Interval Extensions as a Way to Achieve Reduction in Carbon Emissions and Energy Consumption

1st Josefredo Gadelha da Silva
Centre for Ocean Energy Research
Department of Electronic Engineering
Maynooth University
josefredo.silva.2024@mumail.ie

2nd Thalita Emanuelle De Nazaré
Centre for Ocean Energy Research
Department of Electronic Engineering
Maynooth University
thalita.nazare.2023@mumail.ie

3rd Marcio Junior Lacerda
Department of Electrical Engineering
Federal University of São João del-Rei
lacerda@ufsj.edu.br

4th Erivelton Geraldo Nepomuceno
Centre for Ocean Energy Research
Department of Electronic Engineering
Maynooth University
erivelton.nepomuceno@mu.ie

2023 15th IEEE International Conference on Industry Applications (INDUSCON) | 979-8-3503-1418-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/INDUSCON58041.2023.10374821

***Abstract*—The present work explores the use of interval extensions to reduce the carbon footprint of computer simulation. We present a method for estimating the amount of energy expended based on the number of operations performed and we demonstrate how different numerical representations of the same system can lead to a considerable reduction in the number of operations and, consequently, energy consumption. We have applied our technique on three scenarios for discrete simulation of the well-known Chua's circuit. The results show a saving of approximately 22.50% in the values of emitted CO2 and consumed energy when using two different forms of representation of the same system. A simple rewrite of a differential equation could lead to a decrease in the number of operations, mainly the number of multiplications adopted in the representation of Chua's circuit, resulting in reliable and more efficient simulations.**

I. INTRODUCTION

The computational domain is characterised by a paradigm that is somehow different from the world of real numbers. Most modern computers operate according to the standards of floating-point arithmetic IEEE 754 [1] [2]. In practice, mathematical properties that are considered absolute with respect to real numbers, such as distributivity and commutativity, may not be obeyed in the computational domain [3]. This is because computers are finite systems with finite energy, and the floating-point standard establishes rounding and truncation conditions to be observed during the execution of operations. These conditions can be seen as errors added to a system when simulated in the computational domain. Some works have shown how to control these errors [4] [5].

In contrast, recent works have addressed the issue of green computing and the need to adopt practices that reduce energy consumption in computational environments [6]–[8]. In this sense, even though some mathematical properties may not be synthesised in the computational domain, they can serve as a previous alternative to ensure the reliability of simulation results of a system. Simpler equivalent algebraic forms of a system can be explored in order to have the accuracy of results and also ensure lower simulation complexity. This alternative algebraic representations are known as interval extensions [9].

Among the various types of systems, the simulation of nonlinear systems is usually the most problematic, as it requires highly advanced discretization techniques. The present work explores the use of interval extensions for the representation and simulation of Chua's circuit, a dynamic circuit with nonlinear and chaotic behaviour. We present a method for estimating the amount of energy expended based on the number of operations performed, and we demonstrate how different numerical representations of the same system can lead to a considerable reduction in the number of operations, and consequently, energy consumption. The results show that a simple rewrite of a differential equation (ODE) could lead to a decrease in the number of operations, mainly the number of multiplications adopted in the representation of the Chua's circuit, resulting in reliable and more efficient simulations. These findings contribute to the discussion about green computing, energy-saving, and energy efficiency, while also providing insights into modelling techniques for dynamic systems and computational simulations.

II. BACKGROUND

The Chua's circuit's is a three-component circuit that exhibits complex and chaotic behaviour, non-periodic oscillations and is highly sensitive to the initial conditions. The topology of Chua's circuit is shown in Figure 1.

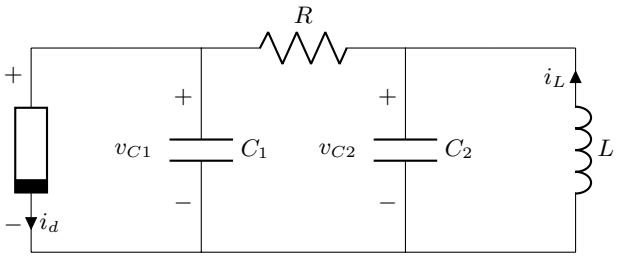

Fig. 1: Chua's circuit topology, adapted from [10]

The dynamics of the circuit can be modelled by the following Equations:

$$\frac{dv_{C1}}{dt} = \frac{1}{C_1}\left[\frac{(v_{C2} - v_{C1})}{R} - i_d(v_{C1})\right], \tag{1}$$

$$\frac{dv_{C2}}{dt} = \frac{1}{C_2}\left[\frac{(v_{C1} - v_{C2})}{R} - i_L\right], \tag{2}$$

$$\frac{di_L}{dt} = -\frac{1}{L}v_{C2}, \tag{3}$$

$$i_d(v_{C1}) = \begin{cases} G_b v_{C1} + B_p(G_b - G_a) & , v_{C1} < -B_p \\ G_a v_{C1} & , |v_{C1}| \leq B_p \\ G_b v_{C1} + B_p(G_a - G_b) & , v_{C1} > B_p, \end{cases} \tag{4}$$

Authorized licensed use limited to: Maynooth University Library. Downloaded on September 15,2025 at 11:32:00 UTC from IEEE Xplore. Restrictions apply.

where v_{C1} and v_{C2} represent the voltages on capacitors C_1 and C_2, respectively; i_L is the current in inductor L and i_d is the current that flows through the Chua's diode and is dependent on voltage v_{C1}. While G_a, G_b, and B_p are related to the slope of the diode curve and are given in terms of the resistances that make up such a component, which can be implemented through operational amplifiers.

A. Karatsuba Method

The Karatsuba algorithm is a fast multiplication algorithm that was first published by Anatolii Karatsuba [11]. The algorithm can perform multiplication of two n-digit numbers in $O(n^{log_2(3)}) \approx O(n^{1.585})$ time complexity, which is faster than the traditional multiplication algorithm's $O(n^2)$ time complexity.

Karatsuba's breakthrough came from observing that the multiplication of two n-digit numbers could be reduced to three multiplications of $\frac{n}{2} - digit$ numbers and some additional operations, instead of four multiplications as in the traditional algorithm. This reduction in the number of multiplications leads to a significant reduction in the overall time complexity of the algorithm.

B. Green Algorithm

Green algorithms is a rapidly growing field that focuses on developing algorithms that are environmentally friendly and energy-efficient. The principal objective of green algorithms is to mitigate the environmental impact of computing and data processing, a matter of growing significance in light of the proliferation of big data and the rapid expansion of the digital economy [12].

The development of green algorithms entails the optimisation of computational methods and techniques for energy efficiency. This includes creating algorithms that consume less energy, reducing the amount of data that must be processed, and optimising the hardware used to implement the algorithms. By reducing energy consumption and minimising the use of non-renewable resources, green algorithms can help mitigate the negative impact of computing on the environment.

Green Algorithm is a website devoted to furthering and disseminating research in the field of green algorithms. The website provides researchers with a multitude of resources, including publications, data sets, and tools for measuring the energy consumption of algorithms. The methodology behind the Green Algorithms project is described in [13] with highlights the need for more research in this field and identifies several areas where green algorithms can make a significant impact, including energy-efficient data processing, renewable energy systems, and smart grids.

III. METHODOLOGY

Our objective is to analyse the voltage through capacitor C_1. We adopted the values for the circuit components and parameters suggested in [14]: $L = 19.2mH$, $C_1 = 10nF$, $C_2 = 100nF$, and $R = 1978.5\Omega$. The adoption of these values guarantees chaotic behaviour. To simulate the dynamics of the system, we applied the fourth-order Runge-Kutta method [15] [16]:

$$\begin{cases} x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ k_1 = f(t_n, x_n), \\ k_2 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_1\right), \quad t_{n+1} = t_n + h, \\ k_3 = f\left(t_n + \frac{h}{2}, x_n + \frac{h}{2}k_2\right), \\ k_4 = f(t_n + h, x_n + hk_3), \end{cases} \tag{5}$$

where $h > 0$ is the integration step size and x represents the voltage states described in Equation (1). In a first scenario, we rewrite Equation (5) proposing the following interval extension:

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2(k_2 + k_3) + k_4). \tag{6}$$

In a second scenario, we fixed the Runge-Kutta Equation shown in Equation (5), and proposed an interval extension of the model itself (Equation (1)), leading to the following representation:

$$\frac{dv_{C1}}{dt} = \frac{1}{C_1 R}v_{C2} - \frac{1}{C_1 R}v_{C1} - \frac{1}{C_1}i_d(v_{C1}) \tag{7}$$

Observe that, in the first scenario, we reduced the number of operations (specially multiplications) in the original equation by applying the associative property. On the other hand, we increased the number of operations in the second scenario by applying the distributive property. Table I summarises the simulated cases.

TABLE I: Summary of simulated cases.

Case	ODE Model	Runge-Kutta Model
Case 1	Equation (1)	Equation (5)
Case 2	Equation (1)	Equation (6)
Case 3	Equation (7)	Equation (5)

We use the software *Matlab*, version *R2022b*, in a computer based on a 12-core, CPU CORE i9-10920XE, 64 GB of RAM with a NVIDIA T400 4GB GPU to run the simulations. Finally, we calculate the amount of energy consumed in the simulation of each representation, considering the simulation time and the number of bits used, using the method proposed by Karatsuba's algorithm. The following equation can be used to estimate the number of bits:

$$N_{BITS} = 64 \times A + 64^{1.585} \times M, \tag{8}$$

where A and M represent the number of additions and multiplications, respectively. Once we had the number of bits, we estimated the amount of energy and the carbon footprint in each simulation case.

IV. RESULTS

As expected, as shown in Figure 2 and Figure 3, no significant differences are observed in the simulation results of the models considered. Figure 2 is the comparison between *Case 1* and *Case 2*, where we altered the Runge-Kutta notation. Therefore, it shows the voltage across capacitor C_1, using the formulation expressed in Equation (1) and considering interval extensions shown in Equation (5) and Equation (6). Figure 3 depicts the comparison between *Case 1* and *Case 3*, where we altered the differential equation for the voltage in capacitor C_1. Therefore, it shows the voltage across capacitor C_1, using the formulation expressed in Equation (1) and Equation (7) and considering Runge-Kutta notation shown in Equation (5). Despite the differences, the results highlight that a small change in the model led to a change in the results without significantly alter the accuracy.

Due to the fact that simulation time varies with each attempt, we simulated the systems together several times and compared the simulation times. Figure 4 demonstrates the ratios between the simulation times at each iteration, considering different representations of the Runge-Kutta model and also Equations (1) and (7). The typical mean of the time ratio was 1.0010 seconds for the first scenario and 1.0025 seconds for the second scenario. In practice, this means that the simulation time using Equation (5) is 0.10% greater than the simulation time of Equation (6). While the simulation time of

Authorized licensed use limited to: Maynooth University Library. Downloaded on September 15,2025 at 11:32:00 UTC from IEEE Xplore. Restrictions apply.

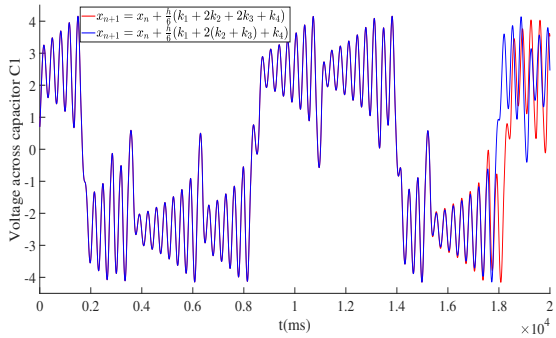

Fig. 2: Comparison between *Case 1* and *Case 2*: Voltage across capacitor C_1, using the formulation expressed in Equation (1) and considering interval extensions shown in Equation (5) and Equation (6)

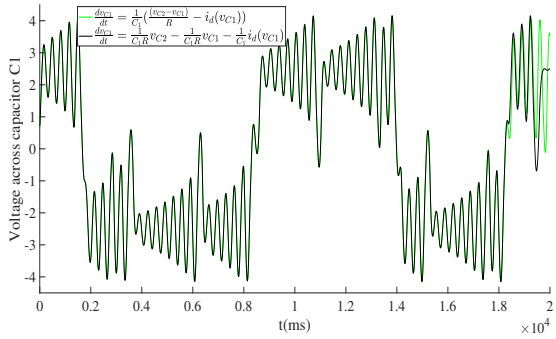

Fig. 3: Comparison between *Case 1* and *Case 3*: Voltage across capacitor C_1, using the formulation expressed in Equation (1) and Equation (7) and considering Runge-Kutta notation shown in Equation (5)

Equation (7) is 0.25% greater than the simulation time of Equation (1). Table II presents the the maximum and minimum times for the time occurrences.

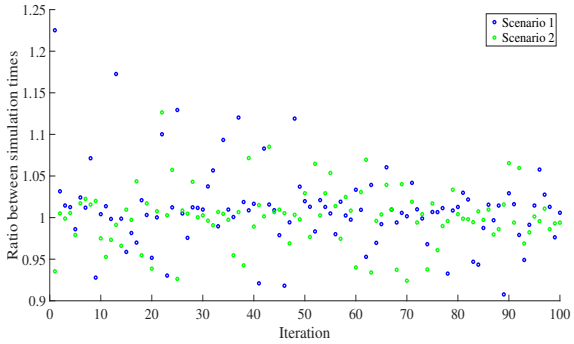

Fig. 4: ***Scenario 1***: Ratio between simulation times, considering comparison between *Case 1* and *Case 2*. ***Scenario 2***: Ratio between simulation times, considering comparison between *Case 1* and *Case 3*.

It may seem like a difference to be disregarded, but it is observed that we are referring to a few seconds of simulation. It is therefore worth analysing the difference in the amount of bits used in each case. We describe below each case in terms of the number of operations

TABLE II: Characteristics of the times of simulation.

Measure	Case 1	Case 2	Case 3
t_{max}	0.4761	0.4599	0.4801
t_{min}	0.4180	0.4191	0.4192

employed, considering the notation of the ODE that models the voltage in capacitor C_1 and the Runge-Kutta model used in the simulation. Note that *Case 2* reduced one multiplication operation compared to *Case 1*, while *Case 3* increased 5 multiplication operations compared to *Case 1*.

- *Case 1*: $26M + 19A$
- *Case 2*: $25M + 19A$
- *Case 3*: $31M + 19A$

Using the Karatsuba method to determine the number of bits, we have:

- *Case 1*: 2.0173×10^4 bits
- *Case 2*: 1.9444×10^4 bits
- *Case 3*: 2.3819×10^4 bits

The Figure 5 shows the amount of carbon emitted and the energy expended for each minute of simulation on the computer used. This represents $0.0205g$ of CO_2 and $0.0610Wh$ per second. Intuitively, we can relate the amount of carbon and energy spent per bit: $4.4186x10^{-7}g/bit$ of CO_2 and $1.3148x10^{-6}Wh/bit$ of energy.

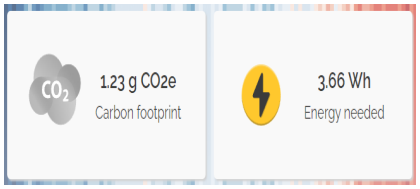

Fig. 5: Amount of carbon emitted and the energy expended for each minute of simulation.

For each iteration of the calculation, using the equations of *Case 2* represents a reduction of 3.63% in carbon emissions and energy consumption when compared to the formulations for *Case 1*. An increase of 18.07% is observed in energy consumption and CO_2 emissions when using the formulations of *Case 3*, compared to *Case 1*. A difference of 22.50% is observed between *Case 3* and *Case 2*.

A. Discussion

Unlike real-world applications where the simulation time is not always pre-established, we applied the Runge-Kutta method along a vector of 75001 positions, with an integration step of 10^{-6}, starting from zero. Thus, it is understood that the results were generated with a maximum of 75001 iterations. In practice, this represents a controlled environment in which we limit the maximum simulation time by establishing the number of iterations. In a different scenario where the total simulation time was longer, the differences could be much more significant.

The results emphasise that the simulation time is a critical factor in the computational performance of many systems. Various mathematical models have different levels of complexity, and therefore, the time required to simulate them can vary considerably. Extended simulation times may lead to increased energy consumption, which can be a significant problem for devices with limited power capacity. Hence, it is crucial to take into account the energy consumption

Authorized licensed use limited to: Maynooth University Library. Downloaded on September 15,2025 at 11:32:00 UTC from IEEE Xplore. Restrictions apply.

implications when selecting a simulation method. By opting for a more efficient simulation approach, such as one that employs simpler mathematical models or reduces the simulation time, we can reduce energy consumption and enhance the overall performance of the system.

Carbon footprint and energy consumption are crucial issues that need to be addressed to achieve environmental sustainability in computational systems. Green algorithms and adoption of simpler interval extensions are some of the solutions that can help reduce the environmental impact of computational systems. These techniques optimize energy consumption by reducing unnecessary computations and data transfers and minimising storage requirements. By developing and implementing the techniques presented in this work, we can reduce the energy consumption and carbon footprint of computational systems, leading to significant environmental benefits.

V. CONCLUSIONS

In this work, we demonstrate how the simple way of representing an equation can affect the energy consumption and carbon dioxide emissions during computational synthesis. The increase in the number of operations represented by the increase of five multiplication operations led to a greater use of the number of bits and consequently to a higher energy consumption. The overall results point to saving of approximately 22.50% in the values of emitted CO2 and consumed energy when using two different forms of representation of the same system.

As climate change becomes a growing concern, the importance of reducing carbon footprint has become increasingly significant. By reducing carbon footprint, we can mitigate the impact of our activities on the environment and take the necessary steps to combat climate change. Measuring and reducing carbon footprint are critical aspects of achieving a sustainable future, and a key element in achieving this is to approach computational systems in a correct way. Therefore, this work intends to encourage constant dialogue about this topic and how healthy computing techniques can contribute to global sustainability.

This work addressed only the issue of using different interval extensions and their impacts, which could be understood as a technique of green computing, but in fact, it focuses on the modelling of the problem. Future works can address the use of more complex interval extensions and compare the cost-benefit with the typical "mean value form" extension [17] [18]. Another approach could consider the use of optimised programming techniques for simulation, focusing on both optimised modelling and the adoption of optimised algorithms for computational simulation of dynamic systems.

ACKNOWLEDGEMENT

This material is based upon works supported by Science Foundation Ireland (SFI) for Josefredo Gadelha, Erivelton Nepomuceno and Marcio Lacerda under contract number SFI/21/FFp-P/10065 and by Maynooth University via a John and Pat Hume Doctoral (WISH) for Thalita Nazaré. The authors also wish to express their gratitude to the International Office of Maynooth University for funding this work.

REFERENCES

[1] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Comput. Surv.*, vol. 23, p. 5–48, mar 1991.

[2] M. Overton, *Numerical computing with IEEE floating point arithmetic*. SIAM, Society for Industrial and Applied Mathematics, reprint ed., 2004. Bibliografia. Índex.

[3] I. C. S. S. C. W. group of the Microprocessor Standards Subcommittee and A. N. S. Institute, *IEEE standard for binary floating-point arithmetic*, vol. 754. IEEE, 1985.

[4] M. R. Silva, E. Nepomuceno, G. F. V. Amaral, S. A. M. Martins, and L. G. Nardo, "Exploiting the rounding mode of floating-point in the simulation of chua's circuit," *Discontinuity, Nonlinearity, and Complexity*, vol. 7, no. 2, pp. 185–193, 2018.

[5] R. Corless, C. Essex, and M. Nerenberg, "Numerical methods can suppress chaos," *Physics Letters A*, vol. 157, no. 1, pp. 27–36, 1991.

[6] A. Tarafdar, S. Sarkar, R. K. Das, and S. Khatua, "Power modeling for energy-efficient resource management in a cloud data center," *Journal of Grid Computing*, vol. 21, no. 1, 2023.

[7] L. Lannelongue, J. Grealey, A. Bateman, and M. Inouye, "Ten simple rules to make your computing more environmentally sustainable," *PLoS Computational Biology*, vol. 17, no. 9, pp. 6–13, 2021.

[8] A. Abugabah and A. Abubaker, "Green computing: Awareness and practices," *2018 4th International Conference on Computer and Technology Applications, ICCTA 2018*, pp. 6–10, 2018.

[9] E. G. Nepomuceno and S. A. M. Martins, "A lower bound error for free-run simulation of the polynomial narmax," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 50–58, 2016.

[10] L. O. Chua, "The genesis of chua's circuit," Tech. Rep. UCB/ERL M92/1, EECS Department, University of California, Berkeley, Jan 1992.

[11] A. Karatsuba, "The complexity of computations," *Proceedings of the Steklov Institute of Mathematics-Interperiodica*, vol. 211, pp. 169–183, 1995.

[12] S. M. Alismail and H. A. Kurdi, "Green algorithm to reduce the energy consumption in cloud computing data centres," *Proceedings of 2016 SAI Computing Conference, SAI 2016*, pp. 557–561, 2016.

[13] L. Lannelongue, J. Grealey, and M. Inouye, "Green algorithms: Quantifying the carbon footprint of computation," *Advanced Science*, vol. 8, p. 2100707, June 2021.

[14] L. A. Aguirre, *Introdução à identificação de sistemas–Técnicas lineares e não-lineares aplicadas a sistemas reais*. Editora UFMG, 2004.

[15] R. Chapra and S. Canale, "Numerical methods for engineers," 1998.

[16] J. H. E. Cartwright and O. Piro, "The dynamics of runge–kutta methods," *International Journal of Bifurcation and Chaos*, vol. 02, pp. 427–449, 1992.

[17] Z. Galias, "The Dangers of Rounding Errors for Simulations and Analysis of Nonlinear Circuits and Systems - and How to Avoid Them," *IEEE Circuits and Systems Magazine*, vol. 13, no. 3, pp. 35–52, 2013.

[18] J. D. Bruguera, "Optimizing the representation of intervals," *Science of Computer Programming*, vol. 90, pp. 21–33, sep 2014.

Authorized licensed use limited to: Maynooth University Library. Downloaded on September 15,2025 at 11:32:00 UTC from IEEE Xplore. Restrictions apply.