo de computo también lo hace.

n	Tiempo de computo
5	00:08
10	00:32
50	13:29
100	53:33

Table 4: Tiempo de computo para $n \in [5, 10, 20, 50]$

Notese que aunque el algoritmo tiene una complejidad mayor que el del punto 1, este se ejecuto con una malla de tamaño $n-1$ para facilitar el computo y aun así presenta tiempos de computo similares al punto anterior.

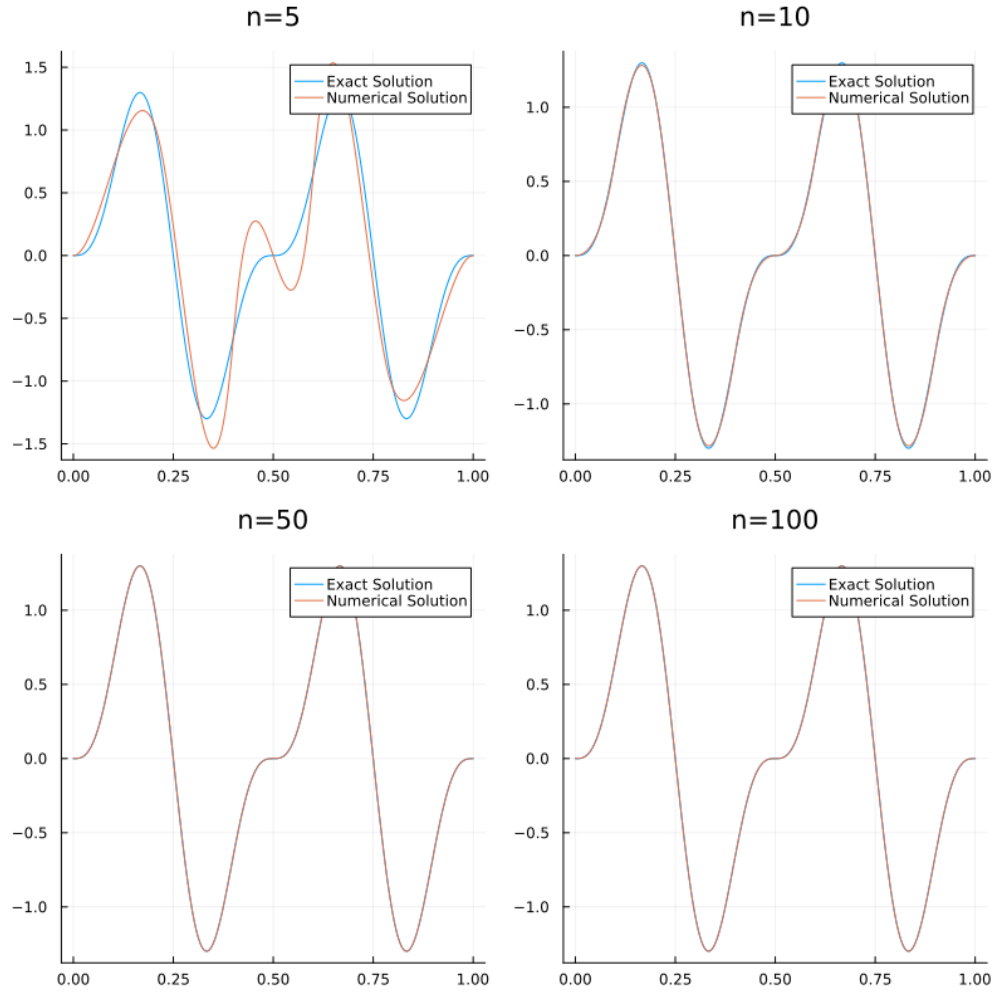


Figure 4: Comparación entre $u(x)$ y $u_h(x)$ para diferentes valores de n

A continuación se encuentran los coeficientes del sistema matricial correspondiente a $n = 5$.

$$K_1 = \begin{bmatrix} -1.36424 \times 10^{-12} & 1.81899 \times 10^{-12} & 0.0 & 0.0 \\ 0.0 & 6.82121 \times 10^{-12} & 1.45519 \times 10^{-11} & 0.0 \\ 0.0 & 1.45519 \times 10^{-11} & -2.18279 \times 10^{-11} & 0.0 \\ 0.0 & 0.0 & 0.0 & -2.91038 \times 10^{-11} \end{bmatrix}$$

$$K_2 = \begin{bmatrix} -50.0 & 25.0 & 0.0 & 0.0 \\ 25.0 & -50.0 & 25.0 & 0.0 \\ 0.0 & 25.0 & -50.0 & 25.0 \\ 0.0 & 0.0 & 25.0 & -50.0 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 50.0 & -25.0 & 0.0 & 0.0 \\ -25.0 & 50.0 & -25.0 & 0.0 \\ 0.0 & -25.0 & 50.0 & -25.0 \\ 0.0 & 0.0 & -25.0 & 50.0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -2.27374 \times 10^{-13} & -12.5 & 0.0 & 0.0 \\ 12.5 & 1.7053 \times 10^{-12} & -12.5 & 0.0 \\ 0.0 & 12.5 & -1.81899 \times 10^{-11} & -12.5 \\ 0.0 & 0.0 & 12.5 & -3.63798 \times 10^{-11} \end{bmatrix}$$

$$b_1 = \begin{bmatrix} -263.5602550092512 \\ 227.74178915706352 \\ 227.74178915712037 \\ -263.560255009259 \end{bmatrix} \quad b_2 = \begin{bmatrix} -165.55643442817004 \\ -38.00076505636798 \\ 38.000765056369346 \\ 165.5564344281256 \end{bmatrix}$$

$$\alpha_1 = \begin{bmatrix} 1.063313510439197 \\ -0.6571638901519665 \\ 0.6571638901424814 \\ -1.0633135104416027 \end{bmatrix} \quad \alpha_2 = \begin{bmatrix} -1.4327386340882358 \\ 7.6769329321933055 \\ 7.676932932194788 \\ -1.432738634088615 \end{bmatrix}$$

2.4 Error L^∞

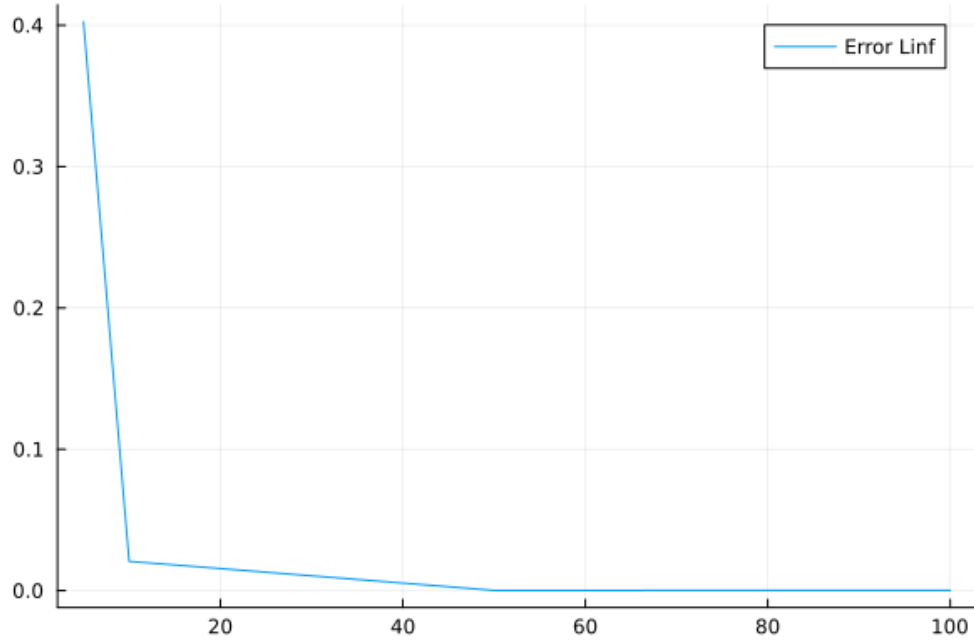


Figure 5: Gráfica del error con norma L^∞ para los valores de n evaluados

n	Error L^∞
5	0.402417
10	0.0207152
50	0.000286928
100	0.000180101

Table 5: Error L^∞ para $n \in [5,10,20,50]$

Como muestra la tabla, el error L^∞ disminuye a medida que aumenta el número de nodos en la malla. Esto indica que a medida que se refinan las mallas (es decir, se aumenta la resolución), se obtiene una solución

más precisa. Al comparar los errores para $n = 50$ y $n = 100$, se puede ver que el decremento en el error no es tan significativo. Esto puede indicar que a partir de un cierto número de nodos, el aumento adicional en la resolución no tiene un impacto significativo en la precisión de la solución. Sin embargo, es importante tener en cuenta que hay un punto de equilibrio entre la precisión deseada y el costo computacional, ya que aumentar demasiado el número de nodos puede llevar a un aumento significativo en el tiempo de cálculo y los recursos necesarios.

3

Sea $\Omega = (0, 1)$. Considere el problema siguiente: Dada $f \in L^2(\Omega)$, determinar la función $u \in H_0^2(\Omega)$ que satisface (en el sentido débil):

$$\begin{cases} u_{xxx}(x) - u_{xx} = f(x) & x \in \Omega \\ u(0) = 0, u(1) = 0 \\ u'(0) = 0, u'(1) = 0 \end{cases}$$

(a) Considere una solución exacta no nula de la ecuación (3), la cual denotaremos por $u_{ex}(x)$ (cada uno de ustedes la propone).

(b) Para una malla uniforme sobre $[0, 1]$, implemente y resuelva el sistema matricial que surge de la formulación variacional al usando la base (4) sobre la ecuación (5). Escriba la solución aproximada $u_h(x)$.

(c) En función del número de nodos n (por ejemplo, $n = 5, 10, 20, 50, 100$), realice una tabla y compare los resultados en la norma $\|u_{ex} - u_h\|$. Comente los resultados obtenidos.

Nota: La norma del error puede ser $L^2(\Omega)$ o $L^\infty(\Omega)$.

3.1 Solución exacta

Una solución exacta no nula del problema es $u(x) = \sin(4\pi x)(1 - \cos(4\pi x))$ con $x \in [0, 1]$. Esta cumple con las condiciones de frontera $u(x = 0) = u(x = 1) = 0$ y $u'(x = 0) = u'(x = 1) = 0$, y es por lo menos tres veces derivable con respecto a x .

- $u_x(x) = 4\pi \cos(4\pi x)(1 - \cos(4\pi x)) + 4\pi \sin^2(4\pi x)$
- $u_{xx}(x) = -16\pi^2 \sin(4\pi x) + 64\pi^2 \sin(4\pi x) \cos(4\pi x)$
- $u_{xxx}(x) = -64\pi^3 \cos(4\pi x) + 256\pi^3 \cos(8\pi x)$

3.2 Formulación variacional y sistema matricial

Sea $v(x)$ una función de prueba

$$\begin{aligned} u_{xxx}(x)v(x) - u_{xx}v(x) &= f(x)v(x) \\ \int_0^1 u_{xxx}(x)v(x) - u_{xx}v(x)dx &= \int_0^1 f(x)v(x)dx \end{aligned} \quad (17)$$

De los resultados de los puntos anteriores sabemos que la ecuación 17 puede reescribirse como a:

$$-\int_0^1 u_{xx}(x)v_x(x)dx + \int_0^1 u_x(x)v_x(x)dx = \int_0^1 f(x)v(x)dx \quad (18)$$

Donde (18) corresponde a la formulación variacional del problema. Y empleando las mismas funciones de referencia y la misma aproximación u_h del punto 2 (la ecuación 14) haremos un análisis análogo para este problema.

Si tomamos $v(x) = \phi_j(x)$ y reemplazando (14) en (18) obtenemos:

$$\begin{aligned}
& - \int_0^1 \frac{d^2}{dx^2} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\phi_i(x)) dx + \int_0^1 \frac{d}{dx} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\phi_i(x)) dx \\
& = \int_0^1 f(x) \phi_i(x) dx
\end{aligned}$$

$$- \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \right) \phi_i'(x) dx + \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j'(x) \right) \phi_i'(x) dx = \int_0^1 f(x) \phi_i(x) dx$$

$$\begin{aligned}
& - \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) \phi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \phi_i'(x) \right) dx + \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j'(x) \phi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j'(x) \phi_i'(x) \right) dx \\
& = \int_0^1 f(x) \phi_i(x) dx
\end{aligned}$$

$$\begin{aligned}
& - \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j''(x) \phi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j''(x) \phi_i'(x) dx + \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j'(x) \phi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j'(x) \phi_i'(x) dx \\
& = \int_0^1 f(x) \phi_i(x) dx
\end{aligned}$$

$$\sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 [\phi_j'(x) \phi_i'(x) - \phi_j''(x) \phi_i'(x)] dx + \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 [\psi_j'(x) \phi_i'(x) - \psi_j''(x) \phi_i'(x)] dx = \int_0^1 f(x) \phi_i(x) dx$$

La ecuación anterior puede ser escrita como un sistema matricial de la forma

$$-\alpha_1 K_1 - \alpha_2 M_1 = b_1 \tag{19}$$

$$K_{ij} = \int_0^1 [\phi_j'(x) \phi_i'(x) - \phi_j''(x) \phi_i'(x)] dx \quad M_{ij} = \int_0^1 [\psi_j'(x) \phi_i'(x) - \psi_j''(x) \phi_i'(x)] dx \quad b_i = \int_0^1 f(x) \phi_i(x) dx$$

Si tomamos $\mathbf{v}(\mathbf{x}) = \boldsymbol{\psi}_j(\mathbf{x})$ y reemplazando (14) en (18) obtenemos:

$$- \int_0^1 \frac{d^2}{dx^2} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\psi_i(x)) dx + \int_0^1 \frac{d}{dx} \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j(x) \right) \frac{d}{dx} (\psi_i(x)) dx$$

$$= \int_0^1 f(x)\psi_i(x)dx$$

$$- \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \right) \psi_i'(x) dx + \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j'(x) \right) \psi_i'(x) dx = \int_0^1 f(x)\psi_i(x)dx$$

$$- \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j''(x) \psi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j''(x) \psi_i'(x) \right) dx + \int_0^1 \left(\sum_{j=0}^{n+1} \alpha_{1j} \phi_j'(x) \psi_i'(x) + \sum_{j=0}^{n+1} \alpha_{2j} \psi_j'(x) \psi_i'(x) \right) dx$$

$$= \int_0^1 f(x)\psi_i(x)dx$$

$$- \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j''(x) \psi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j''(x) \psi_i'(x) dx + \sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 \phi_j'(x) \psi_i'(x) dx - \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 \psi_j'(x) \psi_i'(x) dx$$

$$= \int_0^1 f(x)\psi_i(x)dx$$

$$\sum_{j=0}^{n+1} \alpha_{1j} \int_0^1 [\phi_j'(x) \psi_i'(x) - \phi_j''(x) \psi_i'(x)] dx + \sum_{j=0}^{n+1} \alpha_{2j} \int_0^1 [\psi_j'(x) \psi_i'(x) - \psi_j''(x) \psi_i'(x)] dx = \int_0^1 f(x)\psi_i(x)dx$$

La ecuación anterior puede ser escrita como un sistema matricial de la forma

$$-\alpha_1 K_2 - \alpha_2 M_2 = b_2 \quad (20)$$

$$K_{ij} = \int_0^1 \phi_j''(x) \phi_i'(x) dx \quad M_{ij} = \int_0^1 \psi_j''(x) \psi_i'(x) dx \quad b_i = \int_0^1 f(x) \psi_i(x) dx$$

Luego poseemos ensamblar el siguiente sistema lineal con (19) y (20)

$$- \begin{bmatrix} K_1 & M_1 \\ K_2 & M_2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Basta entonces encontrar el vector de los α para poder encontrar la solución aproximada

3.3 Implementación del sistema matricial y solución aproximada

A continuación se mostraran los resultados obtenidos de la implementación para resolver el sistema matricial y encontrar la solución aproximada.

Se utilizo el software **Julia** con los paquetes *SymPy* para hacer cálculos simbólicos y *LinearSolve* para resolver los sistemas lineales.

Como puede verse en la figura (6), la solución aproximada u_h mejora su ajuste con la solución exacta $u(x)$ a medida que se incrementa el número de nodos (n) para resolver el sistema matricial y resolver la formulación variacional del problema, pero a medida que el número de nodos crece, el tiempo de cómputo también lo hace.

n	Tiempo de cómputo
5	00:16
10	01:22
50	37:51
100	02:35:25

Table 6: Tiempo de cómputo para $n \in [5, 10, 20, 50]$

Notese que aunque el algoritmo tiene una complejidad mayor que el del punto 2, este se ejecuto con una malla de tamaño $n-1$ para facilitar el cómputo.

A continuación se encuentran los coeficientes del sistema matricial correspondiente a $n = 5$.

$$\begin{aligned}
 K_1 &= \begin{bmatrix} 12.0 & -6.0 & 0.0 & 0.0 \\ -6.0 & 12.0 & -6.0 & 0.0 \\ 0.0 & -6.0 & 12.0 & -6.0 \\ 0.0 & 0.0 & -6.0 & 12.0 \end{bmatrix} & K_2 &= \begin{bmatrix} 50.0 & -25.5 & 0.0 & 0.0 \\ -24.5 & 50.0 & -25.5 & 0.0 \\ 0.0 & -24.5 & 50.0 & -25.5 \\ 0.0 & 0.0 & -24.5 & 50.0 \end{bmatrix} \\
 M_1 &= \begin{bmatrix} -50.0 & 25.5 & 0.0 & 0.0 \\ 24.5 & -50.0 & 25.5 & 0.0 \\ 0.0 & 24.5 & -50.0 & 25.5 \\ 0.0 & 0.0 & 24.5 & -50.0 \end{bmatrix} & M_2 &= \begin{bmatrix} 1.33333 & 12.3333 & 0.0 & 0.0 \\ -12.6667 & 1.33333 & 12.3333 & 0.0 \\ 0.0 & -12.6667 & 1.33333 & 12.3333 \\ 0.0 & 0.0 & -12.6667 & 1.33333 \end{bmatrix} \\
 b_1 &= \begin{bmatrix} 282.47175188382766 \\ -242.78051610035288 \\ -212.70306221382953 \\ 244.64875813474828 \end{bmatrix} & b_2 &= \begin{bmatrix} 160.63744948717311 \\ 43.834068158437276 \\ -32.167461954295504 \\ -170.475419369086 \end{bmatrix} \\
 \alpha_1 &= \begin{bmatrix} 0.888242841027757 \\ -0.9364477969668461 \\ 0.35999049173275677 \\ -1.2749787338992893 \end{bmatrix} & \alpha_2 &= \begin{bmatrix} -1.4265999531013558 \\ 7.641731504102811 \\ 7.568020843761932 \\ -1.5338387043858779 \end{bmatrix}
 \end{aligned}$$

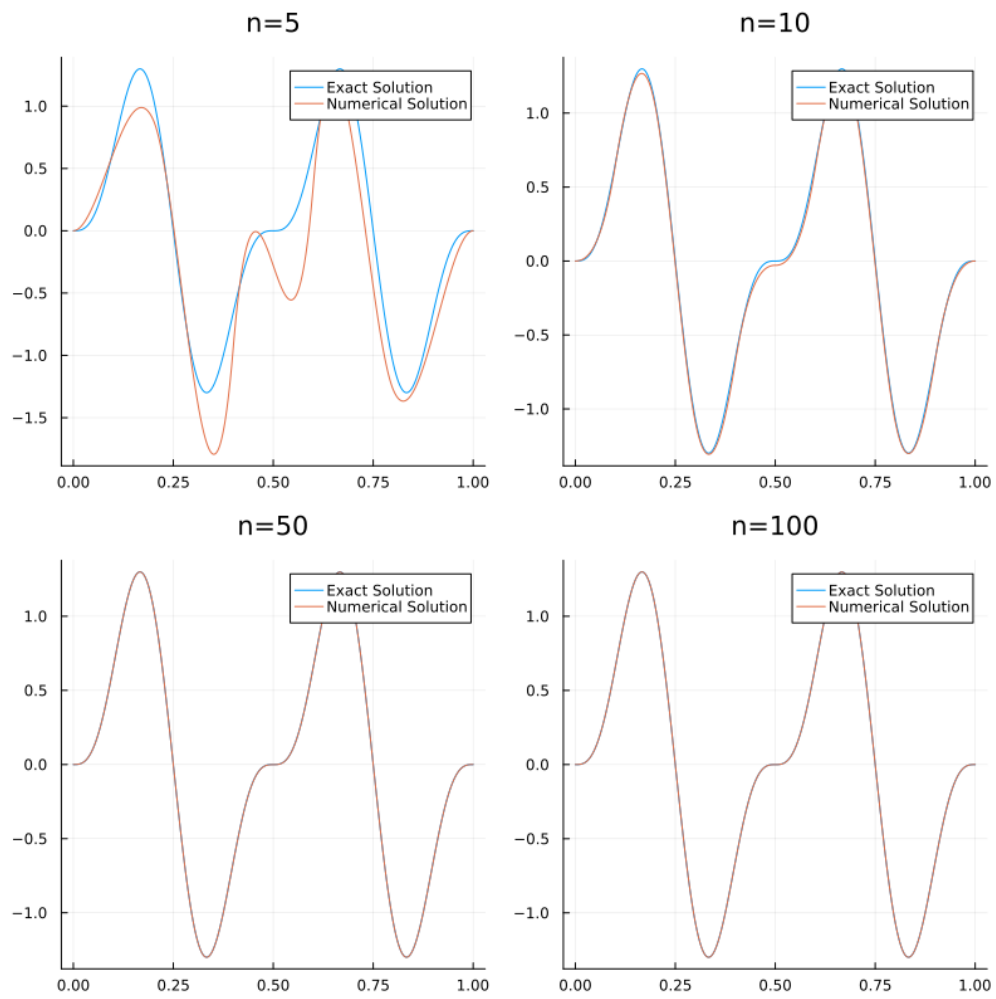


Figure 6: Comparación entre $u(x)$ y $u_h(x)$ para diferentes valores de n

3.4 Error L^∞

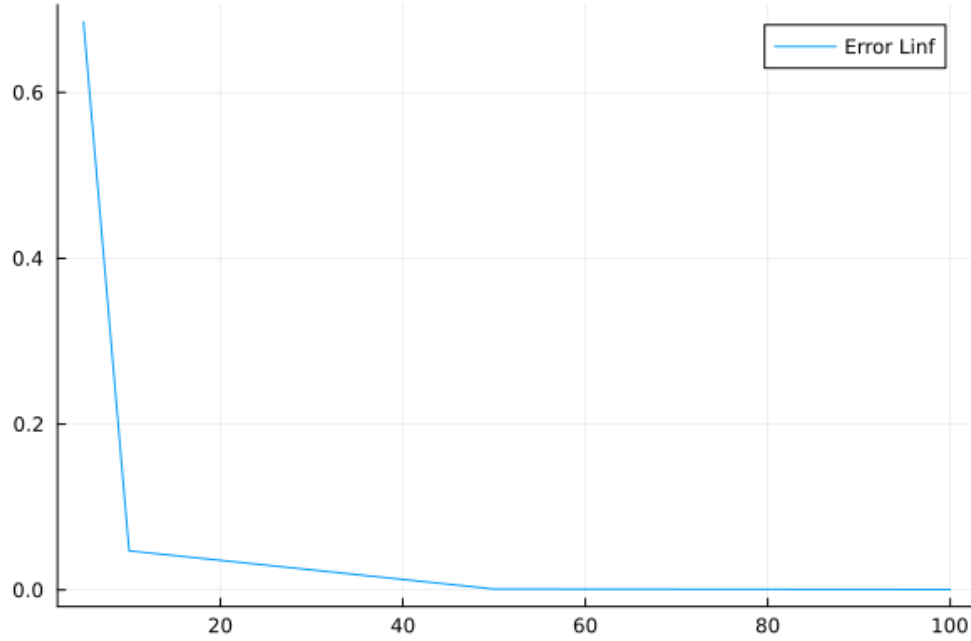


Figure 7: Gráfica del error con norma L^∞ para los valores de n evaluados

n	Error L^∞
5.0	0.685253
10.0	0.0471549
50.0	0.00112675
100.0	0.000581068

Table 7: Error L^∞ para $n \in [5,10,20,50]$

Como muestra la tabla, a medida que el número de nodos en la malla aumenta, el error L^∞ disminuye. Esto indica que una mayor densidad de nodos en la malla proporciona una aproximación más precisa de la solución exacta. Al igual que en el punto 2, la disminución del error a medida que aumenta el número de nodos sugiere una convergencia del método de elementos finitos. En este caso, se observa una disminución significativa del error cuando se pasa de 5 a 10 nodos, y una disminución adicional al aumentar a 50 y 100 nodos. Esto indica que el método está convergiendo hacia la solución exacta a medida que se refinan las mallas.

4 Conclusión

Como se pudo ver a lo largo de este trabajo, el método de elementos finitos (FEM) implica un trade-off entre el costo computacional y la precisión de la solución encontrada. El costo computacional está influenciado principalmente por el número de elementos en la malla y el tamaño del sistema de ecuaciones a resolver. A medida que se aumenta el número de elementos y se refinan las mallas, el costo computacional aumenta debido al incremento en la cantidad de cálculos y la complejidad del sistema a resolver. Pero medida que se aumenta el número de elementos y se refinan las mallas, la solución obtenida converge hacia la solución

exacta. Es importante encontrar un equilibrio adecuado y optimizar los recursos, esto implica evaluar los requisitos de precisión del problema y los recursos computacionales disponibles. En algunos casos, puede ser suficiente utilizar una malla menos refinada y una solución aproximada para obtener resultados aceptables sin incurrir en un alto costo computacional. En otros casos, cuando se requiere una alta precisión, puede ser necesario invertir en más recursos computacionales para obtener una solución más precisa.

5 Requisitos para correr el código

Se necesita de ambos programas, Julia y Python, pues aunque los calculos y el script son hechos en julia, se hace uso de la libreria *sympy* de python para el calculo simbolico.

5.1 Julia 1.9

- Logging
- ProgressBars v1.5.0
- LinearSolve v1.42.0
- Cubature v1.5.1
- BenchmarkTools v1.3.2
- Plots v1.38.13
- SymPy v1.1.8

5.2 Python 3.11.3

- sympy 1.12