

# Sobre el uso de métodos de Arnoldi para la continuación numérica de puntos estacionarios



*Arnoldi methods for numerical continuation of stationary points*

Zenaida Natividad Castillo Marrero.<sup>1</sup>, Gustavo Adolfo Colmenares Pacheco.<sup>2</sup> & Víctor Oswaldo Cevallos Vique.<sup>3</sup>

Recibido: 24-04-2020 / Revisado: 15-05-2020 / Aceptado: 25-06-2020 / Publicado: 03-07-2020

## Abstract.

DOI: <https://doi.org/10.33262/cienciadigital.v4i3.1385>

This work presents an analysis of a continuation algorithm with an embedded Arnoldi eigen solver for the numerical computation of solutions of nonlinear systems  $G(x, \alpha) = 0$ , where  $x$  is a vector in  $\mathbb{R}^n$  and  $\alpha$  is a parameter which takes values in a given interval. This technique allows us to detect and predict particular solutions when computing the eigenvalues of the associate Jacobian matrix, and simultaneously to get the solution of linear systems in each iteration. This method could be applied to problems of electrical engineering, chemical reactions or coating process. The main idea is to embed an Arnoldi eigenvalue solver in a continuation algorithm to compute solutions of a nonlinear system in order to get additional information of these solutions, such as the stability or the periodicity. An example of the usability of this technique is presented, with preliminary results on models used in the bibliography of this topic. The results are encouraged, and show the reliability of this approach in the accurate detection of critical points.

**Keywords:** Path following, Nonlinear systems, Block Arnoldi method, Stability Analysis, Critical points.

## Resumen

En este trabajo se describe un método de continuación numérica, el cual utiliza descomposiciones de Arnoldi para el cálculo de soluciones de un sistema de la forma

<sup>1</sup> Escuela Superior Politécnica de Chimborazo, Facultad de Ciencias, Escuela de Estadística, Grupo CITED. Riobamba, Ecuador, zenaida.castillo@esPOCH.edu.ec

<sup>2</sup> Universidad de Investigación Experimental Yachay, Escuela de Matemáticas y Ciencias Computacionales. Urcuquí, Ecuador, gcolmenares@yachaytech.edu.ec

<sup>3</sup> Escuela Superior Politécnica de Chimborazo, Facultad de Administración de Empresas, Escuela de Finanzas, Grupo CITED, Riobamba, Ecuador, victor.cevallos@esPOCH.edu.ec

$G(x, \alpha) = 0$ , donde  $G$  es un funcional no lineal que depende de un vector  $x$  en  $\mathbb{R}^n$  y de un parámetro  $\alpha$  que toma valores en un intervalo dado. El uso de estas técnicas nos permite predecir y detectar puntos de interés cuando se obtiene información sobre algunos autovalores de la matriz Jacobiana asociada, al mismo tiempo que se resuelven eficientemente sistemas de ecuaciones lineales en forma iterativa. El método puede ser aplicado a modelos provenientes de la ingeniería como circuitos eléctricos, reacciones químicas, o procesos de revestimiento. La idea principal es introducir un módulo de cálculo de autovalores, usando un algoritmo de tipo Arnoldi, en un método iterativo para resolver ecuaciones no lineales, a fin de obtener información sobre la estabilidad de la solución. Se presenta un ejemplo de uso de esta técnica con resultados preliminares de la aplicación del método a problemas tratados en la bibliografía del tópico. Los resultados muestran que la idea de calcular autovalores, en una iteración que resuelve ecuaciones no-lineales, tiene gran potencial en la detección de puntos críticos

**Palabras claves:** Continuación numérica, Sistemas no lineales, Métodos de Arnoldi, Análisis de estabilidad, Puntos críticos.

## Introducción

Un número significativo de problemas que surgen en las áreas científicas, tales como dinámica poblacional, teoría de combustión, dinámica y mecánica de fluidos, pueden ser modelados por sistemas de ecuaciones diferenciales parciales (PDE's), y luego discretizados en espacio, dando lugar a lo que se conoce como sistemas dinámicos no lineales, los cuales dependen de uno o más parámetros de interés.

Una manera de analizar estos fenómenos físicos, una vez que son modelados por PDE's es analizando las características de las soluciones de un sistema de ecuaciones no lineales de la forma  $G(x, \alpha) = 0$ , para un conjunto de valores de  $\alpha$  en un intervalo  $[a, b]$ . Esta técnica, conocida como Continuación Numérica, también es usada para detectar y calcular soluciones especiales que podrían señalar cambios significativos en el proceso de modelado, lo que conlleva a otro procedimiento conocido como análisis de bifurcación

Algunos cambios en estos sistemas son detectados cuando se pierde estabilidad en la continuación de las soluciones, o también cuando las soluciones comienzan o dejan de ser periódicas. En este caso, se requiere calcular con suficiente precisión estos puntos o soluciones especiales, ya que esta información puede ser usada para modificar los parámetros de operación con el fin de mejorar la productividad o la calidad del proceso que se esté modelando.

Los métodos de continuación más usados en la actualidad son aquellos basados en un esquema a dos pasos, del tipo predictor-corrector. En el primer paso (predictor) el método genera un nuevo valor para el parámetro  $\alpha$ , y utiliza la información de la solución anterior para obtener un estimado inicial de la solución para el nuevo valor de  $\alpha$ . En el segundo paso (corrector) esta aproximación inicial es tomada por un método iterativo para ecuaciones no

lineales, como el método de Newton, para hallar efectivamente la solución de  $G(x, \alpha) = 0$ . En las implementaciones actuales de estos métodos la estabilidad es analizada sólo en determinadas soluciones, y el cálculo de soluciones especiales, donde cambios importantes pueden producirse, es una tarea difícil porque la información requerida no está disponible. Para más detalles consultar Castillo (2004), Govaerts (2000) y Seydel (2009).

En este trabajo se muestra como diseñar e implementar un método de continuación numérica con un esquema predictor-corrector, que sea competitivo y que además de generar el diagrama de bifurcación, nos permita visualizar los puntos que representan soluciones estables o inestables o soluciones que señalen cambios de comportamiento del proceso modelado. Para ello, combinaremos el método de continuación con un método de proyección en espacios de Krylov del tipo Arnoldi, a fin de resolver el sistema lineal que aparece en cada iteración del paso corrector y al mismo tiempo obtener información sobre los autovalores y autovectores de la matriz Jacobiana asociada.

### Metodología

Se plantea en este trabajo una propuesta, con experimentación preliminar sobre el uso de la descomposición de Arnoldi dentro de un esquema de continuación, a fin de analizar la estabilidad de las soluciones que se calculan en cada iteración que resuelve el problema  $G(x, \alpha) = 0$ . Para ello, se siguen las siguientes etapas:

- 1) Análisis de los métodos de continuación y la forma general de sus implementaciones.
- 2) A partir de un algoritmo general para un método de continuación se estudiará la inserción del módulo para el cálculo de algunos autovalores de interés, de tal forma que no se degrade la eficiencia y que se contribuya no solo al cálculo de la próxima solución sino también a detectar pérdida de estabilidad en la solución, si la hubiera.
- 3) Diseño e implementación de la solución propuesta.
- 4) Pruebas experimentales en datos sintéticos.
- 5) Organización y divulgación de resultados.

### Continuación Numérica

Muchas aplicaciones industriales son modeladas matemáticamente por sistemas de ecuaciones en derivadas parciales o PDE's, y luego discretizadas en espacio para construir soluciones temporales. Los modelos más usados están basados en esquemas de diferencias finitas o elementos finitos, tal como explica Saad (2011). Cuando en el modelo existe un parámetro de importancia, cuya variación en un rango se desea analizar, esta discretización da origen a un sistema de ecuaciones no lineales parametrizado de la forma:

$$G(x, \alpha) = 0 \tag{1}$$

donde la función  $G$  depende de un vector  $x$  en  $\mathbb{R}^n$  y de un número real  $\alpha$ ; en aplicaciones reales este vector  $x$  depende de  $\alpha$ , como ya veremos más adelante. La solución  $(x, \alpha)$  de (1) es llamada punto estacionario o de equilibrio.

Una metodología muy utilizada para encontrar soluciones que satisfagan (1) variando el parámetro  $\alpha$  es conocida con el nombre de Continuación Numérica o Path Following (en inglés). Los métodos de continuación se caracterizan por hallar una solución de (1) para un valor particular de  $\alpha$ , usando la información de una solución conocida o punto previo. Este proceso se repite hasta hallar un conjunto de salidas que pueden ser usadas para construir una o varias ramas de soluciones.

El conjunto de soluciones o ramas obtenidas en un método de continuación, pueden ser graficadas en 2D o 3D para establecer la relación entre la variación del parámetro  $\alpha$  y el vector  $x$ , obteniendo así lo que se conoce como diagrama de bifurcación. En la práctica se suelen definir puntos de la forma  $(\alpha, [x])$  que luego se grafican en el plano para producir un diagrama de bifurcación; el valor real  $[x]$  es una medida escalar del vector  $x$  (la solución) y se puede tomar como la norma de  $x$ ,  $[x] = \|x\|$ , para alguna norma, o  $[x] = x_i$  para algún  $i = 1 \dots n$ , o como presentaremos en el ejemplo que desarrollamos más adelante, pudiera estar ligada a una medida de la longitud del arco que sigue la rama de soluciones.

El diagrama de bifurcación no es el gráfico de la solución de (1), pero proporciona una forma de analizar el comportamiento de las soluciones cuando se varía el parámetro  $\alpha$ . De esta manera podemos observar en el gráfico puntos de interés tales como:

- 1) Puntos donde dos o más ramas se intersecan BP (Branching Points).
- 2) Puntos donde la rama cambia de dirección TP (Turning Points).
- 3) Puntos que señalan el cambio a solución periódica HP (Hopfs Points).

La detención y cálculo eficiente de estos puntos es de vital importancia en aplicaciones industriales, una vez que permite cambiar o ajustar el esquema de operación. A continuación, detallamos el esquema de todo algoritmo de continuación.

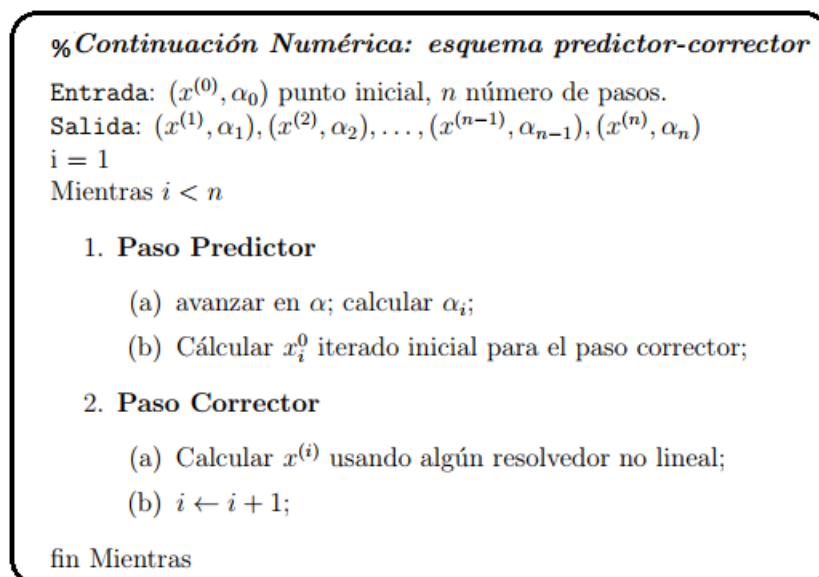
### Métodos de continuación

Dado un punto estacionario inicial  $x^{(0)}$  correspondiente a un valor del parámetro  $\alpha_0$ , un método de continuación calcula un camino de soluciones  $(x^{(1)}, \alpha_1), (x^{(2)}, \alpha_2), \dots, (x^{(k)}, \alpha_k)$ , de la ecuación (1), variando el parámetro  $\alpha$  y usando métodos iterativos tipo Newton para resolver sistemas de ecuaciones no lineales. Las técnicas más populares de continuación numérica están basadas en esquemas de tipo predictor-corrector que se presentan a continuación.

### Algoritmo Predictor-Corrector

La idea detrás del algoritmo predictor-corrector es seguir una curva de soluciones, resolviendo sistemas no lineales parametrizados de la forma dada en (1). Cada punto de la curva se obtiene en dos pasos; en el primer paso se genera una aproximación de la solución usando información previa (paso predictor), mientras que en un segundo paso se usa este predictor como una aproximación inicial para un método iterativo, como por ejemplo el método de Newton (paso corrector), el cual genera una nueva solución o punto estacionario.

Para entender este procedimiento, supongamos que ya tenemos una solución inicial  $(x^{(0)}, \alpha_0)$ , que satisface la ecuación (1) y que se desea encontrar un nuevo punto estacionario  $(x^{(1)}, \alpha_1)$ , para lo cual necesitamos una aproximación inicial que llamaremos  $(x_1^0, \alpha_1^0)$ , y una estrategia para calcular  $(x^{(1)}, \alpha_1)$ . Un esquema básico y general de un algoritmo de continuación se muestra en la Figura 1.



**Figura 1.** Algoritmo (1). Esquema Predictor-Corrector

**Fuente:** Algoritmo desarrollado en la investigación.

En el paso 1(a) de la  $i$ -ésima iteración del Algoritmo (1) se suele asignar  $\alpha_i = \alpha_{i-1} + \delta$  para un valor dado de  $\delta$ , luego en el paso 1(b) tomar  $x_i^0 = x^{i-1}$ . Este esquema es llamado continuación natural o continuación de orden cero, mientras que en los algoritmos de continuación de primer orden se asume que el vector  $x$  es una función que depende a su vez de  $\alpha$ , por lo tanto, en lugar de la ecuación (1), se considera la ecuación:

$$G(x(\alpha), \alpha) = 0 \tag{2}$$

En este caso, el predictor se calcula tomando un punto sobre la recta tangente a  $G$  en el punto anterior  $x^{i-1}$ , para lo cual se toma la derivada de la ecuación (2) con respecto a  $\alpha$  y se resuelve:

$$G_x(x(\alpha), \alpha)x_\alpha + G_\alpha(x(\alpha), \alpha) = 0 \quad (3)$$

Donde  $G_x$  es la matriz de derivadas respecto del vector  $x$ , mientras que  $G_\alpha$  es el vector de derivadas de  $G$  con respecto a  $\alpha$ , y  $x_\alpha$  es el vector tangente.

En ambos tipos de continuación (orden cero y orden uno) el corrector puede usar un método iterativo del tipo Newton o quasi-Newton. La Figura 2 muestra la diferencia entre el predictor de la continuación de orden cero y aquel que determina la continuación de orden uno. En la misma se observa que el predictor en la continuación de orden uno está más cerca de la nueva solución que el predictor de orden natural o cero, lo que sugiere que el método iterativo convergerá más rápidamente a la solución  $x^{(1)}$ .

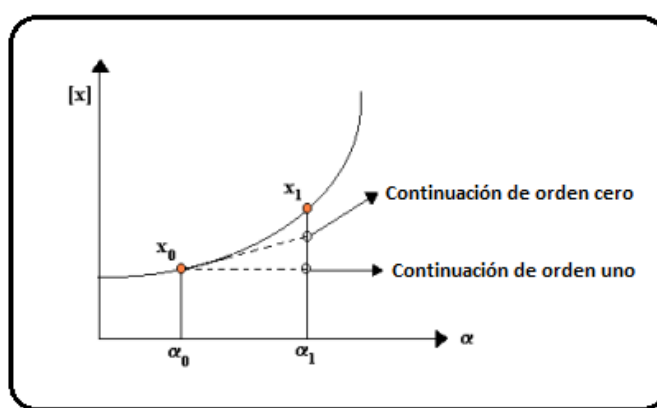


Figura 2. Predictor de orden cero y Predictor de orden 1

Fuente: Elaboración propia.

### Métodos de Krylov

En cada paso del método iterativo que resuelve el funcional no lineal se resuelve un sistema de ecuaciones lineales de la forma  $G_x(x, \alpha) = -G_\alpha(x, \alpha)$ , donde  $G_x$  es la matriz de derivadas o Jacobiana y  $G_\alpha$  el vector gradiente en el punto anterior. Esto permite introducir un método, basado en espacios de Krylov, que permita resolver este sistema, y al mismo tiempo nos de información sobre los autovalores de la matriz Jacobiana en la solución, lo cual constituye la idea principal de los métodos que se promueven en este trabajo.

Hallar un autovalor y/o un autovector de una matriz  $A$  significa calcular un vector  $v$  en  $\mathbb{R}^n$  y un número complejo  $\lambda$ , tales que  $Av = \lambda v$ . Al vector  $v$  se le denomina autovector y al escalar  $\lambda$  autovalor. En los métodos de continuación, la matriz  $A$  es la matriz Jacobiana  $G_x$  y por lo tanto nos referimos a  $\lambda$  como uno de los autovalores de  $G_x$ , el cual puede brindarnos información sobre la estabilidad del proceso asociado. En la práctica, lo ideal es tener información sobre algunos autovalores de interés de la matriz  $G_x$ , por ejemplo, los de mayor módulo o los de menor parte real. En particular, para el estudio de estabilidad interesa obtener los de mayor módulo.

Por otra parte, los problemas de interés en aplicaciones reales conllevan a sistemas lineales con matrices de gran magnitud, generalmente dispersas o ralas, lo que determina que métodos clásicos para matrices densas no son de utilidad. En las últimas décadas se han desarrollado métodos iterativos basados en espacios de Krylov que además de contribuir a la solución del sistema  $Ax = b$  también nos brindan información sobre los autovalores de la matriz de  $A$  de los coeficientes; y esta particularidad los hace ideales para su uso en los métodos de continuación, una vez que pueden ser usados para resolver simultáneamente los problemas  $Ax = b$  y  $Av = \lambda v$ .

Los métodos basados en Arnoldi se fundamentan en la descomposición Hessenberg de la matriz  $A$  o  $AV = VH$ , donde  $V$  es una matriz Ortogonal ( $V^t V = I$ ) y  $H$  una Hessenberg superior. Esta descomposición es ideal para la resolución simultánea de los problemas  $Ax = b$  y  $Av = \lambda v$ , ya que provee un mecanismo mediante el cual se trata un sistema de menor dimensión, además de tomar ventaja de realizar operaciones del tipo producto matriz-vector. Para una mejor comprensión de esta descomposición se sugiere ver Treffethen (1990) y Golub & Van Loan (1996).

Para obtener la descomposición de Hessenberg  $AV = VH$ , el método de Arnoldi construye  $V$  usando una base ortogonal del subespacio de Krylov:

$$K_m(A, v_1) \equiv \text{Span}\{v_1, Av_1, A^2v_1, A^3v_1, \dots, A^{2m-1}v_1\}$$

En este caso, dado un vector inicial  $v_1$  se añade un nuevo vector a la base en cada iteración, multiplicando el último vector por la matriz  $A$ . En la Figura 3 se visualiza un esquema general del algoritmo de Arnoldi.

```

% Arnoldi
Entrada:  $v_1$  un vector unitario.
Salida:  $(A, V_m, H_m, f_m)$  tal que  $AV_m = V_m H_m + f_m e_m^t$ 
 $m$  es el número de columnas de  $V$  y  $H$  que se quieren calcular.
 $V = [v_1]; H = [v_1^t A v_1];$ 
 $f = Av_1 - v_1 H;$ 
Para  $j = 1:m$  hacer
  1.  $\beta = \|f\|; v = \frac{f}{\beta};$ 
  2.  $V = [V, v];$ 
  3.  $H \leftarrow \begin{pmatrix} H \\ \beta e_j^t \end{pmatrix};$ 
  4.  $w = Av;$ 
  5.  $h = V^t Aw;$ 
  6.  $f = w - Vh$ 
  7.  $H = [H, h]$ 
fin

```

**Figura 3.** Algoritmo de Arnoldi

**Fuente:** Elaboración propia.

Los pasos del (4) al (6) del algoritmo (2) están basados en el proceso de Gram Schmidt clásico para ortogonalizar  $w = Av$  con respecto a las columnas de  $V$ . En la  $k$ -ésima iteración de este algoritmo se satisface la ecuación:

$$AV_k = V_k H_k + f_k e_k^t$$

llamada factorización de Arnoldi de orden  $k$ , donde  $V_k$  es una matriz de orden  $n \times k$ , y  $H_k$  es una matriz Hessenberg de orden  $k$ . Cuando la matriz  $A$  es simétrica, la matriz  $H_k$  se reduce a una matriz tridiagonal y entonces se habla del método de Lanczos. El vector  $f_k$ , y la matriz generada por el producto externo  $f_k e_k^t$  señalan el residuo de la descomposición. Si las componentes del vector  $f_k$  de la factorización de Arnoldi tienden a cero, entonces se tiene que  $AV_k \cong V_k H_k$ , y en consecuencia, podemos obtener soluciones aproximadas de los problemas  $Ax = b$  y  $Av = \lambda v$  resolviendo los problemas  $H_k \hat{x} = V_k b$  y  $H_k \hat{x} = \lambda \hat{x}$  respectivamente, donde  $x = V_k \hat{x}$  y  $v = V_k \hat{x}$  en cada problema. Detalles sobre esta descomposición puede encontrarse en Sorensen (1997) y Demmel (1997).

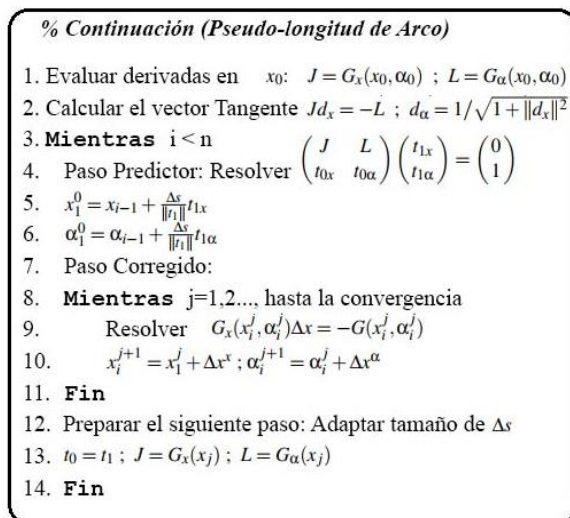
La precisión y la velocidad de convergencia de este tipo de método mejora con el incremento de  $k$ . Pero cuando  $k$  aumenta, también aumenta el número de vectores de Arnoldi y el tamaño de la matriz Hessenberg. Esto hace que el requerimiento de almacenaje y el costo de CPU incrementen, por lo que en la práctica se mantiene  $k$  en un valor bajo ( $k \ll n$ ) hasta lograr una precisión aceptable, o en algunos casos se produce un reinicio del proceso usando como vector inicial la última información obtenida. En Castillo (2004) se pueden encontrar ejemplos numéricos que soportan estas apreciaciones.

También se suelen hacer implementaciones en bloque del método de Arnoldi con un tamaño de bloque que permita agrupar autovalores, siempre evitando calcular la factorización completa, ya que en las aplicaciones de la continuación numérica basta con conocer unos cuantos autovalores para saber si estamos en presencia de un punto crítico o inestable, y lo que debemos asegurar es la obtención de una solución numéricamente estable del sistema de ecuaciones lineales que se resuelve en cada iteración del método de Newton. Ver por ejemplo los trabajos de Calvetti & Reichel (2000) y Baglama et al. (1998).

### **Experimentación y Resultados**

Con las ideas mencionadas previamente se diseñó y programó en Matlab un algoritmo basado en el método de Arnoldi y ajustado para la continuación numérica de un conjunto de soluciones provenientes del modelaje de una reacción química. La medida  $[x]$  utilizada para hacer el diagrama de bifurcación está ligada a una aproximación de la longitud de arco, frecuentemente llamada pseudo longitud de arco, de la curva de continuación. La figura 4 presenta el algoritmo utilizado en esta implementación.





**Figura 4.** Continuación con pseudo-longitud de arco  
**Fuente:** Algoritmo desarrollado en la investigación.

El algoritmo de la Figura 4 fue implementado en Matlab y comparado con CI-MatCont, que es un módulo (toolbox) de Matlab, ver Bindel et al. (2008), que resuelve problemas de continuación numérica. Se generaron los datos del problema de reacción química conocido como Brusselator, descrito en detalle por Prigogine y Lefever (1974), el cual ha sido usado para pruebas experimentales en estudios de continuación. La reacción química que da origen a este problema es oscilante con puntos de bifurcación.

Con la intención de comparar ambas implementaciones en cuanto a uso de memoria y tiempo de cómputo para hallar la solución, se consideraron mallas de discretización del problema seleccionado, que generaban matrices Jacobianas de tamaño variado, entre  $40 \times 40$  y  $100 \times 100$ , y se mide para cada uno de los problemas asociados, el tiempo en hallar la solución y la cantidad de memoria utilizada. El cuadro 1 muestra los primeros resultados de esta comparación; el tamaño del problema se denota con  $n$ , y las siglas *CM* y *CA* simbolizan Continuación con MatCont y Continuación con Arnoldi respectivamente. En el experimento se mide el tiempo que tarda el código en completar una rama de soluciones (CPU) y la memoria utilizada para almacenar soluciones intermedias y estructuras de datos (Memory).

n	CPU(CM)	CPU(CA)	Memory(CM)	Memory(CA)
40	9s	45.3s	72.696 KB	45.768 KB
50	12.7s	48.1s	93.200 KB	46.854 KB
60	13.8s	53.8s	143.122 KB	46.892 KB
70	15.6s	54.2s	251.280 KB	47.076 KB
80	20.1s	56.85s	367.106 KB	47.232 KB
90	24.9s	60.79s	562.174 KB	47.948 KB
100	N/A	66.38s	OUT OF MEMORY!!	48.015 KB

**Cuadro 1.** Comparación CI\_Matcom(CM) y Continuación con Arnoldi (CA)  
**Fuente:** Elaboración propia.

Aunque las mallas definidas para este problema pueden considerarse pequeñas, la generación de soluciones intermedias genera un consumo de memoria que llega a ser muy grande. Podemos observar en la tabla anterior que la implementación del algoritmo que usa un módulo de Arnoldi para resolver el sistema de ecuaciones, mantiene la utilización de memoria en un mismo rango, esto se logra con la característica de reinicio implícito en el cálculo de la base de Krylov (matriz  $V$ ); mientras que CI-MatCont incrementa el uso de la memoria a medida que el problema aumenta en dimensiones, llegando a consumir el máximo de memoria asignado al programa. Al mismo tiempo también notamos que la implementación con Arnoldi va incrementando el tiempo de CPU a medida que el tamaño del problema aumenta, pero no en forma desproporcionada.

Es importante señalar que, en términos de precisión, tanto CI-MatCont como la implementación PA presentada como ejemplo, satisfacen el objetivo de la continuación numérica, hallando diversas ramas de soluciones y detectando los puntos de interés, tal como se muestra en la Figura 5.

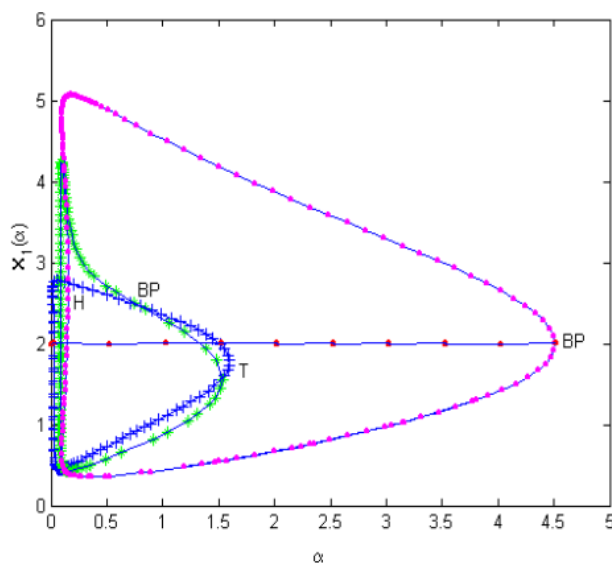


Figura 5. Continuación con pseudo-longitud de arco

Fuente: Elaboración propia.

Los resultados obtenidos solo nos permiten mostrar el uso de la herramienta, y no son concluyentes en términos generales; de hecho, otros autores como han logrado excelentes resultados en otras aplicaciones usando otros métodos iterativos para continuación, o diferentes métodos basados en espacios de Krylov, ver por ejemplo los trabajos de Baglama et al. (1998) y Calvetti & Reichel (2000).

## Resultados y Conclusiones

El análisis de bifurcación es una herramienta importante en muchas aplicaciones que modelan procesos industriales, y una de las técnicas más usadas para este análisis es la

continuación numérica, que consiste en seguir un conjunto de soluciones de una ecuación no lineal  $G(x, \alpha) = 0$  a medida que se varía uno o más parámetros de operación; y detectar soluciones críticas o puntos de interés. El comportamiento de estos puntos de interés usualmente señala situaciones que hay que tomar en cuenta para mejorar procesos y cambiar parámetros de funcionamiento. Estas características pueden descubrirse con el conocimiento de los autovalores de la matriz Jacobiana en estos puntos, pero generalmente los métodos de continuación hallan los puntos especiales, o críticos, y luego analizan sus características en una fase posterior, lo que acarrea más tiempo de cómputo y una posible pérdida de información. Este trabajo soporta la idea de usar un método para el cálculo de autovalores dentro del algoritmo de continuación, y específicamente en la resolución de los sistemas lineales que se resuelven en el paso corrector.

Hemos presentado un ejemplo usando el método de Arnoldi para el cálculo de autovalores y también para resolver estos sistemas lineales; sin embargo, pudiéramos usar algún otro método de Krylov, ya que estos métodos están en capacidad de resolver  $Ax = b$  y  $Ax = \lambda x$  simultáneamente y esa es la característica que los hace ideales en el ámbito de la continuación numérica.

El tamaño de la base de Krylov es un asunto que debe ser estudiado, ya que existe un compromiso entre la precisión de la solución y el uso de recursos computacionales como la memoria y el tiempo de procesamiento. Un algoritmo en bloques basado en espacios de Krylov sería una buena alternativa cuando se establece la característica que se desea evaluar de las soluciones; por ejemplo, si se desea evaluar la estabilidad, podría definirse un bloque de tamaño  $r$  y tratar de conseguir  $r$  autovalores de módulo máximo.

El presente documento estuvo dirigido a:

- 1) Presentar la generalidad de los métodos de continuación numérica, su utilidad e implementación básica en términos de un esquema predictor-corrector.
- 2) Presentar las principales características de los métodos de Krylov para el cálculo de algunos autovalores y autovectores de matrices de gran magnitud.
- 3) Señalar la utilidad de los métodos basados en espacios de Krylov, como los de Arnoldi, cuando se usan en conjunto con un método de continuación numérica.
- 4) Incluir algoritmos básicos que pueden usarse como guía para diseñar un método de continuación.
- 5) Establecer una comparación con un ejemplo práctico, para resaltar y soportar la propuesta de uso de métodos tipo Arnoldi en la continuación numérica de soluciones a problemas provenientes de aplicaciones en las ciencias e ingeniería.
- 6) Instruir en forma general con respecto a la algoritmia detrás de los métodos de continuación presentados.

## Referencias Bibliográficas

- Baglama, J., Calvetti, D., Reichel, L., & Ruttan, A. (1998). Computation of a few small eigenvalues of a large matrix with application to liquid crystal modeling. *Journal of Computational Physics*, 146(1), 203-226.
- Bindel, D., Friedman, M., Govaerts, W., Hughes, J., & Kuznetsov, Y. A. (2008). CL-MATCONTL: a continuation toolbox for large equilibrium problems in MATLAB.
- Calvetti, D., & Reichel, L. (1999). A block-Lanczos method for large continuation problems. *Numerical Algorithms*, 21(1-4), 109-118.
- Calvetti, D., & Reichel, L. (2000). Iterative methods for large continuation problems. *Journal of computational and applied mathematics*, 123(1-2), 217-240.
- Castillo, Z. (2004). A new algorithm for continuation and bifurcation analysis of large scale free surface flows. PhD thesis, Rice University, Houston, Texas.
- Demmel, J. W. (1997). *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics. Berkeley, California, first edition.
- Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computations*. 3rd. edn. ed. The Johns Hopkins University.
- Govaerts, W. J. (2000). *Numerical methods for bifurcations of dynamical equilibria*. Society for Industrial and Applied Mathematics.
- Prigogine, I., & Lefever, R. (1974). *Stability and self-organization in open systems* (pp. 215-240). Wiley, New York.
- Saad, Y. (2011). *Numerical methods for large eigenvalue problems: revised edition*. Society for Industrial and Applied Mathematics.
- Seydel, R. (2009). *Practical bifurcation and stability analysis* (Vol. 5). Springer Science & Business Media.
- Sorensen, D. C. (1997). Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations. In *Parallel Numerical Algorithms* (pp. 119-165). Springer, Dordrecht.
- Trefethen, L. N., & Bau III, D. (1997). *Numerical linear algebra* (Vol. 50). Siam.

**PARA CITAR EL ARTÍCULO INDEXADO.**

Castillo Marrero, Z. N., Colmenares Pacheco, G. A., & Cevallos Vique, V. O. (2020). Sobre el uso de métodos de Arnoldi para la continuación numérica de puntos estacionarios. *Ciencia Digital*, 4(3), 378-390. <https://doi.org/10.33262/cienciadigital.v4i3.1385>



El artículo que se publica es de exclusiva responsabilidad de los autores y no necesariamente reflejan el pensamiento de la **Revista Ciencia Digital**.

El artículo queda en propiedad de la revista y, por tanto, su publicación parcial y/o total en otro medio tiene que ser autorizado por el director de la **Revista Ciencia Digital**.

