

# Aceleración en la resolución de modelos transitorios de transferencia de calor mediante un software de elementos finitos en 2d y CUDA

J. A. Beltrán-Pérez<sup>1\*</sup>, J. M. Díaz-Chacón<sup>1</sup>, C. Amaro-Hernández<sup>1</sup>, A. Navarro-Valle<sup>1</sup>,  
P. F. Rosales-Escobedo<sup>1</sup>

**Resumen**— Desde hace varias décadas, una de las metodologías numéricas más empleadas para el modelado de diversos fenómenos físicos es el Método de los Elementos Finitos (MEF). Sin embargo, una de las principales desventajas en el uso de esta técnica es el elevado tiempo de solución, requerido en el análisis de fenómenos transitorios. En este artículo se presenta la implementación de un solver en CUDA (Compute Unified Device Architecture) incorporado a un software de Elementos Finitos en Dos Dimensiones (2D), desarrollado en el lenguaje de programación C++, con el objetivo de acelerar el tiempo de resolución en el análisis de modelos transitorios de transferencia de calor. Como ejemplos demostrativos se simuló dos modelos transitorios de conducción de calor, tomados de la literatura. El software Gmsh se utiliza para generar las mallas triangulares de segundo orden de estos modelos y para realizar la visualización de los resultados obtenidos por el software desarrollado, en la etapa de post-procesamiento. Finalmente, se lleva a cabo un análisis comparativo entre el solver en CUDA y el solver MKL PARDISO. Los resultados obtenidos con el solver realizado en CUDA demuestran que es posible reducir considerablemente, el tiempo de resolución de modelos transitorios de transferencia de calor.

**Palabras claves**—CUDA, Método de los Elementos Finitos, Software, Transferencia de calor.

**Abstract**— For several decades, one of the most used numerical methodologies for the model of various physical phenomena is the Finite Element Method (FEM). However, one of the main disadvantages in the use of this technique is the long solution time, required in the analysis of transient phenomena. This article presents the implementation of a solver in CUDA (Compute Unified Device Architecture) incorporated into a Finite Element in Two Dimensions (2D) software, developed in the C++ programming language, with the aim of accelerating the resolution time in the analysis of transient heat transfer models. As demonstrative examples, two transient heat conduction models were simulated, taken from the literature. The Gmsh software is used to generate the second order triangular meshes of these models and to visualize the results obtained by the developed software, in the post-processing stage. Finally, a comparative performance analysis between the solver CUDA and the solver MKL PARDISO is carried out. The results obtained with the solver

made in CUDA show that it is possible to considerably reduce the resolution time of transient heat transfer models.

**Keywords**— CUDA, Finite Element Method, heat transfer, software.

## I. INTRODUCCIÓN

Durante el proceso de diseño de las máquinas eléctricas tales como motores, generadores o transformadores, es requerido determinar los puntos calientes que se generan durante el proceso de transferencia de calor debido a que los materiales que lo componen, tienen un límite térmico, que al ser superado, puede producir un cortocircuito que compromete seriamente la seguridad del personal a su alrededor y reduce, de manera significativa, su vida útil. Por tal motivo, el conocimiento previo de su comportamiento permite incorporar materiales y configuraciones que sean capaces de soportar las temperaturas generadas o bien, de incorporar sistemas de enfriamiento que disminuyan la temperatura, aumentando la eficiencia y vida útil de los dispositivos eléctricos [1]-[4]. Para el modelado de esta clase de fenómenos se tiene una representación que es descrita matemáticamente por ecuaciones diferenciales parciales. Sin embargo, su solución es sumamente compleja.

En las últimas décadas ha tomado una gran relevancia, el desarrollo de técnicas numéricas para la obtención de resoluciones aproximadas de fenómenos físicos. En el área eléctrica, una de las técnicas numéricas más utilizadas es el Método de los Elementos Finitos (MEF) [5]-[7].

Sin embargo, la principal desventaja en el uso de este método es el elevado tiempo de cómputo requerido para determinar la solución, en especial en el modelado de fenómenos transitorios. Una alternativa de mejora, ante esta problemática, es el uso de hardware y de software especializado en paralelismo, tales como CUDA, que permitan reducir el tiempo requerido de simulación de esta clase de modelos [8]-[11].

Por tal motivo, en este artículo se realiza la simulación de dos modelos de conducción de calor mediante un software de elementos finitos, el cual tiene implementado un solver

<sup>1</sup> Universidad Autónoma de Baja California, Av. Álvaro Obregón s/n, Nueva, C. P. 21100, Mexicali, Baja California, México.  
\*(jose.angel.beltran.perez@uabc.edu.mx)

paralelizado de CUDA, capaz de reducir el tiempo de resolución requerido.

## II. MODELADO NUMÉRICO DE FENÓMENOS TRANSITORIOS DE CONDUCCIÓN DE CALOR

### A. Modelo matemático de conducción de calor

Para determinar la distribución de calor en un material, en función del tiempo, se utiliza la ley de Fourier, la cual establece el tiempo de transferencia de calor, a través de las características térmicas del mismo, y está dada por:

$$-k\nabla T = \mathbf{q} \quad (1)$$

donde:

$\mathbf{q}$  = Flujo de calor ( $W/m^2$ ).

$k$  = Conductividad térmica del material ( $W/mK$ ).

$\nabla T$  = Gradiente de temperatura ( $K/m$ ).

La proporcionalidad de la variación de energía interna por unidad de volumen en el material y el calor generado están definidos por:

$$\nabla Q = \rho c \nabla T \quad (2)$$

donde  $Q$  es el flujo de calor externamente aplicado,  $\rho$  es la densidad de masa y  $c$  es el calor específico.

Utilizando (1) y (2) es posible determinar la ecuación diferencial general de conducción de calor, para un medio estacionario en dos dimensiones, como se muestra en (3), la cual, describe la distribución inestable de temperatura en un dominio  $\Omega$ , como se muestra en la Figura 1.

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial T}{\partial y} \right) + Q = \rho c \frac{\partial T}{\partial t} \quad (3)$$

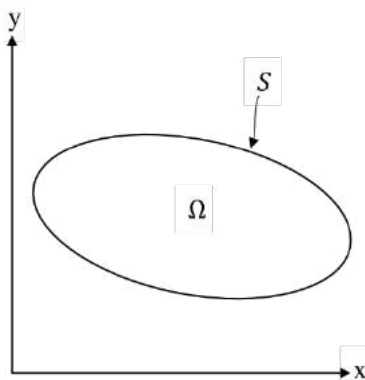


Figura 1. Dominio de solución.

Las condiciones en la frontera  $S$  en el dominio de interés están dadas por:

$$T = T_s \text{ en } S_b \quad (4)$$

$$-k_x \frac{\partial T}{\partial x} l_x - k_y \frac{\partial T}{\partial y} l_y = q \text{ en } S_q \quad (5)$$

$$-k_x \frac{\partial T}{\partial x} l_x - k_y \frac{\partial T}{\partial y} l_y = \alpha(T - T_a) \text{ en } S_h \quad (6)$$

donde  $S_b$  es la parte límite, en la cual, la temperatura ( $T$ ) es prescrita;  $S_q$  es la parte límite, en la cual, el flujo de calor ( $q$ ) es prescrito;  $S_h$  es la parte límite, en la cual, el coeficiente de transferencia de calor,  $\alpha$ , es prescrito;  $k_x$  y  $k_y$  son las especificaciones de conductividad térmica, y donde  $x$  y  $y$  son las direcciones principales del tensor de conductividad térmica;  $l_x$  y  $l_y$  representan las direcciones de los cosenos;  $T_s$  es la temperatura superficial dada en el límite y  $T_a$  es la temperatura atmosférica.

La condición inicial de temperatura está dada por:

$$T(x, y, 0) = T_0(x, y) \quad (7)$$

donde  $T(x, y, t)$  es la temperatura en el dominio y  $T_0$  es la temperatura inicial ( $t = 0$ ).

### B. Método de los Elementos Finitos (MEF)

Para determinar una solución aproximada del modelo de conducción de calor se puede utilizar el MEF mediante el uso del método de residuos ponderados de Galerkin [12]. Implementando este método en (3) se produce un residual,  $R$ , dado por:

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial T}{\partial y} \right) + Q - \rho c \frac{\partial T}{\partial t} = R \quad (8)$$

El residual obtenido es multiplicado por una función de ponderación, la cual se considera igual a una función de forma,  $N$ , por lo que se tiene:

$$\int_{\Omega} N \left[ \frac{\partial}{\partial x} \left( k_x \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial T}{\partial y} \right) + Q - \rho c \frac{\partial T}{\partial t} \right] d\Omega = 0 \quad (9)$$

Para la aproximación de temperaturas dentro de cada elemento se tiene:

$$T(x, y, t) = \sum_{i=1}^n N_i(x, y, t) T_i \quad (10)$$

donde  $n$  es el número de nodos o vértices en el elemento y  $T_i$  es la temperatura dependiente del tiempo en los nodos del elemento respectivo.

Implementando la integración por partes de los tres términos en (4)-(6) y (10) en la representación de Galerkin

de (3), se obtiene:

$$\begin{aligned}
 & - \int_{\Omega} \left[ \frac{\partial N_i}{\partial x} \left( k_x \frac{\partial N_j}{\partial x} T_j \right) + \frac{\partial N_i}{\partial y} \left( k_y \frac{\partial N_j}{\partial y} T_j \right) \right] d\Omega \\
 & + \int_{\Omega} N_i Q - N_i \left( \rho c \frac{\partial N_j}{\partial t} T_j \right) d\Omega \\
 & - \int_{S_q} N_i q dS - \int_{S_h} N_i h (T - T_a) dS = 0 \quad (11)
 \end{aligned}$$

En este artículo, las funciones de forma empleadas constan de elementos triangulares paramétricos de segundo orden de 6 nodos (Figura 2) [13]:

$$N_1 = (1 - 2u - 2v)(1 - u - v) \quad (12)$$

$$N_2 = u(2u - 1) \quad (13)$$

$$N_3 = v(2v - 1) \quad (14)$$

$$N_4 = 4u(1 - u - v) \quad (15)$$

$$N_5 = 4uv \quad (16)$$

$$N_6 = 4v(1 - u - v) \quad (17)$$

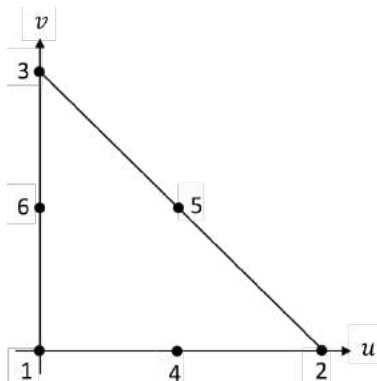


Figura 2. Elemento triangular de 6 nodos con coordenadas paramétricas.

### C. CUDA

En el modelado numérico de fenómenos transitorios, la solución del sistema global de ecuaciones (solver), generada por el MEF, es la sección de código que consume el mayor tiempo de ejecución de la simulación de los modelos. Para disminuir el tiempo de resolución requerido se utiliza la tarjeta gráfica GPU, NVIDIA GTX 1060, y el lenguaje de programación paralelo CUDA, para ejecutar librerías que realizan diversas operaciones matemáticas, de forma más rápida y eficiente [14]-[15]. En este artículo se implementó el solver ICCG (Incomplete Cholesky Gradient

Conjugate) [16] utilizando las librerías de CUDA BLAS y CUDA SPARSE [17]-[18] mediante el lenguaje de programación C++.

### III. SIMULACIÓN DE MODELOS TRANSITORIO DE CONDUCCIÓN DE CALOR

A continuación se presenta la descripción y simulación de dos modelos bajo estudio tomado de [19], mediante la aplicación de un software de elementos finitos en 2D, desarrollado en el lenguaje de programación C++ y ejecutado en una laptop Alienware M15, procesador Intel Core i7 Octava Generación, 16 Gb de memoria RAM. Para la elaboración de las mallas de los modelos se utilizó el software Gmsh [20].

#### A. Simulación transitoria del modelo demostrativo A

El modelo A consiste en determinar la conducción de calor de un material, el cual, tiene una conductividad térmica de  $1.25 \text{ Btu}/(\text{hr m}^\circ\text{F})$ , temperatura inicial de  $30^\circ\text{F}$ , tiempo final de 1.2 hr y condiciones de frontera Dirichlet nulas, las cuales se muestran en la Figura 4.

La malla del modelo A se muestra en la Figura 5. El archivo de texto generado de la malla es utilizado por el software de elementos finitos para la lectura de datos geométricos e inserción de parámetros de materiales, durante la etapa de pre-procesamiento.

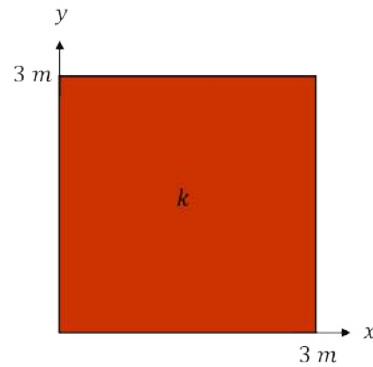


Figura 4. Dimensiones geométricas del modelo A.

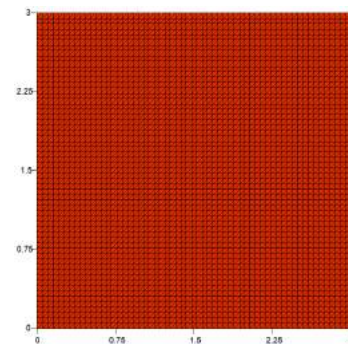


Figura 5. Mallado del modelo A, elaborado en software Gmsh.

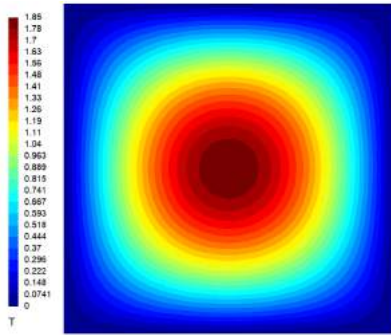


Figura 6. Distribución de temperatura en el modelo A (simulación).

La distribución de temperatura en el modelo A, determinada por la simulación de elementos finitos, se muestra en la Figura 6.

Para validar los resultados generados en la simulación del modelo, se realiza una comparación del mismo con su respectiva solución analítica, obtenida mediante el uso del software Octave [21].

La solución analítica del modelo A, tomado de [19], está dada por (18) y (19) y su visualización correspondiente se muestra en la Figura 7.

$$T(x, y, t) = \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} A_n \sin \left[ \frac{n\pi x}{L_x} \right] \sin \left[ \frac{j\pi y}{L_y} \right] \exp \left[ - \left( \frac{k_x n^2 \pi^2}{L_x^2} + \frac{k_y j^2 \pi^2}{L_y^2} \right) t \right] \quad (18)$$

donde

$$A_n = \frac{4(30)}{nj\pi^2} [(-1)^n - 1][(-1)^j - 1] \quad (19)$$

El porcentaje de error entre los resultados obtenidos de la simulación y la solución analítica se muestra en la Figura 8.

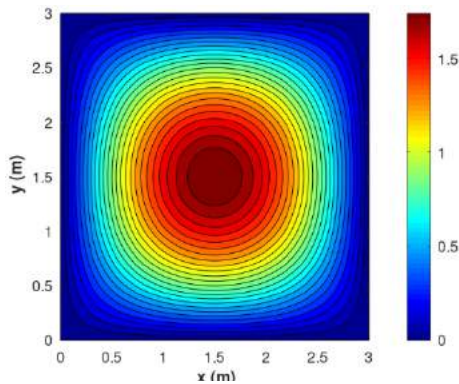


Figura 7. Distribución de temperatura en el modelo A (solución analítica).

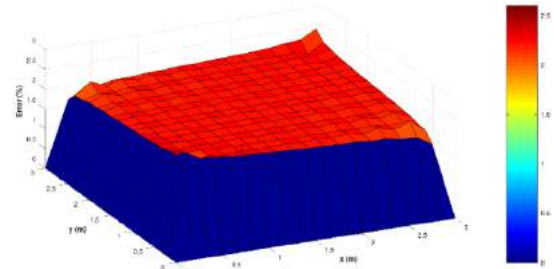


Figura 8. Porcentaje de error entre los resultados obtenidos en simulación y solución analítica de modelo A.

### B. Simulación transitoria del modelo demostrativo B

El modelo B consiste en determinar la conducción de calor de un material, el cual, tiene una conductividad térmica de  $1.0 \text{ Btu}/(\text{hr } m^{\circ}F)$ , temperatura inicial igual a  $0^{\circ}F$ , tiempo final de 0.75 hr, condiciones de frontera Dirichlet de magnitud igual a 1.0 y condiciones de frontera Neumann en los ejes de simetría. Las características geométricas y físicas del modelo B, mencionadas anteriormente, se pueden visualizar en la Figura 9. La malla del modelo B, elaborada en el software Gmsh, se muestra en la Figura 10. Los resultados generados en la simulación del segundo modelo se muestran en la Figura 11. La solución analítica de este modelo, tomado de [19], está dada por (20) y (21) y su visualización se muestra en la Figura 12.

$$T(x, y, t) = 1.0 + \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} C_n \cos \left[ \frac{(2n-1)\pi x}{2L_x} \right] \cos \left[ \frac{(2j-1)\pi y}{2L_y} \right] \exp \left[ - \left( \frac{k_x (2n-1)^2 \pi^2}{4L_x^2} + \frac{k_y (2j-1)^2 \pi^2}{4L_y^2} \right) t \right] \quad (20)$$

donde

$$C_n = \frac{16(-1)(-1)^{n+1}(-1)^{j+1}}{\pi^2(2n-1)(2j-1)} \quad (21)$$

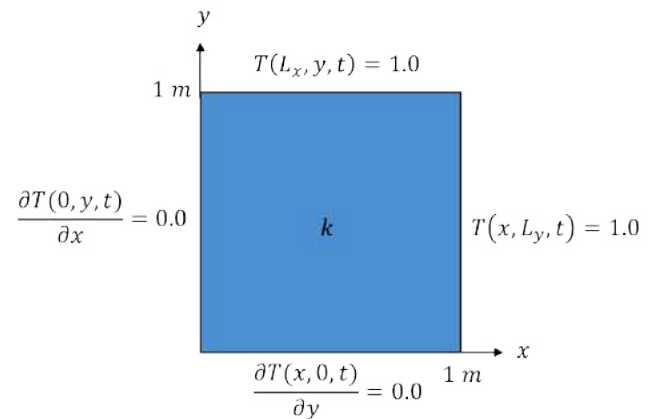


Figura 9. Dimensiones geométricas del modelo demostrativo B.

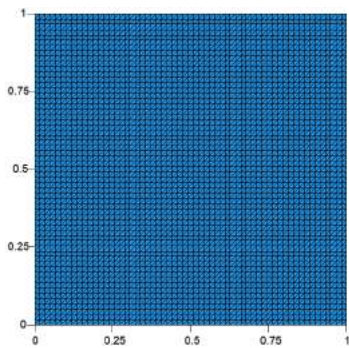


Figura 10. Mallado del modelo B, elaborado en software Gmsh.

El porcentaje de error entre los resultados obtenidos de la simulación y la solución analítica se muestra en la Figura 13.

Para comprobar la aceleración de la resolución del tiempo requerido de solución de los modelos transitorios, antes mencionados, mediante el uso del solver desarrollado en CUDA, se realizó una comparación de desempeño del mismo con el software MKL PARDISO [22], la cual se muestra en la tablas I y II. En estas tablas se puede observar que el tiempo requerido por el solver ICCG es menor al empleado en la simulación de los modelos con el uso del solver comercial MKL PARDISO.

TABLA I  
TIEMPOS DE SOLUCIÓN DEL MODELO A

Solver	Tiempo (s)
PARDISO	940.78
ICCG	458.84

TABLA II  
TIEMPOS DE SOLUCIÓN DEL MODELO B

Solver	Tiempo (s)
PARDISO	1009.72
ICCG	569.631

IV. CONCLUSIÓN

En este artículo se simularon dos fenómenos transitorios de transferencia de calor mediante un software de elementos finitos en 2D, desarrollado por los autores. Se puede visualizar que el porcentaje de error entre los resultados obtenidos en las simulaciones realizadas y sus respectivas soluciones analíticas es menor al 3%, indicando con ello, un grado elevado de confiabilidad. A su vez, se realizó una comparación de tiempos de ejecución de las simulaciones con un solver comercial. Al observar los tiempos de solución requeridos por los solver's, se demuestra que el solver elaborado en CUDA requiere un menor tiempo en comparación con el solver MKL PARDISO.

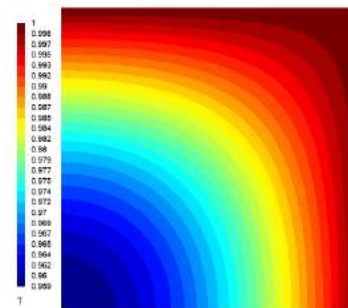


Figura 11. Distribución de temperatura en modelo B (simulación).

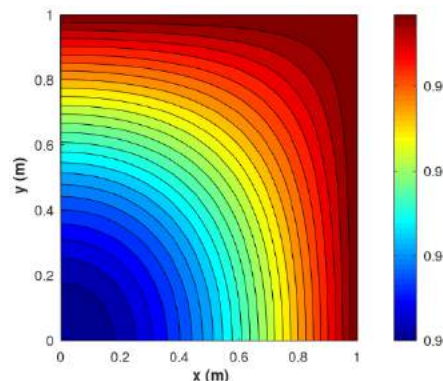


Figura 12. Distribución de temperatura en modelo B (solución analítica).

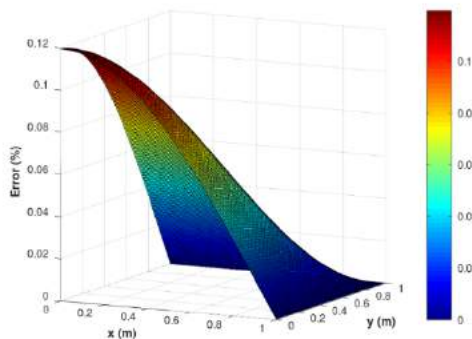


Figura 13. Porcentaje de error entre los resultados obtenidos en simulación y solución analítica de modelo B.

V. REFERENCIAS

- [1] Naskar, A. K. y Sarkar, D. (2014). "Numerical analysis of three dimensional steady state heat conduction in the rotor of an induction motor by finite element method," *Proceedings of the 2014 International Conference on Control, instrumentation, Energy and Communication (CIEC)*, vol. 45(4), pp. 686–690.
- [2] chrittwiesser, M., Bíró, O., Farnleitner, E. y Kastner, G. (2015). "Analysis of temperature distribution in the stator of large synchronous machines considering heat conduction and heat convection," *IEEE Transactions on Magnetics*, vol. 51(3), pp. 20–23.
- [3] Staton, D. A. y Cavagnino, A. (2008). "Convection heat transfer and flow calculations suitable for electric machines thermal models," *IEEE Transactions on Industrial Electronics*, vol. 55(10), pp. 3509–3516.
- [4] Yang, Y., Bilgin, B., Kasprzak, M., Nalakath, S., Sadek, H., Preindl, M., Cotton, J., Schofield, N. y Emadi, A. (2017). "Thermal

- management of electric machines," *IET Electrical Systems in Transportation*, vol. 7(2), pp. 104-116.
- [5] Bastos, J. P. A. y Sadowski, N. (2003). *Electromagnetic Modeling by Finite Element Methods*. 1st ed., CRC Press.
- [6] Jin, J. (2014). *The Finite Element Method in Electromagnetics*. 3rd ed., Wiley-IEEE Press.
- [7] Bianchi, N. (2005). *Electrical Machine Analysis Using Finite Elements*. Boca Raton, USA: CRC Taylor & Francis.
- [8] Capozzoli, A., Kilic, O., Curcio, C., Liseno, A. y Federico, N. (2018). "The success of GPU computing in applied electromagnetics," *Appl. Comput. Electromagn. Soc. J.*, vol. 33(2), pp. 148-151.
- [9] Altinkaynak, A. (2017). "An efficient sparse matrix-vector multiplication on CUDA-enabled graphic processing units for finite element method simulations," *Int. J. Numer. Methods Eng.*, vol. 110(1), pp. 57-78.
- [10] Kindratenko, V. (2014). *Numerical Computations with GPUs*. Springer.
- [11] Couturier, R. (2014). *Designing Scientific Applications on GPUs*. CRC Press.
- [12] Nithiarasu, P., Lewis, R. W. y Seetharamu, K. N. (2016). *Fundamentals of The Finite Element Method for Heat and Mass Transfer*. 2nd ed., Wiley.
- [13] Özgün, Ö. y Kuzuoglu, M. (2019). *MATLAB-based Finite Element Programming in Electromagnetic Modeling*. 1st ed., CRC Press.
- [14] Cheng, J., Grossman, M. y McKercher, T. (2014). *Professional CUDA C Programming*. Wrox Press.
- [15] Cook, S. (2013). *CUDA Programming: a Developer's Guide to Parallel Computing with GPUs*. Morgan Kaufmann Publishers.
- [16] Golub, G. H. y Van Loan, C. F. (1996). *Matrix Computations*. 3rd ed., The Johns Hopkins University Press.
- [17] NVIDIA (2019). *cuBLAS Library User Guide*.
- [18] NVIDIA (2019). *cuSPARSE Library User Guide*.
- [19] John, C., Bruch, JR. y Zyzoloski, G. (1974). "Transient two-dimensional heat conduction problems solved by the finite element method," *International Journal for Numerical Methods in Engineering*, vol. 8, pp. 481-494.
- [20] Geuzaine, C. y Remacle, J.-F. (2009). "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79(11), pp. 1309-1331.
- [21] J. W. E., D. B., S. H. y Wehbring, R. (2020). *GNU Octave version 6.1.0 manual: a high-level interactive language for numerical computations*. [en línea]. Disponible en: <https://www.gnu.org/software/octave/doc/v6.1.0/>.
- [22] Intel (2017). *Intel Math Kernel Library Developer Reference*.

## VI. BIOGRAFÍA



con paralelismo en lenguaje de programación C++.

**Beltrán Pérez José Ángel** nació en Guaymas, Sonora, México en 1996. Obtuvo el título de Ingeniero Eléctrico en 2019 por parte de la Universidad Autónoma de Baja California en la ciudad de Mexicali, B.C., México. Él actualmente cursa la maestría en Ciencias de la Ingeniería en la misma institución. Sus áreas de interés son el diseño de instalaciones eléctricas en baja tensión, el modelado numérico de máquinas eléctricas y el desarrollo de software



Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Autónoma de Baja California, Mexicali, B.C., México. Sus áreas principales de interés son la aplicación de técnicas numéricas computacionales en el modelado y análisis de máquinas eléctricas, especialmente transformadores. El Dr. Díaz cuenta con el perfil deseable PRODEP.



mencionada Institución. Él actualmente labora como Profesor de Tiempo Completo en la carrera de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Autónoma de Baja California. Sus áreas de interés son el análisis de circuitos eléctricos, instrumentación y control, y la adquisición y procesamiento de datos. El Dr. Amaro cuenta con el perfil deseable PRODEP.



B.C., México. Él actualmente labora como Profesor de Tiempo Completo en la carrera de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad Autónoma de Baja California, Mexicali, B. C., México. Sus áreas principales de interés son los procesos industriales aplicados a la ingeniería eléctrica y el desarrollo en los estudiantes de competencias profesionales y académicas. El Dr. Navarro cuenta con el perfil deseable PRODEP.



de la carrera de Ingeniería Eléctrica en la Facultad de Ingeniería de la Universidad Autónoma de Baja California. Cuenta con más de 10 años de experiencia industrial en las áreas de energías renovables, sistemas de generación eólica e instalaciones eléctricas. Sus áreas de interés son las energías renovables, eólica y Sistemas Eléctricos de Potencia. El Dr. Rosales es miembro al Sistema Nacional de Investigadores nivel 1, así como también es miembro del Consejo Técnico del EGEL en Ingeniería Eléctrica. El Dr. Rosales cuenta también con el perfil deseable PRODEP.