



# Análisis computacional de la convergencia del método de Newton-Raphson para el estudio de circuitos con elementos memristivos

## Computational Analysis of the Newton-Raphson Method's Convergence to Study Circuits with Memristive Elements

J. Salamanca<sup>a\*</sup>, J. J. Leal<sup>b</sup>, D. J. Rodríguez<sup>c</sup>.

<sup>a</sup>Universidad Distrital Francisco José de Caldas, Facultad de Ciencias y Educación, Grupo de Física e Informática "FISINFOR".

<sup>b</sup>Universidad Cooperativa de Colombia.

<sup>c</sup>Universidad Distrital Francisco José de Caldas, Facultad de Ingeniería.

Recibido abril 13 de 2014; aceptado diciembre 19 de 2014.

### Resumen

Con el advenimiento del memristor como cuarto elemento electrónico pasivo, se abre un área de investigación que busca consolidar el estudio acerca de dispositivos con memoria y la construcción de los mismos; de hecho, el memristor -resistencia con memoria- ha permitido que la concepción del sistema memristivo sea extendida a otros elementos como memcapacitores y meminductores. Como aporte al estudio de circuitos que incluyen elementos memristivos, en este artículo se analiza la solución computacional de la ecuación diferencial ordinaria no-lineal proveniente del circuito MRLC, Memristor-Resistor-Inductor-Capacitor, por el método de Newton-Raphson, utilizando la discretización proveniente del método del trapecio. Los resultados de la implementación computacional son contrastados con el circuito RLC, asumiendo este último como su límite asintótico cuando  $M \rightarrow 0$ . El resultado de esta implementación servirá como herramienta para validar computacionalmente un estudio futuro de parámetros de circuitos que incluyan elementos memristivos.

**Palabras clave:** memristor, sistemas memristivos, método de Newton-Raphson, ecuaciones diferenciales no-lineales.

### Abstract

The advent of the memristor as a fourth passive electronic element, has opened a research area that seeks to strengthen the study of memory devices and its construction. In fact, the memristor -resistance memory- has allowed to extend the conception of memristive system to other elements such as memcapacitors and meminductors. As a contribution to the study of circuits including memristive elements, this paper analyzes the computational solution of a nonlinear ordinary differential equation from a MRLC circuit, Memristor-Resistor-Inductor-Capacitor, by using the Newton-Raphson method, with a trapezoidal discretization method, to contrast it with the RLC circuit as its asymptotic limit when  $M \rightarrow 0$ . The result of this implementation will serve as a tool to validate, computationally, a further parametric study of circuits that include memristive elements.

**Keywords:** memristor, memristive systems Newton-Raphson, non-linear differential equations.

\* jasalamanca@udistrital.edu.co

## 1. Introducción

La literatura es amplia en el sentido conceptual del memristor dado que se toma como un dispositivo que recuerda el último punto de operación, y que consecuentemente, se le llama “memristor” porque las unidades que lo definen corresponden a unidades de resistencia; es decir, un resistor (resistencia) con memoria. A partir de este concepto, existen diversas variantes en cuanto a la manera de concebir memoria a través de distintos dispositivos como memcapacitores o meminductores [1]. La amplitud de la idea de memristor se extiende a múltiples áreas de la investigación científica y tecnológica como en el desarrollo de dispositivos de almacenamiento de memoria, la implementación de otras lógicas aparte de la binaria en el procesamiento de señales, desarrollo de redes neuronales más complejas, sistemas biológicos (donde se concibe la sangre como un fluido memristivo [2]). Estas aplicaciones involucran análisis de circuitos que acogen elementos memristivos, cuyas ecuaciones pueden llegar a presentar no linealidades debido a la forma funcional del dispositivo memristivo.

El presente trabajo muestra una metodología para validar el análisis de circuitos que incluyen elementos memristivos, implementando computacionalmente el método de Newton-Raphson (utilizando el método del trapecio como discretización) a través del estudio de un circuito MRLC Memristor-Resistor-Inductor-Capacitor, con el objeto de obtener computacionalmente la solución de las ecuaciones que rigen su comportamiento.

## 2. El memristor

En 1971, el profesor Leon Ong Chua propone que a partir de las relaciones de simetría entre carga, voltaje, corriente y flujo de campo magnético, correspondientes a resistencia-capacitancia-inductancia, debe existir un cuarto elemento en los circuitos electrónicos pasivos que relacione directamente la carga eléctrica con el flujo magnético [3] (ver Fig. 1).

Siguiendo el sentido de la simetría en las relaciones ilustradas en la Fig. 1, la relación entre carga y flujo debe ser:

$$d\varphi/dt = M * dq/dt \quad (1)$$

ya que la parte derecha de la diagonal  $q-i$  relaciona primeras derivadas. Aquí, la constante de proporcionalidad  $M$  se le denomina “memristor”.

Se puede obtener una expresión para el memristor eliminando la derivada temporal de la Eq. 1 y haciendo una relación funcional para  $\varphi$  definida por  $f(q)$ . Así, la función de memristencia  $M(q)$  puede ser escrita como:

$$M(q) = \frac{df(q)}{dq} \quad (2)$$

## 3. Circuito MRLC

Un circuito MRLC (con todos sus componentes conectados en serie) puede ser modelado a partir del la Eq. 3 que gobierna un circuito RLC

$$\frac{d^2q}{dt^2} + \frac{R}{L} \frac{dq}{dt} + \frac{q}{LC} = \frac{V_0}{L} \sin \omega t, \quad (3)$$

donde la ecuación diferencial tiene como parámetros a  $R$ ,  $L$ ,  $C$ ,  $\omega$ , la frecuencia de la fuente, y  $V_0$ , el máximo voltaje suministrado por la fuente. El aporte del elemento memristivo es agregado en serie, donde el término resistivo  $R$  ahora cambia a  $R + M(q)$  y la ecuación que describe el comportamiento del circuito memristivo queda expresada como:

$$\ddot{x} + \alpha(x)\dot{x} + \beta x = f_0 \sin \omega t \quad (4)$$

con  $x$ , la carga eléctrica;  $f_0 = V_0/L$ ;  $\alpha(x) = [R + M(x)]/L$  y  $\beta = 1/LC$ . Esta forma generalizada en términos de  $\alpha, \beta, \gamma$  es de gran utilidad para la exposición del método y la implementación computacional.

## 4. Descripción del método de Newton-Raphson

El método de Newton-Raphson es uno de los métodos numéricos más poderosos para encontrar raíces de una función real y es utilizado ampliamente en la solución de ecuaciones diferenciales. El método permite escribir una ecuación diferencial de orden superior en un sistema de ecuaciones algebraicas no lineales que reducidas, como se muestra en seguida, consienten la solución numérica de la ecuación diferencial.

El método define un sistema de ecuaciones homogéneas de la forma:

Este trabajo es publicado por la Sociedad Colombiana de Física y distribuido en open access según los términos de la licencia Creative Commons Attribution.

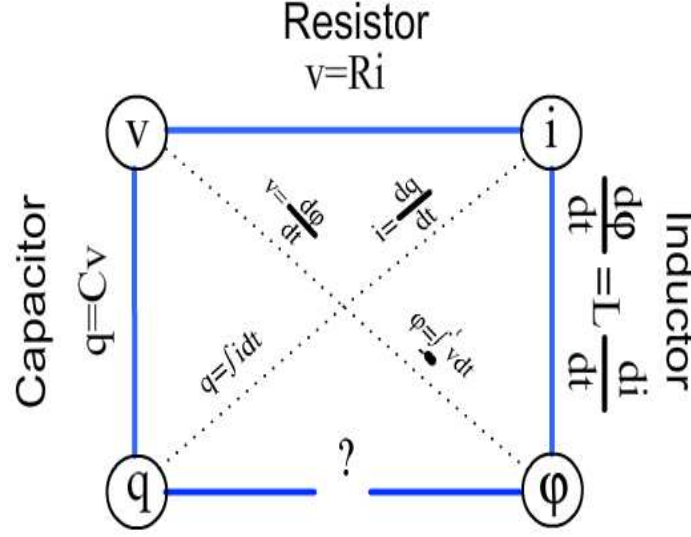


Figura 1. Relaciones de simetría entre los diferentes elementos de los circuitos pasivos.

$$\begin{aligned}
 g_1(x_1, x_2, \dots, x_n; t) &= 0 \\
 g_2(x_1, x_2, \dots, x_n; t) &= 0 \\
 &\vdots \\
 g_n(x_1, x_2, \dots, x_n; t) &= 0,
 \end{aligned} \tag{5}$$

a las que se debe reducir la ecuación diferencial. En este contexto,  $x_1, x_2, \dots, x_n$  son las componentes de un vector en  $\mathbb{R}^n$  (que, a fin de cuentas, llegan a ser derivadas de un orden inferior a la ecuación diferencial) y  $t$ , el parámetro de tiempo que indica dependencia temporal de las componentes  $x_i(t)$ .

Inicialmente, la Eq. 4 (ecuación diferencial de segundo orden) se transforma a un sistema de dos ecuaciones

$$\begin{aligned}
 \dot{x}_1 &= f_1(x_1, x_2, t) \\
 \dot{x}_2 &= f_2(x_1, x_2, t)
 \end{aligned} \tag{6}$$

haciendo  $x_1 = x$  y  $x_2 = \dot{x} = p$ . Así, la Eq.4 queda reducida a un sistema de ecuaciones diferenciales de primer orden:

$$\dot{x} = p \tag{7}$$

$$\dot{p} + \alpha(x)p + \beta x = f_0 \sin(\alpha t). \tag{8}$$

Resolviendo para  $\dot{x}$  y  $\dot{p}$  se encuentra que

$$\dot{x} = p = f_1(x, p, t) \tag{9}$$

$$\dot{p} = f_0 \sin(\alpha t) - \alpha(x)p - \beta x = f_2(x, p, t), \tag{10}$$

lo cual es congruente con la forma deseada de las Eqs. 4.

Como se trata de un problema de valor inicial fijo, el punto de partida  $(x(t_0), p(t_0))$  estará definido por una

referencia de tiempo  $t_0$ . El sistema entonces presenta una evolución temporal donde sus coordenadas  $x$  y  $p$  dependen de la discretización del tiempo. El siguiente punto en el espacio de fase  $x$ - $p$  corresponde a evaluar las coordenadas en  $t_1 = t_0 + h$ , donde  $h$  (conocido como el paso de la iteración) se hace tan pequeño como sea posible para que se minimice la diferencia en tiempos y la solución se aproxime, tanto como sea posible, a la solución real de la ecuación diferencial. De este modo, la discretización para un tiempo  $t_j$  queda definida como  $t_j = t_0 + j * h$ .

Asociando a  $x(t_0) \rightarrow x_{1,0}$ ,  $p(t_0) \rightarrow x_{2,0}$ ,  $x(t_1) \rightarrow x_{1,1}$  y a  $p(t_1) \rightarrow x_{2,1}$ , la relación entre el punto evaluado en  $t_j$  y  $t_{j+1}$  estaría definido por las ecuaciones de recurrencia

$$\begin{aligned}
 x_{1,j+1} &\approx x_{1,j} + \Delta_1(x_{1,j+1}, x_{2,j+1}, x_{1,j}, x_{2,j}; h, t_j)
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 x_{2,j+1} &\approx x_{2,j} + \Delta_2(x_{1,j+1}, x_{2,j+1}, x_{1,j}, x_{2,j}; h, t_j),
 \end{aligned} \tag{12}$$

donde la aproximación  $\Delta_i$  depende de la técnica que se escoja (Euler, Verlet, Runge-Kutta, etc). La literatura establece que la región de estabilidad absoluta “R” es un factor crítico a la hora de escoger el método de aproximación con la mejor exactitud. Según la rigidez (“stiffness”) del sistema, i.e., algunos componentes de la ecuación diferencial decaen más rápido que otros, se requiere utilizar métodos numéricos que ofrezcan un comportamiento estable [7]. Para métodos multi-paso, el método del trapecio es el único que cumple con el criterio de estabilidad [4]. Se dice entonces que el método es “A-Stable”.

Implementando el método del trapecio, las ecuaciones de recurrencia quedan expresadas así:

$$x_{1,j+1} \approx x_{1,j} + \frac{h}{2} [f_1(x_{1,j+1}, x_{2,j+1}; t_{j+1}) + f_1(x_{1,j}, x_{2,j}; t_j)] \quad (13)$$

$$x_{2,j+1} \approx x_{2,j} + \frac{h}{2} [f_2(x_{1,j+1}, x_{2,j+1}; t_{j+1}) + f_2(x_{1,j}, x_{2,j}; t_j)] \quad (14)$$

con lo que se reconoce la forma homogénea acorde a las Eqs. 5, simplemente restando la parte derecha de la izquierda en las Eqs. 13 y 14 respectivamente; esto es:

$$g_1(x_{1,j+1}, x_{2,j+1}; t_{j+1}) = x_{1,j+1} - x_{1,j} - \Delta_1 \approx 0 \quad (15)$$

$$g_2(x_{1,j+1}, x_{2,j+1}; t_{j+1}) = x_{2,j+1} - x_{2,j} - \Delta_2 \approx 0. \quad (16)$$

Con la finalidad de hacer la implementación mucho más sencilla para construir la solución aproximada de la ecuación diferencial se utiliza una generalización del método. Esta consiste en ilustrar las ecuaciones de recurrencia (Eqs. 11 y 12) para un funcional  $\vec{X}_j$  definido en un espacio vectorial sobre un campo de números reales (en general un espacio de Banach) como:

$$\vec{X}_{j+1} = \vec{X}_j - J^{-1}(\vec{X}_j; t_j) * \vec{G}(\vec{X}_j; t_j) \quad (17)$$

con

$$\vec{X}_j = \begin{bmatrix} x_{1,j} \\ x_{2,j} \end{bmatrix}, \vec{G}_j = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, J = \begin{bmatrix} \frac{\partial g_1}{\partial x_{1,j+1}} & \frac{\partial g_1}{\partial x_{2,j+1}} \\ \frac{\partial g_2}{\partial x_{1,j+1}} & \frac{\partial g_2}{\partial x_{2,j+1}} \end{bmatrix} \quad (18)$$

donde  $J$  es el jacobiano de la transformación de coordenadas  $(x_1, x_2)$  a  $(g_1, g_2)$  (o derivada de Frechet).

## 5. Implementación computacional del método aplicado al circuito MRLC

El abordaje computacional que trata este artículo tiene asiento en la implementación computacional de la prescripción que muestra la Eq. 17. Se trata de emular el cálculo a través de operaciones matriciales que dejan expresiones de cálculo simples, facilitando así el mantenimiento y depuración de los códigos. Para tales fines, se utilizó programación en C++ y una infraestructura de software llamada ROOT [5,6] (Rapid Oriented Object Technology) que define una vasta gama de clases que permiten hacer análisis de datos y consienten operaciones matriciales y gráficas.

La Fig. 2 ilustra el diagrama de flujo del algoritmo diseñado para la implementación computacional.

Primeramente se definen las condiciones iniciales para  $x, p, t$  y  $\vec{X}(t_j)$  y luego los valores que parametrizan la ecuación diferencial (Eq. 4). Seguidamente, se establecen las funciones  $g_1$  y  $g_2$  para el caso:

$$g_1 = x_{1,j+1} - x_{1,j} - \frac{h}{2} [x_{2,j+1} + x_{2,j}] \quad (19)$$

$$g_2 = x_{1,j+1} - x_{1,j} - \frac{h}{2} [f_0 \sin(\omega t_{j+1}) - \alpha(x_{1,j+1})x_{2,j+1} - \beta x_{1,j+1} + f_0 \sin(\omega t_j) - \alpha(x_{1,j})x_{2,j} - \beta(x_{1,j})], \quad (20)$$

donde  $\alpha(x)$  se toma como un valor constante del circuito RLC, para el caso MRLC, se implementa como una función que depende de la función de memristencia  $M(x)$  (esto se elaborara con detalle más adelante).

Después de lo anterior, se declaran los valores del paso  $h$  definiendo un intervalo de tiempo  $t_f = t_0 + n * h$  con el ánimo de calcular el jacobiano de la transformación de coordenadas:

$$J(\vec{X}_0) = \begin{bmatrix} 1 & -\frac{h}{2} \\ \frac{h}{2}(x_{2,0} \frac{d\alpha(x_{1,0})}{dx_{1,0}} + \beta) & 1 + \frac{h}{2}\alpha(x_{1,0}) \end{bmatrix} \quad (21)$$

donde el valor entero positivo de  $n$  corresponde al número de iteraciones o puntos a calcular en el espacio de fase  $x-p$ .

Finalmente, se obtiene una expresión de la matriz inversa del jacobiano ( $J^{-1}$ ), que es evaluada en  $t_j$  dentro de un ciclo ("loop") de  $n$  iteraciones, para así obtener el valor del vector  $\vec{X}$  en  $t_{j+1}$  y permitir alojar  $n$  valores de una tetra ordenada  $x, p$  y  $t$ . De esta forma se pueden obtener gráficas de  $x(t), p(t)$  y del espacio de fase  $x-p$ .

Es importante mencionar que el número de iteraciones se debe escoger dependiendo del orden de la diferencia  $\lambda$  entre dos valores calculados de una variable i.e.,  $x$  para un valor fijo  $t$ , esto es:

$$\lambda = |x(t)_{n_{i+1}} - x(t)_{n_i}|, \quad (22)$$

donde  $x(t)_{n_i}$  es el valor de la variable  $x$  a un valor fijo  $t$  calculado a partir de un número  $n_i$  de iteraciones.

Uno de los criterios para escoger el valor  $n$  es el criterio de convergencia de Cauchy, que sugiere encontrar una región de convergencia

$$\lambda = |x(t)_{n_{i+1}} - x(t)_{n_i}| < \epsilon \quad (23)$$

tal que  $\epsilon \rightarrow 0$ . Para valores pequeños de  $n$ , se espera que  $\lambda$  sea grande comparado con valores que se obtienen para cuando  $n$  se hace grande ( $\lambda \ll 0$ ). En este último régimen también se espera que los valores de  $\lambda$

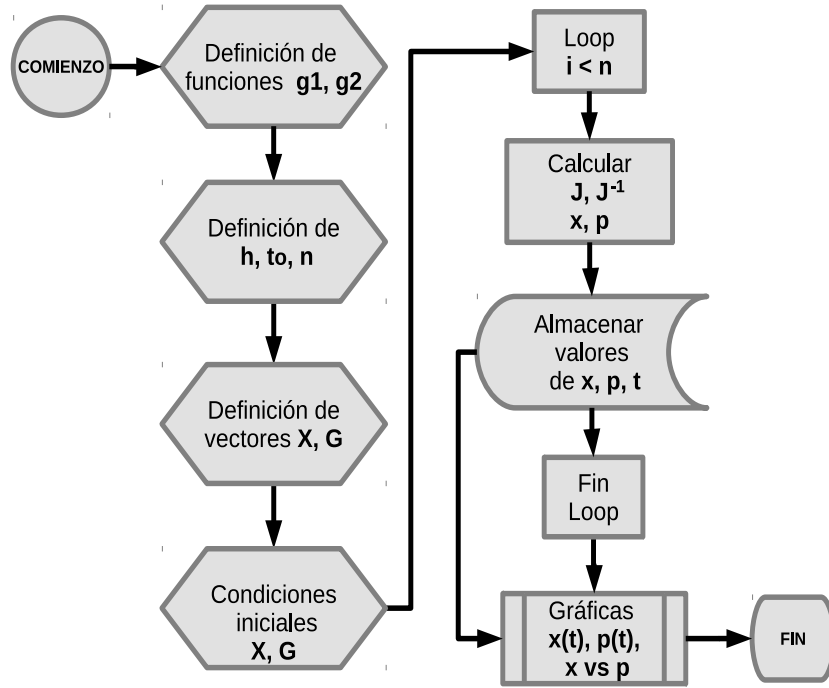


Figura 2. Diagrama de flujo del algoritmo utilizado para la implementación computacional.

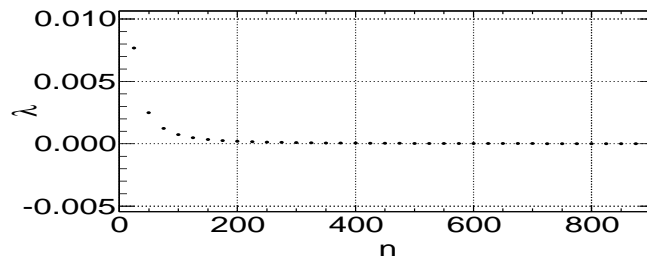


Figura 3. Criterio de convergencia de Cauchy para diferentes valores de iteración  $n_i$ .

cambien muy despacio de manera que sea posible encontrar un radio de convergencia  $\epsilon$ . Este comportamiento es ilustrado en la Fig. 3.

Como se dijo anteriormente, el criterio de selección de  $n$  depende del orden de magnitud de  $\lambda$  que se desee, amigable tanto para la obtención de resultados como para el gasto computacional. Para tal propósito, se puede realizar una gráfica a escala logarítmica de la Fig. 3 que permita mostrar los ordenes de magnitud de  $\lambda$  en función de los  $n_i$ . En efecto, la Fig. 4 muestra que para obtener un orden de magnitud de  $10^{-4}$  en  $\lambda$ , se deben hacer alrededor de 300 iteraciones en el intervalo escogido, tal que  $h = \Delta t/300$ .

## 6. Prueba de validación y análisis de resultados

La solución general para  $M(q) = 0$  se corresponde a un circuito RLC cuyo comportamiento obedece a la Eq. 3. Teniendo en cuenta que la solución general de dicho oscilador corresponde a una parte transitoria  $x_T$  y otra estacionaria  $x_E$

$$x(t) = x_T(t) + x_E(t), \quad (24)$$

donde

$$x_T(t) = \frac{f_0}{\sqrt{(\omega_0^2 - \omega^2)^2 + \alpha^2 \omega^2}} \quad (25)$$

$$x_E(t) = A \cos w_p t + B \cos w_p t, \quad (26)$$

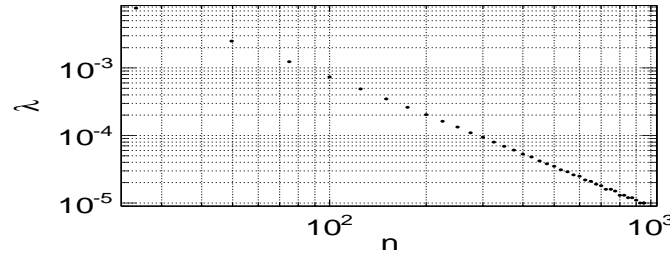


Figura 4. Criterio de convergencia de Cauchy para diferentes valores de iteración  $n_i$  en escala logarítmica.

la prueba de validación del algoritmo se evalúa en el caso

$$\omega_p = \sqrt{\omega_0 - \frac{\alpha^2}{4}} > 0 \quad (27)$$

siendo  $\omega_0$  la frecuencia natural del circuito ( $\beta = \omega_0^2$ ).

Si el amortiguamiento (descrito por  $\alpha$ ) y el valor en amplitud de la fuerza electromotriz externa ( $V_0$ ) son pequeños, debemos recobrar sólo la parte estacionaria: un oscilador armónico (ver Fig. 5).

Igualmente, para el caso amortiguado con  $V_0 = 0$ , el resultado esperado corresponde a la parte transitoria: un oscilador amortiguado (ver Fig. 6).

En concierto, dadas las condiciones de amortiguamiento y forzamiento, la Fig. 7 ilustra claramente la parte transitoria y estacionaria.

### 6.1. Función de memristencia

La función de memristencia utilizada en la implementación computacional corresponde al modelo de Hernández-Noguera [8]:  $M(q) = \kappa q^2$ , el cual ilustra las propiedades físicas de pasividad ( $M(q) > 0$ ) e histéresis en voltaje vs. corriente eléctrica [8], dejando que el valor de  $\alpha(t)$  quede expresado como:  $1/L(R + \kappa x^2)$ , donde  $\kappa$  corresponde a una constante que mantiene las dimensiones de resistencia de la función.

Las Figs. 8, 9 y 10 ilustran las soluciones para tres diferentes regímenes de  $\kappa$ : I ( $R \gg M(q)$ ), II ( $R \sim M(q)$ ) y III ( $R \ll M(q)$ ), donde se muestra el comportamiento esperado: I) MRLC análogo a RLC, II) y III) MRLC muestra un comportamiento diferente al esperado de un circuito RLC.

Los valores utilizados tanto para la prueba de validación como para la implementación del circuito MRLC se ilustran en la Tabla 1.

Tabla 1. Parámetros utilizados en la implementación computacional.

Parámetro	Valor	Circuito
$t_0$	0	RLC, MRLC
$n$	300	RLC, MRLC
$x(t_0)$	1	RLC, MRLC
$p(t_0)$	0	RLC, MRLC
$\alpha = R/L$	0.2	RLC
$\beta = 1/LC$	1	RLC, MRLC
$V_0$	1	RLC, MRLC
$f_0 = V_0/L$	1	RLC, MRLC
$\omega$	1	RLC, MRLC
$R$	0.2	MRLC
$L$	1	MRLC
$\kappa$ (Región)	0.1(I) ,10(II) ,100(III)	MRLC

## 7. Conclusiones

Al hacer una revisión detallada de los métodos computacionales para resolver ecuaciones diferenciales ordinarias se encuentra que en la literatura hay pocas referencias que utilizan el método de Newton-Raphson con la discretización del método del trapecio, la cual presenta una baja sensibilidad al tamaño del paso después de definir una región de convergencia como se ilustra en la Fig. 3. Consecuentemente, se puede escoger un valor del paso para la implementación computacional tal que el orden en el error pueda ser estimado como un orden de magnitud a partir de una escala logarítmica; esto se ilustra en la Fig. 4.

De otro lado, la validación computacional para el circuito MRLC cumple con el circuito RLC, entendido este último como su límite asintótico para  $M(q) \rightarrow 0$ . Allí se recuperan resultados ya conocidos para un circuito RLC (Figs. 5, 6 y 7) y se corroboran con el comportamiento esperado de las comparaciones en términos

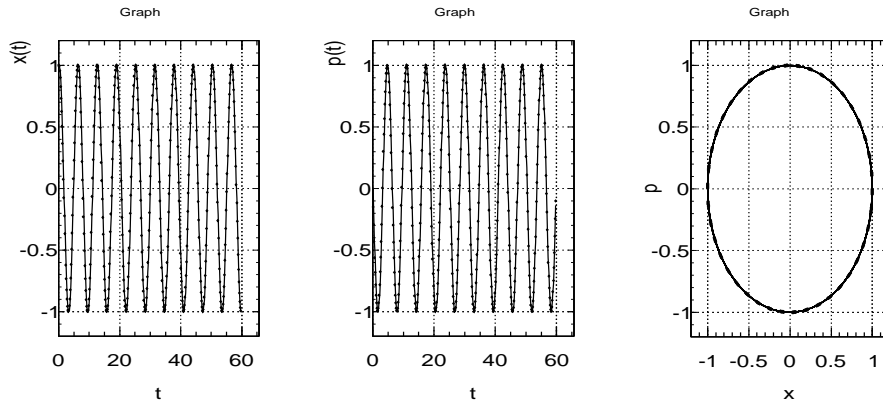


Figura 5. Gráfica comparativa entre datos calculados por el método de Newton-Raphson (puntos) y la solución analítica (línea continua) Eq. 24 con  $\alpha = 0$  y  $V_0 = 0$ .

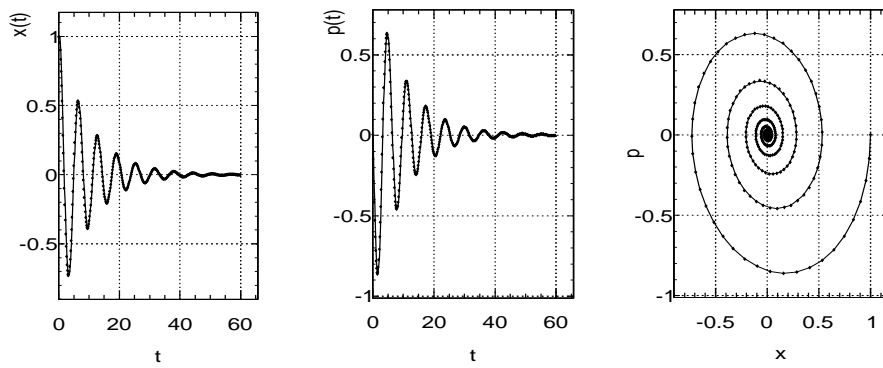


Figura 6. Gráfica comparativa entre datos calculados por el método de Newton-Raphson (puntos) y la solución analítica (línea continua) Eq. 24 con  $\alpha = 0,2$  y  $V_0 = 0$ .

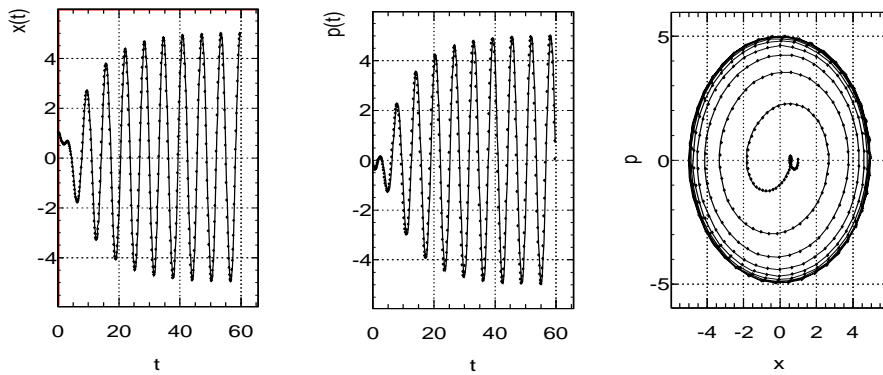


Figura 7. Gráfica comparativa entre datos calculados por el método de Newton-Raphson (puntos) y la solución analítica (línea continua) Eq. 24 con  $\alpha = 0,2$  y  $V_0 \neq 0$ .

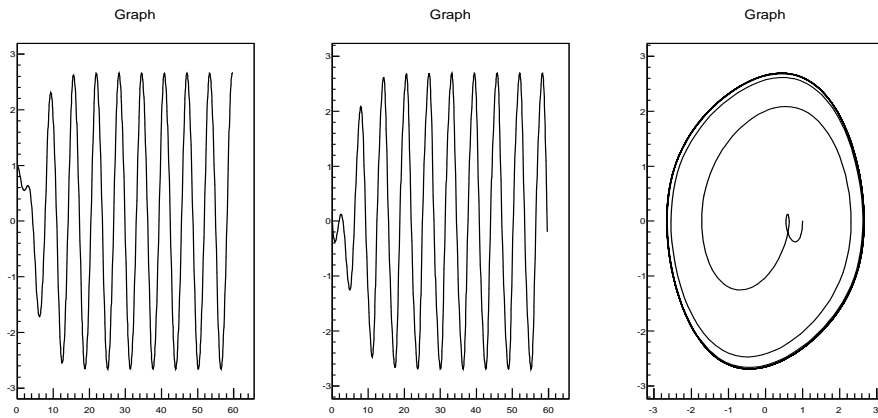


Figura 8. Circuito MRLC, régime I ( $R \gg M(q)$ ).

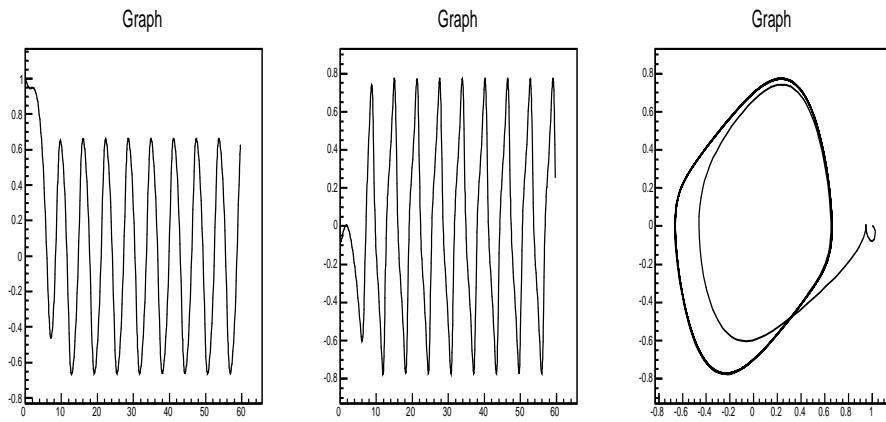


Figura 9. Circuito MRLC, régime II ( $R \sim M(q)$ ).

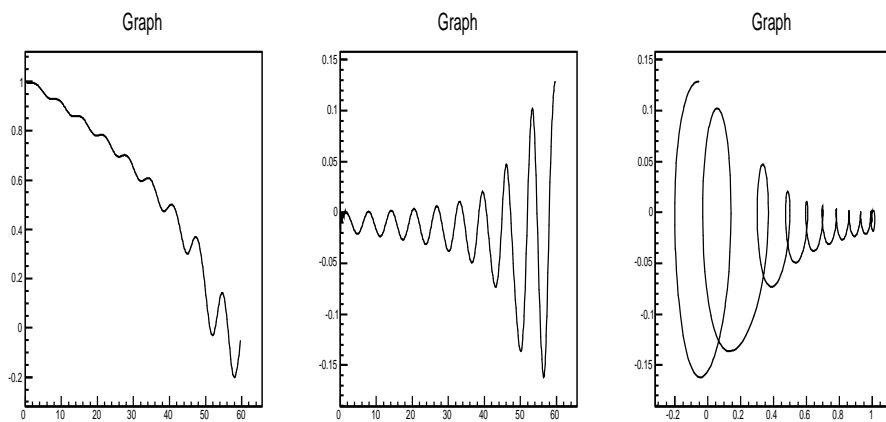


Figura 10. Circuito MRLC, régime III ( $R \ll M(q)$ ).

de orden de magnitud entre  $R$  y  $M(q)$  mostradas en las Figs. 8, 9 y 10 para el circuito MRLC.

Adicionalmente, los resultados de este trabajo pueden ser utilizados para hacer análisis de parámetros para el circuito MRLC y también pueden ser la base para soportar otros análisis para circuitos que incluyan elementos memristivos (memristores, memcapacitores, meminductores) y en diferentes configuraciones (serie, paralelo), sin contar con otros sistemas que impliquen, en particular, no-linealidades en su ecuaciones de movimiento (modelos físico-matemáticos).

## Referencias

- [1] M. Di Ventra, V. Pershin, L. O. Chua, *Proceedings of the IEEE*, **97-10**, 2009, pp. 1717-1724.
- [2] R. Courtland, *Memristors...Made of Blood?*, 2011, p. 1. Disponible en: <http://goo.gl/lqhvpr>
- [3] L. O. Chua, *IEEE Transactions on Circuit Theory*, **18**, 1971, pp. 507-519.
- [4] R. L. Burden, J. D. Faires, *Numerical Analysis*, Belmont, C. A., Thompson Brooks/Cole, Seventh edition, 2005.
- [5] R. Brun, F. Rademakers, *Nucl. Inst. Meth. in Phys Res*, **A-389**, 1997, pp. 81-86.
- [6] The ROOT Team. *The ROOT Users Guide v.5.26*, 2009, p. 1. Disponible en: <http://root.cern.ch/drupal/content/users-guide>
- [7] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley, 1992.
- [8] Hernandez-Noguera, R., *Diseño de aplicaciones interactivas mediante Easy Java Simulation, para el estudio del memristor y elementos de orden superior*, Catalunya, España, 2011. Proyecto final de carrera, Dept. Ing. Elec. Univ. Politécnica de Catalunya. Disponible en: <http://goo.gl/2DPZgZ>