Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Scalar hyperspherical tracking on a PIC18F4620 microcontroller

Seguimiento hiperesférico escalar en un PIC18F4620

Molina-Herrera, F.I.[1], Sandoval-Hernández, M.A.[1,3,4], Jiménez-Islas, H.[1], Martínez-González, G.M.[1], Quemada-Villagómez, M.L.[3], Vázquez-Leal, H.[5,*]

[1] Tecnológico Nacional de México en Celaya, Celaya Gto. Antonio García Cubas 600, Fovissste, 38010 Celaya, Gto.
[2] Universidad de Guanajuato, Mutualismo S/N, 38060 Celaya, Gto.
[3] Universidad de Xalapa. Carr. Veracruz-Xalapa 341-KM 2, Dos Vistas, Las Ánimas, Acueducto Ánimas, 91190 Xalapa-Enríquez, Ver.
[4] Centro de Bachillerato Tecnológico industrial y de servicios No. 190. Av 15, Venustiano Carranza, Carranza 2da Secc, 94297, Boca del Río, Ver.
[5] Universidad Veracruzana, Cto Aguirre Beltrán S/N, Zona Universitaria, 91090 Xalapa-Enríquez, Ver.
fi.molina@ugto.mx; xallitic476@gmail.com; hugo.jimenez@itcelaya.edu.mx; gloriam@iqcelaya.itc.mx; mairim_yata_01@hotmail.com; hvazquez@uv.mx*

Technological innovation: Low-cost educational implementation of continuation homotopy on a PIC microcontroller, allowing students to visualize numerical solutions on a GLCD.

Industrial application area: Intended for engineering students in numerical analysis and microcontroller courses.

Received: February 28, 2025
Accepted: October 20, 2025

Resumen

Para implementar la homotopía de continuación se requiere el uso de métodos numéricos como Newton–Raphson, el predictor de Euler y la interpolación; al crecer el tamaño o la no linealidad del sistema de ecuaciones, los cálculos se vuelven más intensivos y, por lo tanto, se requiere de computadoras con amplia RAM, procesadores rápidos y entornos de programación como Maple, MATLAB, Fortran 90 o C++. Este trabajo explora una alternativa de bajo costo: una implementación escalar del seguimiento hiperesférico en un microcontrolador PIC18F4620 de 8 bits, dispositivo sin coprocesador matemático ni las prestaciones de una PC convencional. El algoritmo se programa en C y se visualiza en un display monocromático gráfico de 128 × 64 píxeles, mostrando el trazado del camino homotópico y la visualización de las raíces. Se presentan cuatro casos de estudio; en los cuales el error en la determinación de la raíz se mantiene por debajo de 10^{-6}, pese a

las limitaciones de resolución de la pantalla. Se demuestra que conceptos avanzados de homotopía pueden materializarse en hardware modesto, convirtiéndose en un recurso didáctico valioso para asignaturas que combinan programación de microcontroladores y análisis numérico. Además, el artículo propone dos estrategias pedagógicas para su implementación: una dirigida a cursos de licenciatura en ingeniería y otra diseñada para bachillerato tecnológico con enfoque STEM, con el fin de evidenciar la aplicabilidad de las matemáticas en contextos reales de ingeniería y fomentar el aprendizaje interdisciplinario.

Palabras clave: Análisis numérico, discurso matemático escolar, homotopía de continuación con seguimiento hiperesférico, matemática educativa, raíces de ecuaciones.

Abstract

Implementing continuation homotopy typically relies on numerical techniques such as Newton–Raphson iteration, Euler prediction, and interpolation. As the size or nonlinearity of the system of equations increases, the computational demands escalate, necessitating workstations with ample RAM, high-performance processors, and software environments like Maple, MATLAB, Fortran 90, or C++. This study explores a low-cost alternative: a scalar implementation of hyperspherical tracking on the 8-bit PIC18F4620 microcontroller—hardware that lacks a mathematical coprocessor and the resources of a conventional PC. The algorithm, written in MikroC, drives a 128 × 64-pixel monochrome graphic display that plots the homotopy trajectory and marks the converged root. Four case studies demonstrate that, despite the display's limited resolution, the root error can be maintained below 10^{-6}. These results confirm that advanced homotopy concepts can be realized on modest hardware, providing a practical teaching tool for courses that integrate microcontroller programming with numerical analysis.

The article outlines two instructional pathways. The first targets undergraduate engineering courses, where students already have a foundation in numerical methods and embedded C. The second is tailored to technology-oriented baccalaureate programs, adopting a STEM (Pantoja-Amaro, 2020) perspective to highlight the real-world relevance of mathematics. Both pathways emphasize interdisciplinary learning: students connect algebra, analytic geometry, and differential calculus to hands-on electronics, deepening their understanding while developing transferable engineering skills.

Keywords: PIC microcontroller, homotopy continuation with spherical tracking, school mathematics speech, numerical analysis, educational mathematics.

1 Introduction

Many engineering problems lead to nonlinear algebraic equations (NAEs) that cannot be solved through simple rearrangement of terms (Baldor, 2019). When this happens, engineers rely on numerical methods instead (Buden & Faires, 2017; Chapra & Canale, 2015). The most well-known of these, Newton–Raphson (N-R), works effectively only when the initial guess is close to a root: each iteration finds at most one solution, solely if the starting point is appropriate. If

the number of real roots is unknown, or if multiple roots are located close together, randomly selecting starting points often results in N-R converging to the same answer repeatedly, while other roots remain elusive. These limitations highlight the necessity for broader, "global" techniques, such as continuation methods, to identify every possible solution from a wider range of starting values.

Continuation homotopy methods (HCM) offer a robust alternative numerical approach to local schemes by ensuring provable global convergence (Roychowdhury & Melville, 2006). This property is vital for locating several—or even all—solutions of a nonlinear system, especially in polynomial problems. Their capability to trace each solution set branch makes them well-suited for complex engineering tasks.

From a topological perspective, homotopy is defined as the continuous deformation of one object into another without interruption. In this context, an equation (or equation's system if the case) is continuously deformed into a different form (another equation or equation's system) to determine its roots (Zangwill & Garcia, 1984). Therefore, the HCM essentially consists of two parts: homotopy and numerical continuation, meaning numerical schemes occur during the deformation process.

Implementing HCM involves several numerical techniques, notably. Newton's homotopy (among others), for example, defines the deformation process that transforms an initial equation into its target form as the homotopy varies. The numerical continuation includes the Newton–Raphson iteration, Euler's predictor, and interpolation. As the system to be solved increases in size or nonlinearity, the associated computations become more intensive and, consequently, more time-consuming. In practice, optimal performance requires a computer with sufficient RAM and a high-performance processor capable of rapidly executing a large number of floating-point operations. The implementation also benefits from (and is often developed with) mathematical software environments such as Maple, MATLAB, Fortran 90, or C++.

HCM has been applied to obstacle-avoiding path planning in mobile robotics (Díaz-Arango *et al*, 2017); computing the admissible poses of a parallel manipulator with dual triangular platforms (Gallardo-Alvarado, 2019; Quemada-Villagómez *et al*., 2017); uncovering multiple operating points in the DC analysis of a tunnel-diode circuit (Sandoval-Hernández, 2019); and estimating sugar-solution concentration in a triple-effect evaporator (Quemada-Villagómez *et al*., 2017). In each case, the method's global search capability proved essential, delivering all feasible solutions without the trial-and-error reliance on initial guesses that trouble purely local techniques.

This study explores an alternative, scalar implementation of continuation homotopy using a Microchip PIC18F-series microcontroller. The device operates without a hardware math coprocessor or the computational resources of a personal computer, serving as a low-cost platform to demonstrate the algorithm's feasibility under constrained hardware conditions. Similarly, this work proposes two instructional strategies: one geared toward undergraduate engineering courses and another tailored to STEM-focused high school programs. The microcontroller approach is particularly valuable for regions with limited economic resources, providing a cost-effective solution for teaching and research in low-income settings. STEM education emphasizes real-world problem-solving, thereby enriching student learning (Garrigós-Sabaté & Valero-García, 2012; Martín-Páez *et al*., 2019). Its

interdisciplinary framework easily accommodates topics such as numerical analysis, microcontroller programming, and homotopy methods in university courses (Garrigós-Sabaté & Valero-García, J., 2012; Martín-Páez, 2019). By merging knowledge and techniques from multiple fields, STEM enables the development of new concepts, processes, skills, and perspectives.

2 Materials and equipment

In developing this system, the GCC 9.4.0 and Maple 2019 compilers were used in a Linux environment, while MikroC 7.6.0 (MikroElektronika, 2009) and Visual GLCD 2.7.1.0 (MikroElektronika, 2009) for Windows. Additionally, the 8-bit PIC18F4620 microcontroller, featuring a flash memory capacity of 64 KB (Microchip, 2008), was used. The microcontroller is configured to operate at a frequency of 32 MHz and is encapsulated in a 40-pin DIP package, among other relevant specifications. The development board used was the Mikroe EasyPIC v7 Connectivity, which includes a monochrome GLCD display with a 128x64 pixel KS0108 controller (MikroElektronika, 2014). The programming language used was C, compiled with the Mikroe compiler.

2.1 Homotopy of continuation with hyperspherical tracking

Consider the system of equations given by

$$f(x) = 0, f: R^n \to R^n \textbf{ (Eq. 1)}$$

In numerical continuation homotopy (HCM), a perturbation is introduced into the system $f(x)$, which incorporates a function $G(x)$ and a homotopy parameter τ. As a result, the original system is transformed into a homotopy system, expressed as follows:

$$H_i(f(x), \tau) = \tau f(x) + (1 - \tau)G(x) = 0, \textbf{ (Eq. 2)}$$

$H(x, \tau): R^{n+1} \to R^n i = 1, \cdots, n, \tau$Where, and is the homotopy parameter. Equation (2) represents any homotopy formulation that satisfies the following conditions:

- If $\tau = 0$ y $H(x, 0) = G(x) = 0$, a known solution is obtained.
- $\tau = 1 H(x, 1) = G(x) = 0$If y, the solution of the original system is recovered.

During the deformation process, the system is initiated at $\tau = 0$ and ideally progresses toward $\tau = 1$, the system starts at $\tau = 0$ and, ideally, reaches $\tau = 1$, where each intersection yields a solution vector of Eq. (2). These solution branches form continuous curves in the τ-space (Wayburn & Seader, 1987; Jiménez-Islas, 1996; Sandoval-Hernández *et al.*, 2025a). In Figure 1, four representative homotopy curves are shown. One of them illustrates a successful path extending from the initial point A ($\tau = 0$) to its corresponding root at $\tau = 1$. The remaining curves represent failure modes in which the homotopy continuation procedure either diverges or follows a branch that, although containing valid roots, is disconnected from the starting point and therefore cannot reach $\tau = 1$.

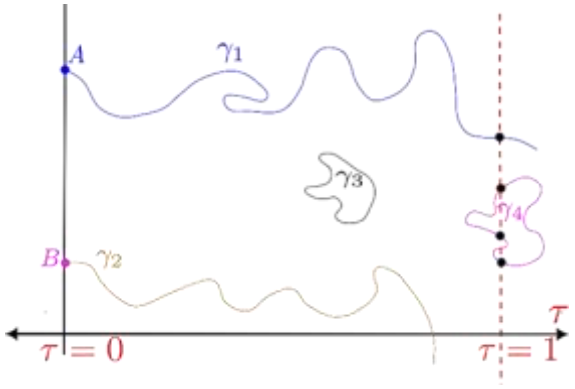


Figure 1. Family of homotopy curves.

The choice of G(x) determines the formulated homotopy. In this study, we propose using Newton's homotopy, as defined in (Wayburn & Seader, 1987; Jiménez-Islas, 1996; Sandoval-Hernández *et al.*, 2025a), yielding:

$$H_i(x,\tau) = f(x) - (1-\tau)G(x), x \in R^n, \textbf{(Eq. 3)}$$

x_0Where is the starting point (SP).

Implementation of the homotopy continuation method requires a procedure for mapping the homotopy path (Wayburn y Seader, 1987). One technique consists of placing a hypersphere S_2 with radius r and center O_2 along this path. The boundary of the hypersphere intersects the homotopy curve at two points, O_1 and O_3. As illustrated in Figure 2, O_1 represents the center from the previous iteration, whereas O_3 becomes the center for the next one.

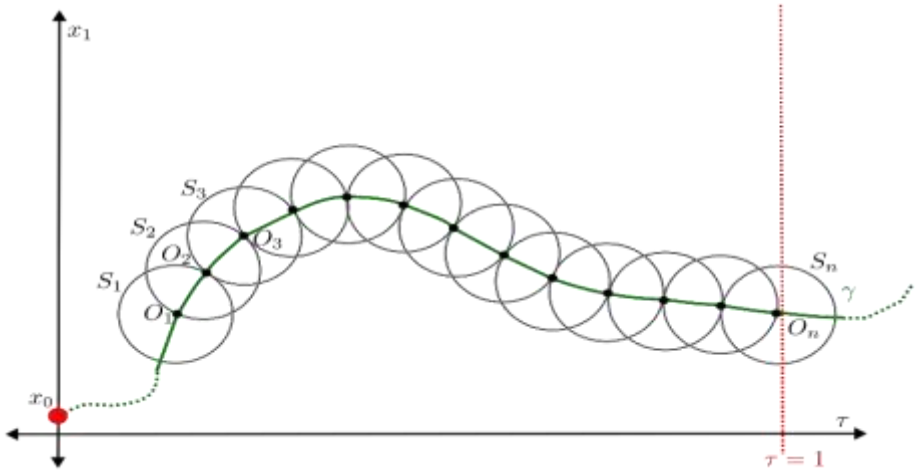


Figure 2. Homotopic path tracking.

The homotopy system is defined with n equations and $n+1$ variables, corresponding to the n variables of the original problem and the homotopy parameter τ (Díaz-Arango *et al.*, 2017; Wayburn & Seader, 1987; Jiménez-Islas, 1996; Sandoval-Hernández *et al.*, 2025a). In the hyperspherical tracking technique, an additional equation defining a $n+1$ -dimensional sphere is introduced to trace the homotopy path. The hypersphere equation is expressed as.

$$S_i(x,\tau) = \sum_{k=1}^{n}(x_k - c_k)^2 + (\tau - c_{n+1})^n - r^n = 0, \textbf{(Eq. 4)}$$

$r(c_k, c_{n+1})i = 1, \cdots N$ Where is the radius, represents the center of the hypersphere along the homotopy path, and denotes the iterations of the hyperspheres. By combining equations (3) and (4), the following system of equations is obtained:

$$H_s(x,\tau) = \begin{cases} H_1(x,\tau) = 0, \\ H_2(x,\tau) = 0, \\ \vdots \\ S_i(x,\tau) = 0, \end{cases} \textbf{(Eq. 5)}$$

$H_S(x,\tau)$Where represents the system used to define the homotopy path.

$\tau 1 f(x) = 0$During the tracking process, the center of the hypersphere is continuously updated, and the corresponding system of equations is solved iteratively. The algorithm begins with a prediction step that estimates the next point along the trajectory, followed by a correction step that ensures accurate tracking of the homotopy curve. The hypersphere center is then updated, and this cycle is repeated until the parameter reaches a value of. At this stage, interpolation and refinement procedures are applied to obtain the solution of.

$\tau\,\tau\, =\, 1\gamma\tau$The process continues until an additional root is located or the maximum number of iterations is achieved. During this iterative procedure, the parameter may cross the value several times in both directions. The predictor–corrector scheme enables precise tracking of the -homotopy curve through the hyperspherical algorithm. Figure 3 illustrates this process.

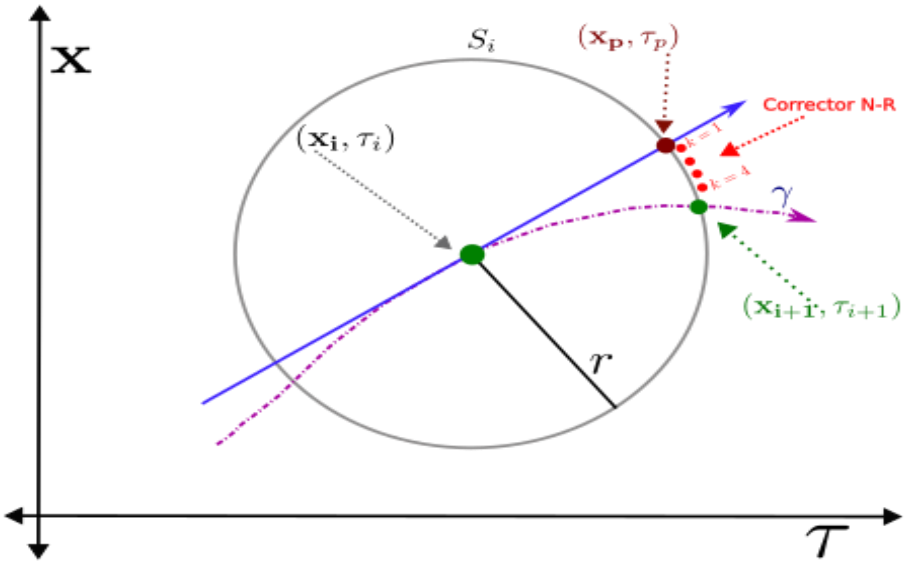


Figure 3. Predictor-corrector algorithm in the hypersphere.

The Euler predictor defines a tangent vector at each point along the homotopy path. The predicted point is obtained at the intersection between this tangent vector and the hypersphere. The Newton–Raphson (N–R) method serves as the corrector step, refining the intersection point between the homotopy trajectory and the hypersphere to produce a point closer to the solution of the homotopy system in the subsequent tracking iteration (Díaz-Arango *et al*, 2017; Wayburn & Seader, 1987; Jiménez-Islas, 1996; Sandoval-Hernández *et al*., 2025a). The N–R algorithm iterates until either the maximum number of iterations j_{max} is attained or convergence to a solution (x_{i+1}, τ_{i+1}) within a specified tolerance is achieved, as depicted in Figure 2. Figure 3 presents an example in which four correction iterations occur within the N–R method. By parameterizing equation (3) with parameter p, the homotopy curve $H^{-1}(x, \tau)$ is defined as $H^{-1}\big(x_i(p), \tau(p)\big)$. In equation (3), the system is reformulated with parameter p, incorporating the corresponding elements of the homotopy functions.

$$H_i\big(f(x(p)), \tau(p)\big) = f\big(x(p)\big) - \big(1 - \tau(p)\big) f(x_0),\ x \in R^n. \textbf{ (Eq. 6)}$$

$\frac{\partial \tau}{\partial p} x_0 H_i\big(f\big(x(p)\big), \tau\big) H$The homotopy path is constructed by generating a tangent vector passing through the initial point, enabling the prediction of the next point. The gradients are obtained by computing the partial derivatives of each homotopy function with respect to the parameterized elements. Defining the matrix, we obtain:

$$H = \begin{bmatrix} \frac{\partial H_1}{\partial x_1}\frac{\partial x_1}{\partial p} & \cdots \frac{\partial H_1}{\partial x_n}\frac{\partial x_n}{\partial p} & \frac{\partial H_1}{\partial \tau}\frac{\partial \tau}{\partial p} \\ \frac{\partial H_2}{\partial x_1}\frac{\partial x_1}{\partial p} & \cdots \frac{\partial H_2}{\partial x_n}\frac{\partial x_n}{\partial p} & \frac{\partial H_2}{\partial \tau}\frac{\partial \tau}{\partial p} \\ \vdots & \ddots \quad \vdots & \vdots \\ \frac{\partial H_n}{\partial x_1}\frac{\partial x_1}{\partial p} & \cdots \frac{\partial H_n}{\partial x_n}\frac{\partial x_n}{\partial p} & \frac{\partial H_n}{\partial \tau}\frac{\partial \tau}{\partial p} \end{bmatrix}, H \in M_{n,n+1}R \textbf{ (Eq. 7)}$$

Expressing (7) as a system in its matrix form, we have

$$\begin{bmatrix} \frac{\partial H_1}{\partial x_1} & \cdots & \frac{\partial H_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial H_n}{\partial x_1} & \cdots & \frac{\partial H_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial p} \\ \vdots \\ \frac{\partial x_n}{\partial p} \end{bmatrix} = -\begin{bmatrix} \frac{\partial H_1}{\partial \tau}\frac{\partial \tau}{\partial p} \\ \vdots \\ \frac{\partial H_n}{\partial \tau}\frac{\partial \tau}{\partial p} \end{bmatrix} \textbf{.(Eq. 8)}$$

By applying Cramer's rule to Equation (8), the solution of the system is obtained. After simplification and generalization using determinant properties, the following expression is derived:

$$\dot{y_k} = (-1)^k \|H_K\| \begin{bmatrix} \frac{\partial x_k}{\partial p} \\ \vdots \\ \frac{\partial \tau}{\partial p} \end{bmatrix}, k = 1, \cdots, n+1, \textbf{ (Eq. 9)}$$

$\|H_K\|, k = 1, \cdots, n$Where corresponds to the determinant with the ith column removed, following Cramer's rule. Equation (9) defines the slope of each variable component along the homotopy path, which is essential for computing the predictor vectors. These vectors are determined by calculating the slopes from Equation (9) and subsequently applying the Euler predictor algorithm, which yields:

$$x_p = x_0 + \Delta p \frac{dx}{dp}, \textbf{ (Eq. 10)}$$

$$\tau_p = \tau_0 + \Delta p \frac{d\tau}{dp}, \textbf{ (Eq. 11)}$$

Where x_p and τ_p represent the predicted values obtained using the Euler predictor (Faires y Burden, 2017), which lie on the surface of the hypersphere with radius r. In this study, the subscript $i = 1$ is used, indicating a one-dimensional case. To compute the horizontal advance Δp , equations (10) and (11) are applied. Thus, we obtain:

$$\Delta p = \pm \frac{r^2}{\sqrt{\sum_{k=1}^{n}\left[\frac{\partial x_k}{\partial p}\right]^2 + \left[\frac{\partial \tau}{\partial p}\right]^2}}. \textbf{ (Eq. 12)}$$

The sign will depend on the tracking direction, either right or left. To determine the point corresponding to the advancement of the homotopic path, the equation will be used:

$$p = p_0 + \Delta p. \textbf{ (Eq. 13)}$$

The prediction vectors act as the initial points for the hyperspherical correction stage, which identifies the intersection between the hypersphere and the homotopy path (Fig 2). This stage is based on the Newton–Raphson (N–R) method to solve the homotopy system (Eq. 3) together with the hypersphere constraint (Eq. 4). In multidimensional problems, the N–R update is determined by the Jacobian matrix and the function vector evaluated at the current estimate.

$$(x,\tau)_{k+1} = (x,\tau)_k - [J(H_s(x,\tau)_k)]^{-1} H_s(x,\tau)_k, \textbf{(Eq. 14)}$$

Where the values of the x vector will be added to the values of the predictor vector to obtain the correction vector. The determination of roots occurs at $\tau = 1$; however, accurately identifying the crossing point along the path can be challenging. Therefore, in the vicinity of this value, when transitioning from a number smaller than $(\tau_j < 1)$ to a number larger than $(\tau_{j+1} > 1)$, a preliminary step is required before reaching the root. This refinement process involves linear interpolation using the following formula:

$$\hat{x} = x_j + \frac{x_{j+1} - x_j}{\tau_{j+1} - \tau_j}(1 - \tau_j). \textbf{(Eq. 15)}$$

To generate the initial points for refinement (Finding Strategy Zero, FSZ), the variables x_{j+1} and τ_{j+1}, correspond to values for $(\tau > 1)$, while x_j and τ_j correspond to values for $(\tau < 1)$. Here, x represents the set of variables, and $\hat{x}$ denotes the interpolated values for each variable when $\tau = 1$, as shown in Figure 4. The values obtained from Equation (15) serve as the initial inputs for the Newton–Raphson method in the root refinement stage at $\tau = 1$. In this formulation, the update rule is given by:

$$x_{k+1} = x_k - \left[J\left(H_s(x_k)\right)\right]^{-1} f(x_k). \textbf{(Eq. 16)}$$

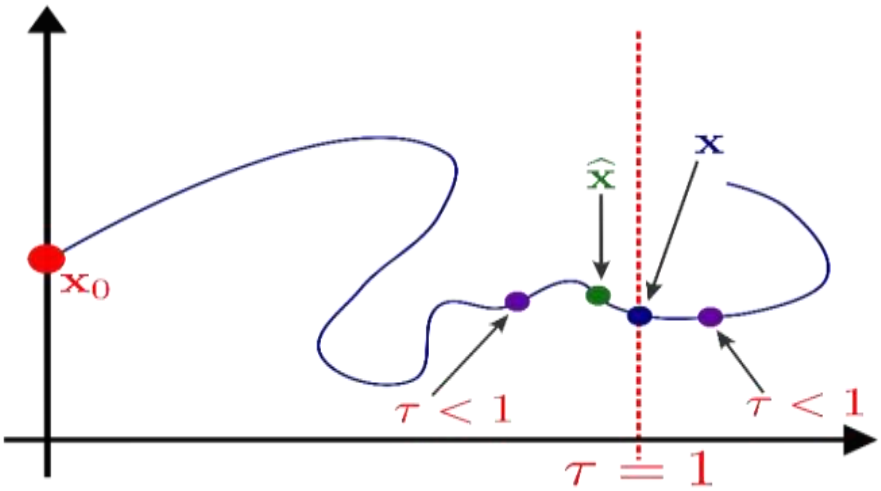


Figure 4. Interpolation for finding strategy zero.

In this study, we focus on the scalar case of x, reinterpreting the continuation homotopy literature within the context of School Mathematical speech (SMS) (Sandoval-Hernández *et al*., 2023; Uriza *et al*., 2015; Sandoval-Hernández *et al*., 2025b), to implement tracking on a microcontroller. Figure 5 depicts the one-dimensional hyperspherical homotopy flow. The diagram begins with the formulation of the homotopy equation and the hypersphere equation. The number of iterations corresponding to the hyperspheres is then determined, and the associated counter is initialized. Subsequently, a rectangular matrix is constructed, the respective determinants in one dimension are computed, and the horizontal advancement Δp is computed.

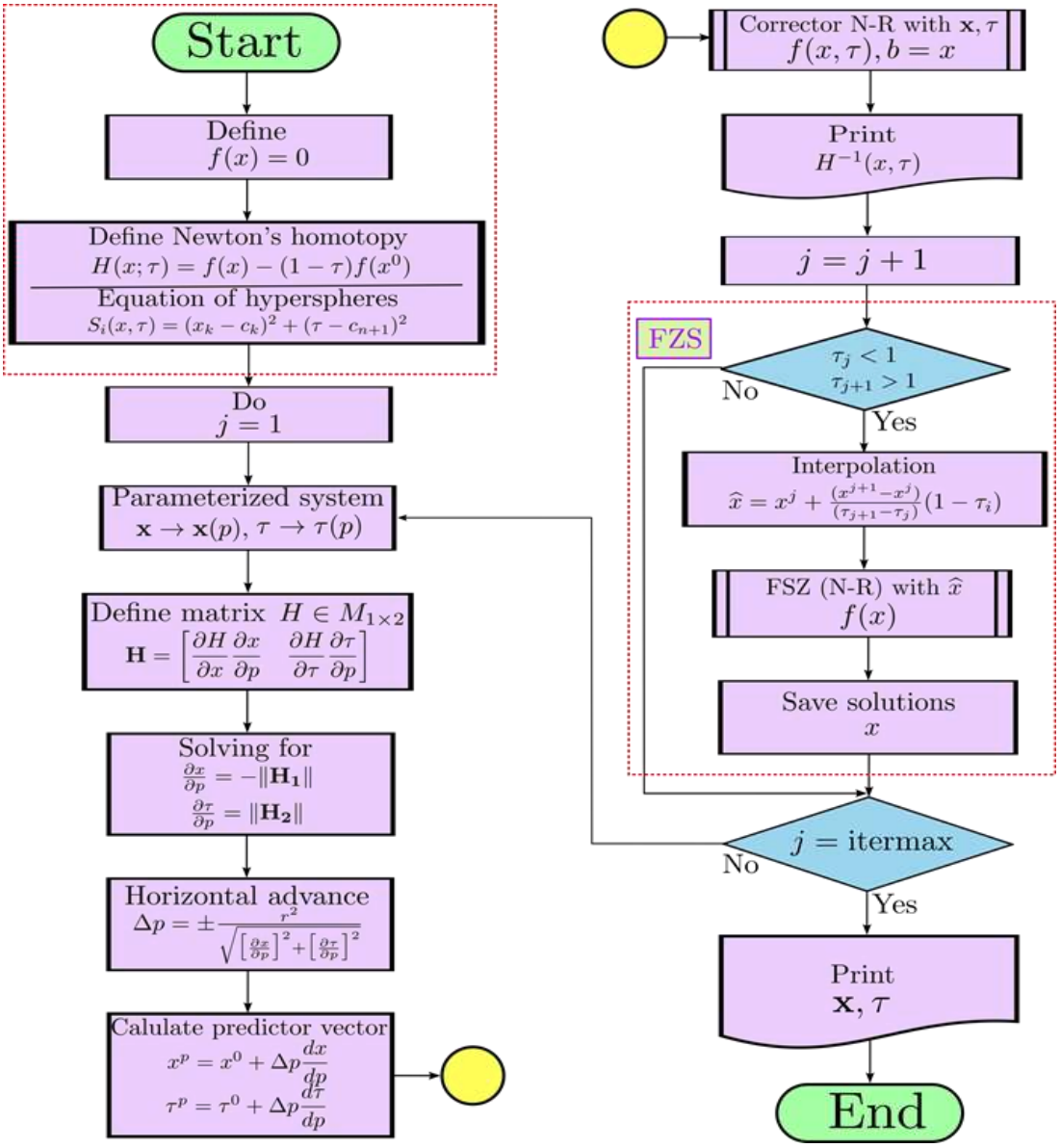


Figure 5. Homotopy flowchart with hyperspherical path tracking.

2.2 System design with PIC microcontroller

In Figure 6, the block diagram of the microcontroller-based homotopy implementation is shown. The architecture consists of a *main*() routine that calls three specialized subroutines and relies on four pre-designed GLCD screens to display status and results. Upon power-up, the graphical LCD is initialized and the I/O ports are configured according to the EasyPIC v7 Connectivity board (MikroElektronika, 2014).

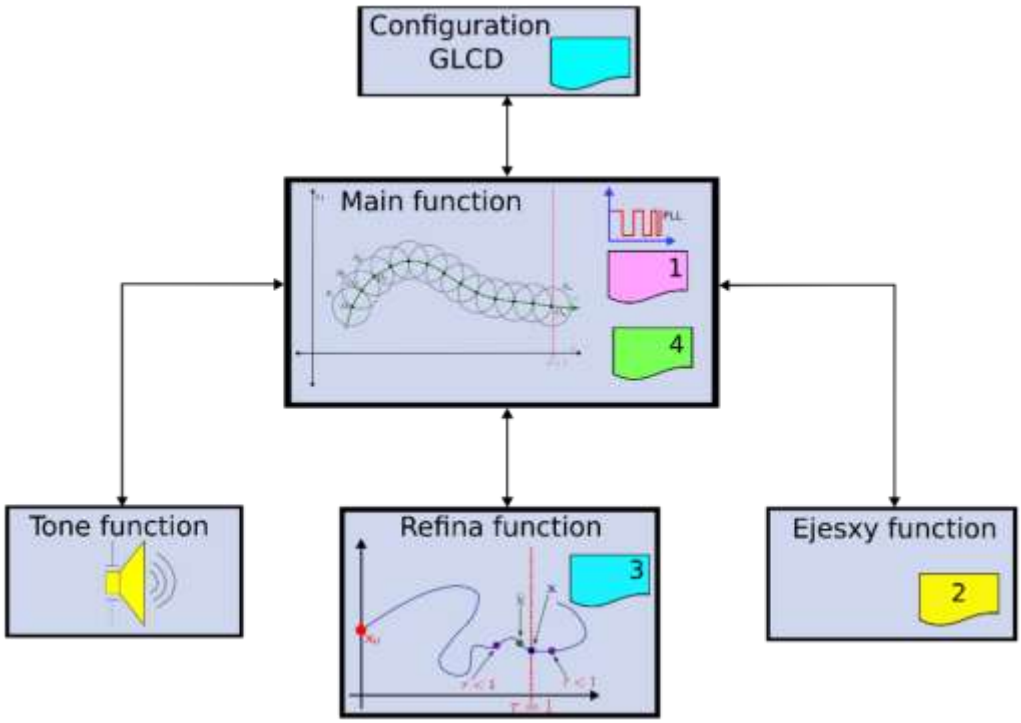


Figure 6. System design with PIC microcontroller.

Within *main*(), the internal phase-locked loop (PLL) is configured to drive the PIC18F4620 at 32 MHz. The routine then executes the one-dimensional hyperspherical homotopy algorithm outlined in Figure 5. Before tracking begins, the function *ejesxy*() is invoked to draw Cartesian axes on the GLCD—an essential step, since the built-in graphics functions in MikroC do not generate axes automatically. During execution of *main*(), the hyperspherical tracking routine is launched.

Upon entry, Screen 1 is displayed, showing the initial point, the selected hypersphere radius, and the equation whose roots will be computed. If the chosen seed fails to converge to any root, Screen 4 prompts the user to supply a new starting value; in the current prototype, this action requires recompiling the firmware. As an extension, a routine may be implemented to accept the starting point via a 4 × 4 keypad or a PC keyboard, following the input-handling guidelines in (Clavijo, 2011).

During execution, the algorithm plots the homotopy path on the GLCD using Screen 2. When the tracking routine satisfies the convergence criteria described in Figure 6, the program calls *refina*() function, which switches to Screen 3, displays the refined roots, and reports the number of hyperspheres traversed in the forward and reverse directions. The auxiliary routine *tone*() generates an 800 Hz tone for two seconds; it is triggered twice: immediately before the refined roots are shown and again after the final hypersphere counts are reported upon completion of tracking.

Figure 7 illustrates the four GLCD screens employed by the system. Screen 1 displays the initial seed, the selected hypersphere radius, and the equation under analysis. Screen 2 shows the direction of the hyperspherical trajectory of the homotopy. Screen 3 presents the roots obtained once tracking converges. Throughout the process, direction indicators appear: “SG.D” denotes

forward tracking, whereas “SG.R” signals reverse tracking. These labels refer solely to the algorithmic direction and must not be confused with the physical top or bottom position of the messages on the display.

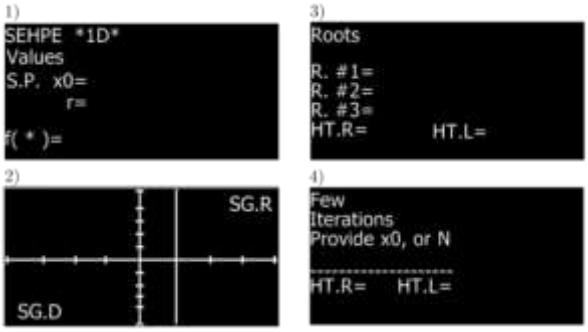


Figure 7. Text design on the system’s GLCD screen.

The last screen is activated only when the selected starting point is considered unsuitable. The GLCD font size is set to 3 × 5 pixels on all screens to ensure consistency and legibility. This interface design facilitates interpretation and monitoring of the homotopy process in educational context.

3 Results and discussion

Four case studies were proposed for implementing this prototype. The equations, radius size, and starting point were entered into the program at compile time.

Case study 1: Calculate the roots for the equation

$$x^3 - 1.473x^2 - 5.78x + 6.763 = 0, \textbf{(Eq. 17)}$$

$x_0 = 0.7r = 0.1.$ with hyperspherical tracking for an initial point and hypersphere radius

Solution. Figure 8 shows the equation's graph. The hyperspherical tracking and the three roots are also presented on each GLCD screen. In this case study, 73 hyperspheres were used for right tracking and 82 for left tracking.

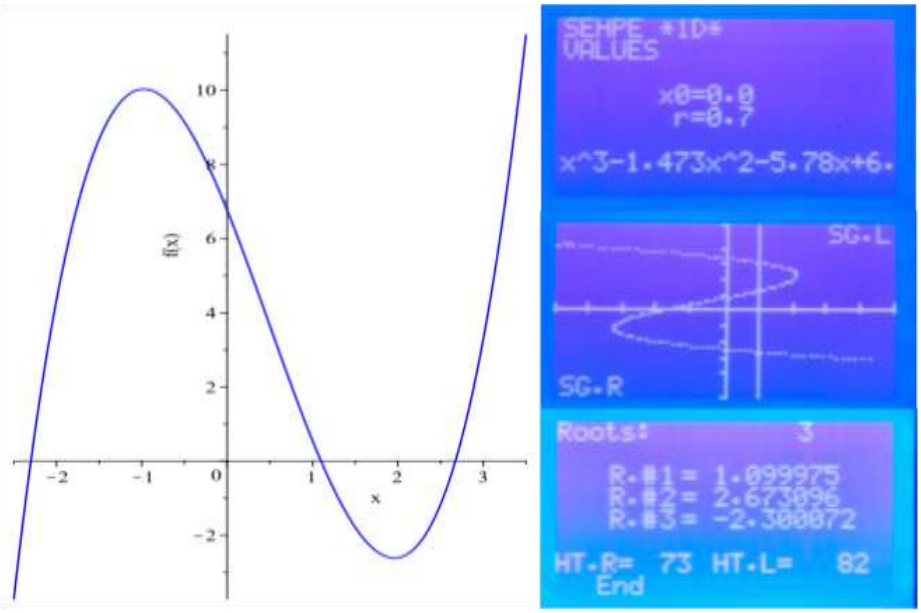


Figure 8. Graph of the equation, homotopy tracking and its roots.

Case study 2: Obtain the hyperspherical path tracking and its roots for the system of nonlinear simultaneous equations given by:

$$\begin{cases} x^2 + y^2 - 4 = 0, \\ x + y^2 - 4 = 0. \end{cases} \textbf{(Eq. 18)}$$

Solution. This system of equations can be

solved conventionally using homotopy, incorporating matrices and the Jacobian (Burden & Faires, 2017; Chapra & Canale, 2015). To solve this system using the one-dimensional homotopy method implemented on the microcontroller, we substitute x from the second equation into the first, expressing it as a function of y, thus obtaining:

$$y^4 - 7y^2 + 12 = 0. \textbf{ (Eq. 19)}$$

Figure 9 shows the graph of (19), the homotopic path, and the roots found on the GLCD screen. The starting point $x_0 = 0$ is and the $r = 0.1$. Substituting the solutions into any of the equations of (18), we obtain the solution pairs given by (0.000004,-1.999999), (0.000004, 1.999999), (1.000002798, 1.732050). In this case of study, 66 and 64 hyperspheres were used for tracking to the left and right direction, respectively.

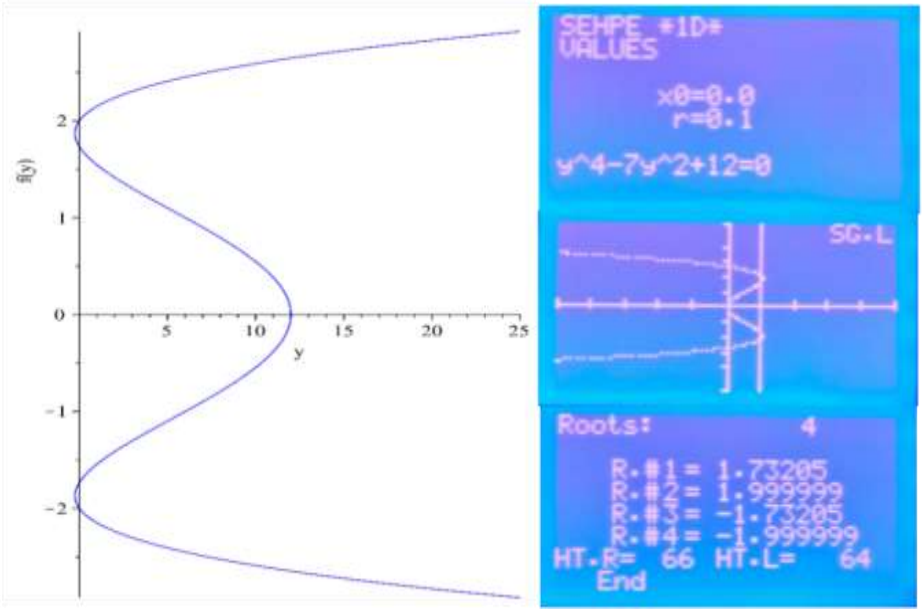


Figure 9. Graph of the equation, homotopy tracking and its roots.

Case study 3: Find the hyperspherical tracking and its roots for the system of nonlinear simultaneous equations:

$$\begin{cases} x - 2y + y^2 + y^3 - 4 = 0, \\ -x - y + 2y^2 - 1 = 0. \end{cases} \textbf{ (Eq. 20)}$$

Solution. Solving the first equation and substituting in the second one obtains

$$y^3 + 3y^2 - 3y - 5 = 0. \textbf{ (Eq. 21)}$$

To perform homotopic tracking, an initial value $x_0 = -0.55$ and $r = 0.1$ are proposed. Figure 10 shows the graph of equation (21), the hyperspherical tracking and the roots found in the GLCD of the electronic system with a PIC microcontroller.

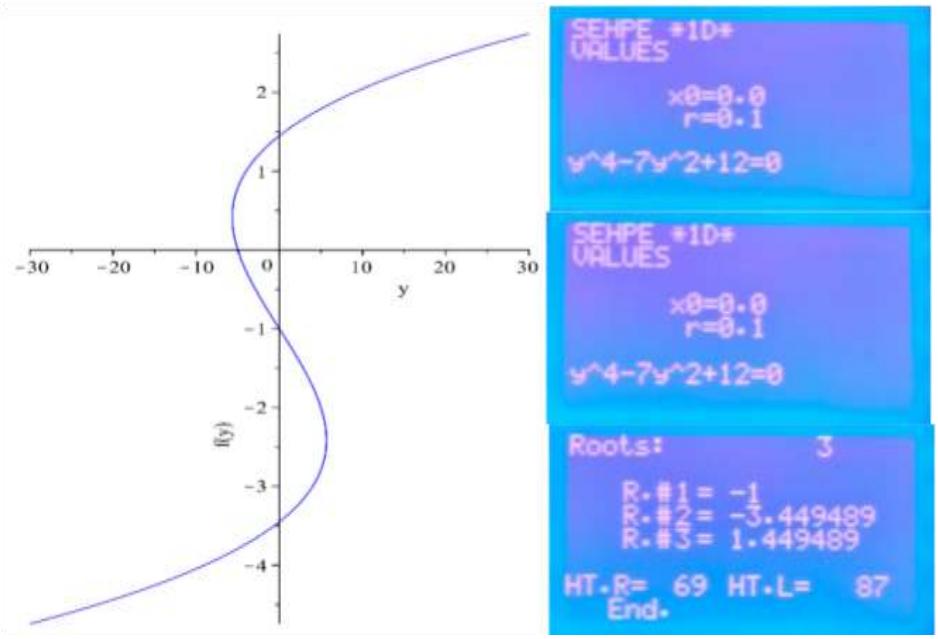


Figure 10. Graph of the equation, homotopy tracking and its roots.

$(2,-1)(26.247429,-3.449489),(1.7525566,1.449489)6987$ Homotopy tracking.

By substituting each of the roots of equation (21), the solutions of equation (20) are obtained:, and. In this case study, hyperspheres were used for right tracking and

Case study 4: Obtain the hyperspherical tracking for the system of nonlinear simultaneous equations for the tunnel diode circuit [8] given by

$$\begin{cases} y - \frac{0.3-x}{4} = 0, \\ y - 2x\exp(1-20x) - 1x10^{-9}\exp(40x) = 0, \end{cases} \textbf{(Eq. 22)}$$

yx Where represents the current and the voltage across the terminals of the tunnel diode, for the circuit in Figure 11.

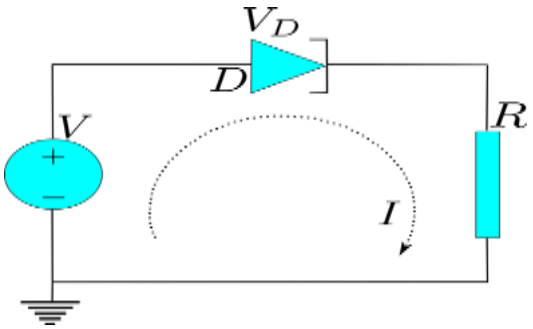


Figure 11. Circuit with a tunnel diode.

Solution. By solving y from the first equation and substituting it in the second equation of the system (22), we obtain:

$$0.075 - 0.25x - 2x\exp(1-20x) - 1 \times 10^{-9} - 9\exp(40x) = 0. \textbf{(Eq. 23)}$$

For hyperspherical tracking, a starting point x 0 =−0.4 and $r = 0.0001$ were proposed. Figure 12 shows the graph, the path and the roots obtained with the homotopy implemented in the system. Substituting the solutions presented in the GLCD of the system in (22) we obtain $(0.1619701,0.034507475)$, $(0.01884819,0.07028795250),(0.2755245,0.00611887500)$ In this case study, 103 and 4 hyperspheres were used in the left and right tracks, respectively.

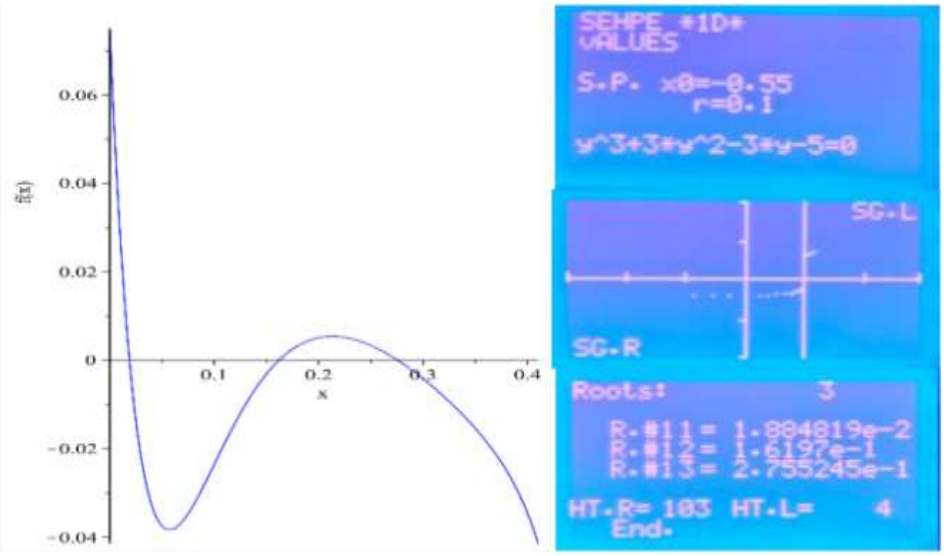


Figure 12. Graph of the equation, homotopy tracking and its roots.

Most published works on continuation homotopy focus on its theoretical foundations and high-level engineering applications—primarily in chemical process modeling and electronic circuit analysis—while offering limited attention to educational contexts. Moreover, comprehensive mathematical treatments can be found in standard references such as (Allgower & Georg, 2012), and several surveys provide detailed theoretical discussions (Wayburn & Seader, 1987; López-Gozález *et al*., 2023). Additional studies address practical implementations in robotics and kinematics (Díaz-Arango *et al*., 2017; Gallardo-Alvarado, 2019; Quemada-Villagómez *et al*., 2022). Only in recent years has a didactic adaptation for undergraduate students, based on hyperspherical tracking, been proposed (Quemada-Villagómez *et al*., 2022).

From the perspective of school mathematical discourse (Sandoval-Hernández *et al*, 2022; Uriza *et al*., 2015; Sandoval-Hernández *et al*., 2025b), the existing literature emphasizes formal rigor and professional outcomes but tends to overlook pedagogical considerations. Within academic curricula, continuation methods are generally introduced at the graduate level and are primarily applied to the solution of large systems of nonlinear algebraic equations. However, their ability to determine all possible roots makes them particularly suitable for introductory contexts involving single-variable equations—an educational opportunity that the present study seeks to address.

Case Study 1 addresses a third-degree polynomial equation whose three corresponding roots were determined. A limitation arises in the presentation of the results on the GLCD screen, which displays only six digits in the mantissa of each computed value. Table I compares the numerical solutions obtained through the electronic system implemented on the MikroE EasyPIC V7 connectivity board with the exact analytical solutions, the latter being computed using the homotopy routine described in (MikroElektronika, 2012). It is important to note that, in determining each root, an absolute error of $1x10^{-6}$ is introduced. A formula relating the relative error to the number of significant digits is presented in (Sandoval-Hernández *et al*., 2021).

Table 1. Significant digits of case study 1.

Case 1	Found roots	Exact roots	GLCD error
x_1	-1.099975	-1.099975	0.000000
x_2	2.673096	2.673097	0.000001
x_3	-2.300072	-2.300072	0.000000

Case Study 2 exhibited a similar situation when determining the first two roots of the equation. The variables shown in bold type y were computed using the proposed system, and the corresponding values x produced an absolute error of 4×10^{-6}. In contrast, the remaining two roots were found to be exact, displaying identical significant digits. The comparison results are summarized in Table 2. For Case Studies 3 and 4, the roots obtained showed complete correspondence in significant digits with the solutions calculated using the algorithm described in (Quemada-Villagómez, *et al*., 2022; MikroElektronika, 2012).

Table 2. Significant digits of case studies 2, 3 and 4.

Case 2	Found roots	Exact roots	GLCD error	Error in substitution
$x =$	0.000004	0.000000		0.000004
$\boldsymbol{y} =$	-1.999999	-2.000000	0.000001	
$x =$	0.000004	0.000000		0.000004
$\boldsymbol{y} =$	1.999999	2.000000	0.000001	
$x =$	1.000000	1.000000		0.000000
$\boldsymbol{y} =$	-1.732050	-1.732050	0.000000	
$x =$	1.000000	1.000000		0.000000
$\boldsymbol{y} =$	1.732050	1.732050	0.000000	
Case 3				
$x =$	2.000000	2.000000		0.000000
$\boldsymbol{y} =$	-1.000000	-1.000000	0.000000	
$x =$	1.752551	1.7525513		0.000000
$\boldsymbol{y} =$	1.449489	1.449489	0.000000	
$x =$	26.247449	26.247449		0.000000
$\boldsymbol{y} =$	-3.449489	-3.449489	0.00000	
Case 4				
$\boldsymbol{x} =$	0.01884819	0.01884819	0.0000000	
$y =$	0.07028795	0.07028795		0.000000
$\boldsymbol{x} =$	0.1619701	0.1619701	0.000000	
$y =$	0.03450748	0.03450748		0.000000
$\boldsymbol{x} =$	0.02755245	0.02755245	0.000000	
$y =$	0.0061188	0.00611885		0.000000

$\tau = 1[0.99999999,1.00000001]$ In the homotopy tracking performed across the case studies, it was observed that in Case 4 (Figure 12), where visualization was carried out using the GLCD, the homotopy path appeared to be traced along the line. This behavior occurred because the homotopy path intersections oscillate within the interval, and the limited pixel resolution of the GLCD restricts a finer graphical representation of the trajectory.

4, 3570754% Table 3 details the used memory, measured in Bytes, for each study case. Notably, in the implementation corresponding to case Bytes were used, representing of the total capacity of the microcontroller. This increase in ROM memory usage compared to the previous cases is due to incorporating two exponential functions in the algorithm. This data is relevant since it indicates how the complexity of the functions involved in an algorithm significantly influences the hardware resources required for its implementation and execution.

Table 3. ROM Memory Usage in Each Case Study.

Case 1	ROM in Bytes	ROM usage [%] on PIC
1	29657	45
2	29657	45
3	31031	47
4	35707	54

Although the prototype was developed on an EasyPIC v7 board, the microcontroller and GLCD can alternatively be assembled on a breadboard at substantially lower cost—an advantage for educational institutions in developing countries. However, implementing continuation homotopy on embedded hardware involves inherent limitations: the PIC18F microcontroller lacks a floating-point coprocessor, program memory is restricted, and the 128 × 64-pixel GLCD imposes constraints on graphical resolution.

3.4 Project-based teaching strategies

120Two instructional pathways are proposed. The first pathway, intended for university students who have completed coursework in numerical analysis—particularly those in electronic or mechatronic engineering—guides participants through a microcontroller-based homotopy project developed using MikroC and Visual GLCD. The second pathway, framed within a STEM methodology for technology-oriented high-school programs, introduces the same approach to illustrate the practical relevance of mathematics. At this level, a preliminary introduction to numerical analysis is required to facilitate comprehension of the method. Both pathways allow the use of equivalent compilers or graphical environments and can operate on minimal hardware configurations, consisting of a PIC18F microcontroller, a GLCD, and a programmer such as the PICkit. Each session is designed to last approximately minutes, although timing can be adjusted according to institutional schedules and student needs.

3.4.1 Didactic Sequence – Scalar Implementation of Hyperspherical Homotopy Tracking

The instructional sequence comprises a nine-session plan—consisting of an Opening session, seven Development meetings, and a Closing session—structured to guide engineering students from a symbolic-numeric understanding of the hyperspherical homotopy continuation method toward a fully scalar implementation running on a PIC18F4620 microcontroller with a KS0108 GLCD.

Δp Using MikroC PRO or any equivalent PIC/GLCD toolchain, each session specifies learning activities and explains the key elements from both pedagogical and technical perspectives: Jacobian calculation, tangent-vector determination and signed, adaptive step-size control, fine-tuning through linear interpolation, graphical user interface (GUI) construction, and validation against benchmark cases.

Session 1 – Opening: From Multivariable to Scalar Homotopy

Central purpose: Refresh the full SHH algorithm with the n-dimensional Jacobian

and connect it with the scalar flow implemented in hardware.

Key activities

- ❖ Brainstorm on the Jacobian's role in SHH.
- ❖ Comparative map: Fig. 5 (n D algorithm (Sandoval-Hernández *et al*., 2025) and Fig. 5 (scalar version).
- ❖ Maple demo: 2×2 Jacobian, 1 D derivative, calculation of Δp.
- ❖ Checklist of concepts transferred to the microcontroller.

Pedagogical and technical rationale

1. Reviewing the complete theory before simplifying to 1 D highlights methodological continuity.
2. The Fig. 5 in (Sandoval-Hernández *et al*., 2025a) and Fig. 5 parallel will serve as a guide throughout the sequence.
3. Δpn Understanding in dimension makes its scalar formula readily recognizable.

Session 2 – Development: EasyPIC v7, MikroC and GLCD Environment

Central purpose: Become familiar with the hardware platform, IDE and libraries that will support the algorithm.

Key activities

- ❖ Guided tour of the EasyPIC v7 board and its jumpers.
- ❖ Create a starter project in MikroC and program via mikroProg.
- ❖ Initialize the GLCD and display the test string "Hello SHH".
- ❖ Record FLASH/RAM usage and use the built-in Stopwatch.

Pedagogical and technical rationale

4. Handling the toolchain first prevents confusing hardware faults with numerical errors.
5. Knowing memory limits informs early optimization choices.

Session 3 – Development: Porting Predictor–Corrector and Signed Δp

$y'\Delta p = \pm r^2 / ||y'||$ Central purpose: Translate and into C; demonstrate forward and reverse tracking.

Key activities

- ❖ Copy Fx() and dFx().
- ❖ Compute the derivative and the Δp value.
- ❖ Add the flag dir $= +1 / -1$ and plot both tracks on the GLCD.
- ❖ Inspect variables with the MikroC debugger.

Pedagogical and technical rationale

6. The Jacobian is reduced to a derivative, reinforcing the transition n-D to 1D.
7. Δp Demonstrating direction inversion confirms that controls the tracking sense.
8. The debugger links internal data with on-screen results, strengthening numerical analysis.

Session 4 – Development: Fixed Radius Hypersphere Loop

Central purpose: Integrate predictor–corrector inside a loop and plot the complete trajectory.

Key activities

- ❖ Complete the shh_fixed module, storing centers.
- ❖ Run in forward and reverse directions and show hypersphere counters.
- ❖ Validate Case A by displaying the expected root.

Pedagogical and technical rationale

9. Δp The loop is implemented only after validation to guarantee correct steps.
10. The GLCD trace provides immediate visual feedback of the algorithm.

Session 5 – Development: Adaptive Step Size Control

hCentral purpose: Add convergence failure detection and halving of.

Key activities

- ❖ Insert the condition iterations > 8 or $|F| > 1 \times 10^2$.
- ❖ Δp Halve and emit a beep on every reduction.
- ❖ Record the number of reductions and measure time with Stopwatch.

Pedagogical and technical rationale

11. Control adapts the step to problem difficulty and reinforces algorithm robustness.
12. The beep makes the h reduction event tangible without looking at code.

Session 6 – Development: Fine Tuning: Linear Interpolation and Final Newton

$\lambda \rightarrow 1$Central purpose: Refine the real root when and display it on-screen.

Key activities

- ❖ Interpolate the last two points to approximate the root.
- ❖ Apply one additional Newton iteration.
- ❖ Show the root on results screen and store it in EEPROM.

Pedagogical and technical rationale

13. Interpolation is applied when a full trajectory exists; earlier it would lack data.
14. Saving the result in EEPROM preserves data after a board reset.

Session 7 – Development: Visual GLCD Interface Design

Central purpose: Create GUI screens and link them to the algorithm.

Key activities

- ❖ Use Visual GLCD wizard with KS0108 controller.
- ❖ Design start up, tracking, results and error screens.
- ❖ Generate code and add it to the main project.

Pedagogical and technical rationale

15. Separating the presentation layer from the numerical core improves maintainability and eases design changes.
16. The rapid-application-development (RAD) approach accelerates GUI development.

Session 8 – Development: Full Integration and Internal Testing

Central purpose: Merge all modules in SHH_Newton_Lineal.c and measure multi-criteria metrics.

Key activities

- ❖ Create the master project and compile without warnings.
- ❖ Run Cases A and B, measure steps, time, FLASH and RAM.
- ❖ Document changes and results in the code header.

Pedagogical and technical rationale

17. Integration before final validation allows debugging while there is still time to fix issues.
18. Metrics quantify efficiency and help fine-tune parameters if necessary.

Session 9 – Closing: Validation with Six Cases and Final Reflection

Central purpose: Demonstrate the routine on six benchmark functions and reflect on lessons learned.

Key activities

- ❖ Each team solves two assigned cases, completes the global table.
- ❖ Present the GLCD route and root, explain the main challenge overcome.

❖ Write a 1-1-1 reflection: clear concept, difficult concept, improvement proposal.

Pedagogical and technical rationale

19. Validating against published roots proves precision and robustness.
20. Presenting develops technical communication competence.
21. The final reflection closes the learning cycle and informs future improvements.

3.4.2 STEM Project Interactive Scalar Homotopy on PIC18F4620

The following didactic sequences will be carried out to assist students in implementing the homotopy system. The summary outlines a six-week STEM project for 6th semester Technological High School students (Electronics or Mechatronic specialization). The activity links Mathematics (Algebra, Analytic Geometry, Differential/Integral Calculus), Electronics, and Programming by implementing and visualizing the scalar hyperspherical homotopy continuation method (SHH) on a PIC18F4620 microcontroller with GLCD.

General and Specific Objectives

General objective: build a didactic prototype that solves $F(x) = 0$ for real roots and draws the homotopy path in real time on the GLCD, demonstrating the transition from theoretical calculus to embedded implementation.

❖ Explain the circle equation as a 2D hypersphere guiding the Euler predictor.
❖ Translate predictor–corrector and $\Delta p = \pm\, r^2/|y'|$ from Maple to MikroC.
❖ Optimize FLASH (< 60 kB) and RAM (< 2 kB) usage while keeping $|F(x)| < 10$.
❖ Design a four-screen GUI (start up, tracking, results, error) with Visual GLCD.
❖ Validate three benchmark polynomials against Maple and discuss execution metrics.

Table 4. shows Timeline and Milestones.

Table 4. Timeline and Milestones (8 weeks).

Week	Focus	Key Deliverable
1	Mathematical bridge: circle - hypersphere, derivative refresher	Infographic “From Circle to Homotopy”
2-3	Technology set-up: EasyPIC v7, MikroC, GLCD “Hello SHH”	Project blinky_glcd.c running
4-5	Core algorithm: port Fx, dFx, predictor–corrector, signed Δp	Screenshot of first hypersphere on GLCD
5-6	Adaptive control & fine-tuning: h reduction, interpolation + NR	Table Steps–Time–Reductions; root in EEPROM
7	GUI design in Visual GLCD	VGLCD file + 1min demo
8	Global validation: three functions, comparison vs Maple	Final report PDF + 8 min oral presentation

Competencies Developed

❖ Mathematics 1-2-4-5: modelling, problem-solving, error analysis.
❖ Autonomous Learning: independent consultation of MikroElektronika manuals.
❖ Collaborative Work: coder/tester/GUI roles documented in a shared log.
❖ Communication 5: graphical reporting, metric interpretation, oral defense.

Assessment Strategy

❖ A four-dimensional rubric evaluates mathematical rigor, code quality, analytical communicative skills, and collaboration. Each team submits:
❖ Source code repository with changelog

- ❖ Video demo and GUI screenshots
- ❖ Final PDF report and metric table

Required Resources

- ❖ EasyPIC v7 board + PIC18F4620 (DIP 40)
- ❖ MikroC PRO for PIC and Visual GLCD licenses
- ❖ MikroElektronika manuals and datasheets
- ❖ Base code homotopia1Dc3.c and online Maple for verification

Educational Impact

Students witness abstract calculus concepts materialize on hardware, linking circle geometry, derivatives, and numerical methods to real-time graphics. The activity nurtures interdisciplinary thinking, fosters technical communication, and lays the groundwork for sensor-driven homotopy or low-power optimization extensions.

This eight-week STEM project invites 6th-semester Electronics students to take the scalar hyperspherical homotopy continuation method (SHH) from textbook to hardware. Working on an EasyPIC v7 board with a PIC18F4620 and GLCD, they port the predictor-corrector algorithm in C, compute the signed step Δp, and plot each hypersphere while the root of a cubic equation converges in real time. The timeline moves from a mathematical bridge linking the familiar circle equation to the project's "hypersphere" through technology setup ("Hello SHH" on the GLCD), core algorithm migration, adaptive step-size control with fine-tuning, GUI design in Visual GLCD, and a final validation against three benchmark polynomials verified in Maple. The activity fuses Algebra, Analytic Geometry, Differential Calculus, Programming, and Electronics by turning abstract calculus into animated graphics and beeps on a real microcontroller. It leaves the class with a fully documented, extensible prototype ready for sensor-driven or low-power follow-ups.

Conclusions

Despite the challenges inherent in implementing homotopy on an embedded system—such as the absence of a mathematical coprocessor and the limited resolution of the 128 × 64-pixel GLCD display—the proposed implementation achieved satisfactory results in all study cases. The Newton–Raphson (NR) algorithm and the Euler predictor were able to estimate subsequent values for tracing the homotopic path. Likewise, the interpolation procedure and the NR routine in the FZS stage successfully located the numerical roots.

1×10^{-6}The maximum error recorded in determining the roots on the GLCD was. All root values obtained in the study cases were validated using Maple. Furthermore, despite the hardware limitations, in Case Study 4 the path tracking was not completely displayed on the GLCD due to pixel constraints; however, no observable numerical error was detected.

Overall, the system demonstrated robust performance and effective visualization capabilities for hyperspherical tracking, yielding results that indicate high reliability in its implementation and representing a significant achievement given the hardware constraints.

The numerical methods involved in the scalar implementation of the continuation homotopy method offer valuable opportunities to clarify several geometric and analytical concepts, such as numerical derivatives, the Euler method, linear interpolation, and the numerical solution of systems of equations. The numerical analysis employed in the implementation allows students to learn these concepts through structured programming practices. Moreover, the approach naturally

integrates concepts from algebra, analytic geometry, and calculus. This represents a valuable opportunity, particularly for high-school students, as it enables them to recognize how these mathematical principles are applied to determine the roots of nonlinear equations in cases where traditional algebraic methods are insufficient.

$x_o\tau 1$Likewise, all numerical algorithms employed constitute the HCM while the homotopy deformation process is taking place. In this process, the homotopy acts as a continuous deformation of the original equation, starting from a known initial value and progressing toward the determination of the roots. The final form of the equation is obtained when the homotopic parameter reaches a value of.

Implementing continuation homotopy with scalar hyperspherical tracking on a microcontroller resignifies the current mathematical discourse in education (Sandoval-Hernández *et al*., 2022; Uriza *et al*., 2015). It adds a new perspective to the state-of-the-art literature on homotopy continuation methods. Scalar implementations have received little attention, and homotopy algorithms are generally viewed as the exclusive domain of desktop computers. Demonstrating their viability on a resource-limited microcontroller challenges this assumption and widens the horizon for research and classroom applications.

Furthermore, we believe that integrating all elements into a practical project highlights the importance of combining theory and practice in scientific and technical education, preparing students to address real-world challenges in their future careers. Additionally, the components used in its implementation can be purchased separately and simply connect the microcontroller and the PIC on a breadboard. Future work will concentrate on applying the proposed teaching strategies at the engineering and high school levels. These approaches can be tailored to fit the institutional environment and the students' economic and academic situations. The overarching aim is to cultivate research skills and inspire a lasting interest in science and engineering.

Acknowledgements

The authors would like to thank the TecNM in Celaya for the support received.

Declaration of interests statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

Allgower, E. L., & Georg, K. (2012). *Numerical continuation methods: An introduction* (Vol. 13). Springer Science & Business Media.

Baldor, A. (2019). *Álgebra*. Patria Educación.

Burden, R. L. & Faires, J. D. (2017). *Numerical methods* (9th ed.). Cengage Learning.

Clavijo, R. (2011). *Diseño y simulación de sistemas microcontrolados en lenguaje C*. MikroElektronika.

Chapra, S. C., & Canale, R. (2015). *Métodos numéricos para ingenieros*. McGraw-Hill.

Díaz-Arango, G., Vázquez-Leal, H., Hernández-Martínez, L., Sanz-Pascual, M. T., & Sandoval-Hernández, M. (2017). Homotopy path planning for terrestrial robots using spherical algorithm. *IEEE Transactions on Automation Science and Engineering, 15*(2). https://doi.org/10.1109/TASE.2016.2638208

Gallardo-Alvarado, J. (2019). An application of the Newton homotopy continuation method for solving the forward kinematic problem of the 3-RRS parallel manipulator.

Mathematical Problems in Engineering, 2019, Article 3123808. https://doi.org/10.1155/2019/3123808

Garrigós-Sabaté, J., & Valero-García, J. (2012). Hablando sobre aprendizaje basado en proyectos con Júlia. *Revista de Docencia Universitaria, 10*(3). https://doi.org/10.4995/redu.2012.6017

Jiménez-Islas, H. (1996). SEHPE: Programa para la solución de sistemas de ecuaciones no lineales mediante método homotópico con seguimiento hiperesférico. *Avances en Ingeniería Química, 6*(2).

López-González, M. de la L., Quemada-Villagómez, M. L., Martínez-González, G. M., Oliveros-Muñoz, J. M., & Jiménez-Islas, H. (2023). A novel predictive homotopic path tracking algorithm to solve non-linear algebraic equations. *The Canadian Journal of Chemical Engineering, 101*(6), 3382–3404. https://doi.org/10.1002/cjce.24694

Martín-Páez, T., Aguilera, D., Perales-Palacios, F. J., & Vílchez-González, J. M. (2019). What are we talking about when we talk about STEM education? A review of literature. *Science Education, 103*(4), 799–822. https://doi.org/10.1002/sce.21522

Mendoza, J. C. (2011). *Diseño y simulación de sistemas microcontrolados en lenguaje C*. C Colombia.

Microchip Technology Inc. (2008). *PIC18F2525/2620/4525/4620 data sheet: 28/40/44-pin enhanced flash microcontrollers with 10-bit A/D and nanoWatt technology* (Publication No. 39626E). https://ww1.microchip.com/downloads/en/devicedoc/39626e.pdf

MikroElektronika. (2009). *MikroC PRO for PIC user manual*. https://download.mikroe.com/documents/compilers/mikroc/pic/mikroc-pic-manual-v101.pdf

MikroElektronika. (2012). *Visual GLCD: GLCD GUI design software*. https://www.mikroe.com/visual-glcd

MikroElektronika. (2014). *EasyPIC V7 connectivity user's guide*. https://download.mikroe.com/documents/full-featured-boards/easy/easypic-v7/easypic-v7-manual-v104c.pdf

Olvera-Martínez, M. C., Reyes-Rodríguez, A. V., Campos-Nava, M., Torres-Rodríguez, A. A., & Soto-Campos, C. A. (2022). El enfoque STEM y el aprendizaje de las matemáticas. *Unión. Revista Iberoamericana de Educación Matemática, 18*(66).

Pantoja-Amaro, L. F., Peña-Aguilar, J. M., & Mendoza-Torres, C. P. (2020). Desarrollo de habilidades STEM en media superior como mecanismo para impulsar la continuidad en educación superior: Caso programa Bases de Ingeniería. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo, 10*(20).

Quemada-Villagómez, M. L., Jiménez Islas, H., López-González, M. de la L., & Oliveros-Muñoz, J. M. (2022). Enseñanza del método de continuación homotópica con seguimiento hiperesférico para estudiantes de ingeniería. *Acta Universitaria, 32*, Article e3358. https://doi.org/10.15174/au.2022.3358

Roychowdhury, J., & Melville, R. (2006). Delivering global DC convergence for large mixed-signal circuits via homotopy/continuation methods. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 25*(1), 66–78. https://doi.org/10.1109/TCAD.2005.852461

Sandoval-Hernández, M., Álvarez-Gasca, M., Contreras-Hernández, A. D., Pretelín-Canela, J. E., Palma-Grayeb, B. E., Jiménez-Fernández, V. M., *et al*. (2019). Exploring the classic perturbation method for obtaining single and multiple solutions of nonlinear algebraic problems with application to microelectronic circuits. *International Journal of Engineering Research & Technology, 8*(9). https://www.ijert.org/exploring-the-classic-perturbation-method-for-obtaining-single-and-multiple-solutions-of-nonlinear-algebraic-problems-with-application-to-microelectronic-circuits

Sandoval-Hernández, M., Vázquez-Leal, H., Filobello-Niño, U., De-Leo-Baquero, E., Bielma-Pérez, A. C., Vichi-Mendoza, J. C., Álvarez-Gasca, O., Contreras-Hernández, A. D., Bagatella-Flores, N., Palma-Grayeb, B. E., Sánchez-Orea, J., & Cuellar-Hernández, L. (2021). The quadratic equation and its numerical roots. *International Journal of Engineering Research & Technology, 10*(6), 301–305. https://www.ijert.org/the-quadratic-equation-and-its-numerical-roots

Sandoval-Hernández, M. A., Vázquez-Leal, H., Huerta-Chua, J., Filobello-Niño, U., & Mayorga-Cruz, D. (2022). La didáctica del cálculo integral: El caso de los procedimientos de integración. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo, 13*(25). https://doi.org/10.23913/ride.v13i25.1245

Sandoval-Hernández, M., Jiménez-Islas, H., Vázquez-Leal, H., Quemada-Villagómez, M., & López-González, M. de la L. (2025a). Exploring homotopy with hyperspherical tracking to find complex roots with application to electrical circuits. *Open Mathematics, 23*(1). https://doi.org/10.1515/math-2024-0115

Sandoval-Hernández, M. A., López-Velez, G. C., Filobello-Niño, U. A., Morales-Alarcón, G. J., Jiménez-Fernández, V. M., & Vázquez-Leal, H. (2025b). Algoritmo iterativo para calcular raíces cuadradas y cúbicas para estudiantes de educación media: Iterative algorithm to calculate square and cubic roots for secondary and high school students. *Revista Digital: Matemática, Educación e Internet, 25*(2). https://doi.org/10.18845/rdmei.v25i2.7604

Uriza, R. C., Espinosa, G. M., & Gasperini, D. R. (2015). Análisis del discurso matemático escolar en los libros de texto: Una mirada desde la Teoría Socioepistemológica. *Avances de Investigación en Educación Matemática, 2015*(8), 9–28. https://doi.org/10.35763/aiem.v1i8.123

Wayburn, T. I., & Seader, J. D. (1987). Homotopy continuation methods for computer-aided process design. *Computers & Chemical Engineering, 11*(1), 7–25. https://doi.org/10.1016/0098-1354(87)80002-9

Zangwill, I. J., & Garcia, C.B. (1984). *Pathways to solutions, fixed points, and equilibria.* Prentice-Hall.