REVISTA PROYECCIONES N° 9: 75-131 Julio 1985 - I.S.S.N. 0716-0917

RESOLUCIÓN NUMÉRICA DE SISTEMAS NO LINEALES MEDIANTE CURVAS DE HOMOTOPÍA

GABRIEL GATICA PEREZ*
EMILIANO HENRIQUEZ GODOY*

RESUMEN.

Se describe un método debido a KELLER [6], el que usa curvas de homotopía para hallar una o más raíces de un sistema de ecuaciones no lineales algebraico. Se especifica claramente el algoritmo respectivo y se introducen variantes en algunos de sus pasos. También se presenta una técnica alternativa basada en los fundamentos del método. Finalmente se muestran algunos ensayos numéricos que resaltan el carácter de convergencia global de los algoritmos. Se adjunta un apéndice con el programa computacional FORTRAN.

[★] DEPARTAMENTO DE MATEMATICA - UNIVERSIDAD DE CONCEPCION.

1. INTRODUCCION.

Hoy en día es muy importante poder disponer de métodos eficientes para resolver sistemas no lineales algebraicos. Ellos aparecen con frecuencia en muchos problemas de aplicación, como por ejemplo en la discretización de ecuaciones diferenciales y ecuaciones integrales no lineales. Puede mencionarse también el ajuste de mínimos cuadrados no lineal, en el cual la igualdad del gradiente (respecto a los parámetros), con el vector nulo constituye precisamente un sistema no lineal de ecuaciones.

Los métodos del tipo NEWTON-RAPHSON, BROWN, o BRENT (ref. [3], [4], [8]) son todos de convergencia local; es decir, su acercamiento a la solución está asegurado sólo si el punto inicial elegido está suficientemente cerca de ella. Sin embargo, ante un desconocimiento de las características del problema, el investigador debe tantear con cualquier punto, en cuyo caso, obviamente, tendrá mayor probabilidad de obtener una estimación inicial. En la mayoría de estas situaciones los algoritmos mencionados tienden a fallar.

Por otro lado, existen también algunos sistemas que poseen más de una solución, las cuales desean encontrarse en su totalidad. Ante estos casos, los métodos de convergencia local son igualmente incapaces de satisfacer esos requerimientos.

No obstante, las técnicas de seguimiento o de trazado de curvas de homotopía pretenden resolver las dificultades mencionadas anteriormente. Más específicamente, el método de KELLER descrito aquí, es capaz de hallar secuencialmente varias raíces de un sistema no lineal, y lo que es más importante, puede lograrlo a partir de malas estimaciones iniciales.

El artículo se presenta como sigue: en sección 2 se describe un método de KELLER [6], y se especifica su algoritmo con algunas variantes. Se presenta en sección 3, un método alternativo basado en los fundamentos dados por KELLER [6] y en una proposición original. Por último, en sección 4 se dan algunos ensayos numéricos que muestran la efectividad de las técnicas descritas.

2. METODO DE KELLER.

Dada f : $\mathbb{R}^N \to \mathbb{R}^N$ de clase C , interesa resolver el sistema no lineal

(2.1)
$$f(u) = 0$$

Para $u \in \mathbb{R}^N$, fijo, y $\alpha > 0$ se considera inicialmente la homotopía H dada por:

$$H : \mathbb{R}^{N} \times [0, +\infty) \to \mathbb{R}^{N}$$

$$(u,t) \longrightarrow H(u,t) = f(u) - e^{-\alpha t} f(u_{o})$$

Además, se define la curva de homotopía [asociada a H como:

$$\Gamma = H^{-1}(\{0\}) = \{(u,t) \in \mathbb{R}^N \times [0,\infty)/H(u,t) = 0\}$$

Entonces, si a través de Γ existe lím u(t) y vale u* \mathbb{R}^N , resulta:

$$\lim_{t\to +\infty} (f(u(t)) - e^{-\alpha t} f(u_0)) = 0$$

esto es:

$$f(u^*) = 0$$

Lo anterior indica que cuando $t \to +\infty$, la proyección de Γ sobre \mathbb{R}^N tiende a una solución de (2.1). Además, si se deriva implícitamente con respecto a t la ecuación H(u,t)=0, obtenemos:

(2.2)
$$f'(u) u'(t) + e^{-\alpha t} f(u) = 0 ; \forall (u,t) \in \Gamma$$

donde f'(u) indica Jacobiana de f.

Puesto que f(u) = $e^{-\alpha t}$ f(u_o) para cada (u,t) $\in \Gamma$, (2.2) puede escribirse:

(2.3)
$$f'(u)u'(t) + \alpha f(u) = 0 ; \forall (u,t) \in \Gamma.$$

Así la solución del sistema de ecuaciones diferenciales no $1\underline{i}$ neal (2.3) sujeto a la condición inicial $u(0) = u_0$, conduce la trayectoria de homotopía Γ desde $(u_0, 0)$ hasta $(: , +\infty)$, donde $u^* = 1$ ím u(t).

Observemos que si se usa Método de Euler sobre (2.3) para aproximar la curva Γ , entonces se obtiene la sucesión $\{u_m\}$ dada por:

(2.4)
$$u_{m} = u_{m-1} + \Delta t u'_{m-1}$$

es decir:

(2.5)
$$u_{m} = u_{m-1} + \Delta t \left(-\alpha [f'(u_{m-1})]^{-1} f(u_{m-1})\right)$$

o bien:

(2.6)
$$u_{m} = u_{m-1} - \alpha \Delta t \left[f'(u_{m-1}) \right]^{-1} f(u_{m-1}).$$

Ahora, si tomamos un paso uniforme $\Delta t = \frac{1}{x}$, nos queda:

(2.7)
$$u_{m} = u_{m-1} - [f'(u_{m-1})]^{-1} f(u_{m-1})$$

el cual es precisamente el **Método de Newton Raphson** para aproximar una raíz de (2.1) con punto inicial u .

Este solo hecho nos hace pensar que el método de Newton es altamente mejorable ya sea considerando para (2.3) un método superior al

de Euler, o bien definiendo previamente una homotopía H con mejores propiedades.

Un modo de hacer esto último es tomar $\alpha = \alpha(u)$ en (2.3). Por ejemplo, BRANIN [2] emplea $\alpha(u) = \text{sig}(\det f'(u))$ y SMALE [10] considera varias elecciones del tipo sig $(\alpha(u)) = \text{sig}(\det (f'(u)))$. Sin embargo, estas técnicas causan problemas en la implementación numérica de los métodos en el sentido que deben tomarse pasos muy pequeños en la veca dad de Jacobianos con determinante nulo.

pretende l'impresas dificultades usando una homotopía algo diferente.

2.1. METOTO DE KETTER: DESCRIPCION Y FUNDAMENTOS.

2.1.1. Homotopía y curva de Homotopía.

Sea $\mathbb{R} \stackrel{C}{=} \mathbb{R}^N$, abierto y acotado. Para u e $\partial \Omega$, fijo, se sugiere la homotopía:

$$G: \mathbb{R}^{\mathbb{N}} \times \mathbb{R} \to \mathbb{R}^{\mathbb{N}}$$

(2.8)
$$(u,\lambda) \rightarrow G(u,\lambda) = f(u) - \lambda f(u_0)$$

Además se define la curva de homotopía I como:

$$\Gamma = G^{-1}(\{0\}) = \{(u,\lambda) \in \mathbb{R}^N \times \mathbb{R}/G(u,\lambda) = 0\}$$

Nhora, a través de Γ se introduce el parámetro longitud de ar-

(2.10)
$$\Gamma(s) = \{(u(s), \lambda(s)) \in \mathbb{R}^N \times \mathbb{R}/\mathbb{G}(u(s), \lambda(s)) = 0\}$$

y obviamente se verifica:

(2.11)
$$||\dot{\mathbf{u}}(s)||^2 + |\dot{\lambda}(s)|^2 = 1; \ \forall (\mathbf{u}(s), \lambda(s)) \in \Gamma(s)$$

donde el punto indica derivada respecto a s.

Por otro lado, a partir de la identidad:

$$(2.12) \qquad G(u(s),\lambda(s)) = f(u(s)) - \lambda(s) f(u(s)) = 0; \quad \forall (u(s),\lambda(s)) \in \Gamma \quad (s)$$

se obtiene claramente la siguiente ecuación diferencial:

(2.13)
$$f'(u)\dot{u} - \dot{\lambda}f(u) = 0; \forall (u,) \in \Gamma (s).$$

Nota 2.1. Es importante resaltar aquí que si $\lambda(s^*)=0$, para alguna longitud s* sobre Γ , entonces $u(s^*)$ es una raíz de (2.1). Más aún, varias raíces pueden obtenerse si $\lambda(s)$ se anula más de una vez sobre Γ . Este hecho, bastante simple en apariencia, que es la clave de lo ideado por KELLER [6], nos confirma en gran medida lo adelantado en la introducción de este artículo.

2.1.2. Raíces sobre $\Gamma(s)$.

Sea el conjunto:

(2.14)
$$C = \{ (u,\lambda) \in \mathbb{R}^{N} \mathbb{R}/\text{rango} \quad \frac{\partial G(u,\lambda)}{\partial (u,\lambda)} < N \}$$

entonces los vectores de C se llaman puntos críticos de G. De la misma manera los elementos del cojunto:

(2.15)
$$G(C) = \{ Y \in \mathbb{R}^N | Y = G(u, \lambda) ; \forall (u, \lambda) \in C \}$$

se llaman valores críticos de G. Además, si $Z \in Rec(G)-G(C)$, entonces Z se llama valor regular de G.

Aquí conviene recordar de MILNOR [7], los siguientes resultados:

Proposición 2.1. (teorema de Sard)

Sea $g: U \to \mathbb{R}^N$ suave; $U \subseteq \mathbb{R}^m$, abierto.

Sea
$$C = \{x \in U / rango g'(x) < N\}$$

entonces $\mu(G(C)) = 0$. (μ : medida de Lebesgue)

Proposición 2.2.

Sean W, V variedades de dimensión m y n respectivamente. (m > n). Sea g: W \rightarrow V suave y consideremos z \in V un valor regular de g. Entonces $g^{-1}(\{z\})$ \subseteq W es una variedad suave de dimensión m-n.

Con la suavidad necesaria para f obtenemos G \in C²($\mathbb{R}^N \times \mathbb{R}$), y por lo tanto, el teorema de Sard nos asegura que la medida de Lebesgue de G(C) es nula.

Por otro lado, la importancia de la proposición 2.2. radica en el siguiente hecho: si $u_0 \in \partial \Omega$ se elige de modo que 0 es un valor regular de $G(u,\lambda)$, entonces la curva de homotopía $r=G^{-1}(\{0\})$ que nos interesa, es una C^1 - variedad de dimensión 1; es decir, Γ consiste de arcos suaves o curvas cerradas difeomorfas a un círculo. Además, una vez elegido $u_0 \in \partial \Omega$, la probabilidad de que 0 sea valor regular es bastante al ta puesto que los valores críticos de G constituyen un conjunto de medida nula.

Suponemos en lo que sigue que el conjunto abierto acotado Ω con la una trontera $\partial\Omega$ convexa y suave. Se supone también que f verifica $\partial\Omega$ las siguientes condiciones de frontera de SMALE:

(2.16) f'(u) es invertible sobre $\partial\Omega$.

Una de las siguientes se cumple:

(2.17)
$$(f'(u))^{-1} f(u) \in \Omega, \quad \forall u \in \partial\Omega$$

0

$$(2.18) \qquad (f'(u))^{-1} f(u) \in \Omega^{C} ; \forall u \in \partial\Omega.$$

De acuerdo a esto se enuncia a ontinuación el teorema más relevante para el objetivo del método:

Teorema 2.1.

Sea f : $\bar{\Omega}$ \rightarrow IR $^{\rm N}$ de clase C 2 tal que f cumple las condiciones de SMALE (2.16)-(2.18).

Sea $u_0 \in \partial\Omega$ tal que 0 es valor regular de $G(u, \lambda) = f(u) - \lambda f(u_0)$. Entonces: existe una curva $\Gamma(s)$ de clase C^1 , solución de (2.11), (2.13) sobre $s \in [0, \overline{s}]$, que comienza en $(u(0), \lambda(0)) = (u_0, 1)$, y que termina en $(u(s), \lambda(\overline{s}))$ donde:

$$\mathbf{u}\left(\mathbf{\tilde{s}}\right) \in \partial\Omega\;;\;\lambda\left(\mathbf{\tilde{s}}\right) < 0\;;\;\; \left|\;\;\lambda\left(\mathbf{\tilde{s}}\right)\;\right| < \frac{\max \quad \left\|\;\mathbf{f}\left(\mathbf{x}\right)\;\right\|}{\min \quad \left\|\;\mathbf{f}\left(\mathbf{x}\right)\;\right\|}$$

Demostración: Ver KELLER [6]: Teorema 2.4.

Como corolario inmediato del teorema 2.1 observemos que debido al hecho que $\lambda(0)=1>0$ y $\lambda(\bar{s})<0$, existe un número impar de puntos $s_j\in [0,\bar{s}]$ tales que $\lambda(s_j)=0$, y por consiguiente $f(u(s_j))=0$.

Es importante notar que la condición 0 valor regular de G implica que la matriz jacobiana $\frac{\partial G(u,\lambda)}{\partial (u,\lambda)}$ tiene rango N para todos los puntos (u,λ) $\in \Gamma$. En efecto, se tiene el siguiente resultado:

Proposición 2.3.

Sea f: $\bar{\Omega} \to \mathbb{R}^N$ de clase C^2 tal que f verifica las condiciones de SMALE (2.16)-(2.18). Sea $u_0 \in \partial \Omega$ tal que 0 es valor regular de G de finida por $G: \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$; $G(u,\lambda) = f(u) - \lambda f(u_0)$. Entonces existe una curva Γ de clase C^1 , solución de (2.11), (2.13), a través de la cual se cumple

(2.19) rango
$$\frac{\partial G(u,\lambda)}{\partial (u,\lambda)} = N; \quad \forall (u,\lambda) \in \Gamma.$$

Demostración: La primera parte es consecuencia inmediata del teorema 2.1.

Supongamos que existe $(\bar{u}, \bar{\lambda})$ e Γ tal que rango $\frac{\partial \ G(u, \lambda)}{\partial (u, \bar{\lambda})} / (\bar{u}, \bar{\lambda}) \leq N-1. \quad \text{Entonces } (\bar{u}, \bar{\lambda}) \text{ es un punto crítico de G.}$

Además, es claro que $G(\bar{u}, \bar{\lambda}) = 0$ (porque $(\bar{u}, \bar{\lambda}) \in \Gamma$). Sin embargo, esto contradice el hecho que 0 es valor regular de G.

Nota 2.2. Debido a la condición (2.19) de proposición 2.3, la curva Γ dada en el teorema 2.1 se llamará curva regular o trayectoria de homotopía regular. Sin embargo, no debe olvidarse que la conclusión de dicho teorema sólo ocurre si $\mathbf{u}_0 \in \partial \Omega$ es tal que 0 es valor regular de G. Por lo tanto, la pregunta obvia es: ¿Para qué valores $\mathbf{u}_0 \in \partial \Omega$ puede asegurar se, que 0 es valor regular de $G(\mathbf{u},\lambda) = f(\mathbf{u}) - \lambda f(\mathbf{u}_0)$?. La respuesta la da el siguiente resultado obtenido por PERCELL [9]:

Proposición 2.4.

Sea f : $\vec{\Omega}$ \rightarrow IR N de class \mathbb{C}^{2} tal que f cumple las condiciones de SMALE (2.16)-(2.18). Suponga además que:

- i) rango $f'(u) \ge N-1$, $\forall u \in \Omega \text{ con } f(u) = 0$.
- ii) f(u) = 0, para a lo más un número contable de elementos en 12.

Entonces, existe B \underline{c} $\partial\Omega$, con modida de B nula (relativa a $\partial\Omega$), tal que \forall u_{Ω} \in $(\partial\Omega$ -B), 0 es valor regula \exists e G.

Demostración: Ver PERCELL [9].

En resumen, el teorema 2.1. junto con la ayuda de la proposición 2.4 nos aseguran que la trayectoria de homotopía regular Γ definida en (2.9), conduce a al menos una raíz de (2.1), para casi todos los $\mathbf{u}_0\in\mathbb{N}$. Por lo tanto, el objetivo será avanzar a través de Γ hasta detectar un cambio de signo en λ , lo cual indicará la proximidad de una raíz de (2.1). La forma de hacer este recorrido es tratada en la siguiente sección.

2.1.3. Cálculo de Trayectorias.

Un procedimiento obvio para aproximar la curva l'es resolver me diante un método de integración numérica el sistema de ecuaciones diferenciales ordinarias no lineal (2.11), (2.13) con la condición inicial (u(3), $\lambda(0)$) = (u₀,1). Una primera dificultad con esto es que el sistema de ecuaciones no está en forma estándar. Sin embargo, puede demostrarse que $\hat{\bf u}(s)$ y $\hat{\lambda}(s)$ se despejan de (2.11), (2.13) en función sólo de $\bf u(s)$. Este es precisamente el objetivo de las siguientes dos proposiciones. Para ello supongamos primero que $\bf u_0$ e λn se ha elegido de modo que 0 es valor regular de G; así, las hipótesis de Teorema 2.1 ocurren.

Proposición 2.5.

Sea $\Gamma(s) = \{(u(s), \lambda(s)) \in \mathbb{R}^N \times \mathbb{R}/G(u(s), \lambda(s)) = 0\}$ la trayectoria regular de clase C^1 con tangente $(\mathring{u}(s), \mathring{\lambda}(s))$ definida por (2.11) y (2.13). Entonces, la matriz:

(2.20)
$$A(s) = \begin{bmatrix} f'(u(s)) & -f(u_0) \\ & & \\ 2 \dot{u}^{T}(s) & 2 \dot{\lambda}(s) \end{bmatrix}$$
 es invertible

para todo (u(s), λ (s)) $\in \Gamma$ (s).

Demostración:

Sea $(u(s), \lambda(s)) \in \Gamma(s)$ y notemos J(s) = f'(u(s)). La demostración se separa en dos casos según J(s) sea singular o no.

i) Supongamos que J(s) es invertible. Entonces, puesto que:

$$A(s) = \begin{bmatrix} J(s) & 0 \\ & & \\ 2 & \dot{u}^{T}(s) & 1 \end{bmatrix} \begin{bmatrix} I & -J(s) & f(u_0) \\ 0 & 2 & d(s) \end{bmatrix}$$

con $d(s) = \dot{\lambda}(s) + \dot{u}^{T}(s) J(s)$ $f(u_0)$, vemos que A(s) es no singular si y sólo si $d(s) \neq 0$.

Por otro lado, si $\dot{\lambda}(s) = 0$, entonces de (2.13) resulta f'(u(s)) $\dot{u}(s) = 0$, y como f'(u(s)) es invertible se obtiene $\dot{u}(s) = 0$. Así, la relación (2.11) no se satisface, lo cual es una contradicción. Luego, necesariamente debe tenerse $\dot{\lambda}(s) \neq 0$. Además, de (2.13) se despeja

$$J(s) f(u_0) = \frac{\dot{u}(s)}{\lambda(s)}$$

y de (2.11) se tiene:

$$\dot{u}^{T}(s) \dot{u}(s) = 1 - (\lambda(s))^{2}$$
.

Por lo tanto:

$$d(s) = \dot{\lambda}(s) + \dot{u}^{T}(s) \frac{\dot{u}(s)}{\dot{\lambda}(s)} = \dot{\lambda}(s) + \frac{(1-(\dot{\lambda}(s))^{2}}{\dot{\lambda}(s)} = \frac{1}{\dot{\lambda}(s)} \neq 0.$$

Esto prueba que A(s) es invertible.

ii) Supongamos ahora que J(s) es singular. Esto significa obviamente rango $J(s) \le N-1$. Puesto que

$$\frac{\partial G(u,\lambda)}{\partial (u,\lambda)} = [f'(u(s)), -f(u_0)] \text{ y dado que}$$

$$(u(s),\lambda(s))$$

rango $\frac{\partial G(u,\lambda)}{\partial (u,\lambda)} = N$, \forall $(u,\lambda) \in \Gamma$, se concluye que rango f'(u(s)) no

puede ser inferior a N-1; así rango J(s) = N-1.

Además, $f(u_0)$ no pertenece al recorrido de J(s). En efecto, si $f(u_0)$ estuviera en dicho recorrido entonces

$$\frac{\partial G(u,\lambda)}{\partial (u,\lambda)} = [J(s), -f(u_0)] \text{ tendría rango } \leq N-1, \text{ lo}$$

$$(u(s),\lambda(s))$$

cual contradice el hecho que $(u(s),\lambda(s)) \in \Gamma$.

Por otra parte, $\dot{\lambda}(s)$ no puede ser distinto de cero ya que en tal caso, de (2.13) resulta:

$$f(u_0) = J(s) \frac{\dot{u}(s)}{\dot{\lambda}(s)}$$
, lo que contradice el hecho que $f(u_0)$ ¢

Rec J(s). Ası, necesariamente $\lambda(s) = 0$, y de (2.13) se obtiene:

(2.21)
$$J(s) \dot{u}(s) = 0$$

Además, como rango J(s) - N-1, se deduce dimensión (Ker J(s))=1 De esto y de (2.21) se concluye que Ker J(s) es generado por $\dot{u}(s)$ Ahora, consideremos el sistema homogéneo

$$A(s) \begin{bmatrix} R \\ \xi \end{bmatrix} = 0, \text{ con } (R, \xi) \in \mathbb{R}^{N} \times \mathbb{R}$$

Esto es:

(2.22)
$$\begin{bmatrix} J(s) & -f(u_0) \\ 2 \dot{u}^T(s) & 0 \end{bmatrix} \begin{bmatrix} R \\ \xi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

o bien:

(2.23)
$$J(s) R - \xi f(u_0) = 0$$

(2.24)
$$2 \dot{u}^{T}(s) R = 0.$$

Recordando nuevamente que $f(u_0)$ \notin Rec J(s), se deduce de (2.23) que $\xi = 0$. Así queda J(s) R = 0, esto es $R \in \text{Ker } J(s)$. Luego, existe $K \in \mathbb{R}$ tal que $R = K \dot{u}(s)$. Reemplazando en (2.24) obtenemos $2 K \dot{u}^T(s) \dot{u}(s) = 0$. Como $\dot{u}(s) \neq 0$, se concluye K = 0 y por lo tanto R = 0. Esto prueba que la única solución del sistema homogéneo (2.22) es el vector nulo. En consecuencia, la matriz A(s) es no singular.

Nota 2.3. De la demostración de esta proposición se extrae que f'(u(s)) es singular para algún (u(s), λ (s)) e Γ ssi $\dot{\lambda}$ (s) = 0 y $\dot{\mathbf{u}}$ (s) es un $\ddot{\mathbf{v}}_{F}$ ciado al vp nulo de f'(u(s)). Además, la relación $\|\dot{\mathbf{u}}$ (s) $\|^2$ + $|\dot{\lambda}$ (s) $\|^2$ = impone sobre $\dot{\mathbf{u}}$ (s) la restricción adicional $\|\dot{\mathbf{u}}$ (s) $\|$ = 1.

La proposición 2.5 nos permite demostrar el siguiente resultado:

Proposición 2.6.

Sea $(u(s_0), \lambda(s_0))$ un punto de la trayectoria regular Γ . Entonces existen funciones R(u), $\xi(u)$ de clase C^1 tales que:

(2.25)
$$\dot{u}(s_0) = R(u(s_0))$$

$$\dot{\lambda}(s_0) = \xi(u(s)).$$

Demostración: Consideremos de (2.11) y (2.13) las funciones:

(2.26)
$$H_{1}(u,\dot{u},\dot{\lambda}) = f'(u)\dot{u} - \dot{\lambda} f(u_{0}) = 0$$

(2.27)
$$H_{2}(u,\dot{u},\dot{\lambda}) = ||\dot{u}||^{2} + |\dot{\lambda}|^{2} - 1 = 0$$

Entonces:

$$\frac{\partial (H_1, H_2)}{\partial (u, \lambda)} = \begin{bmatrix} f'(u) & -f(u_0) \\ \\ 2 \dot{u}^T & 2 \lambda \end{bmatrix}$$

En particular para $(u(s_0), \lambda(s_0)) \in \Gamma$ resulta

$$\frac{\partial (H_{1}, H_{2})}{\partial (u, \lambda)} = \begin{bmatrix} f'(u(s_{0})) & -f(u_{0}) \\ & & \\ (u(s_{0}), \lambda(s_{0})) \end{bmatrix} = A(s_{0})$$

$$2 \dot{u}^{T}(s_{0}) \qquad 2 \dot{\lambda}(s_{0}) \end{bmatrix}$$

Luego, aplicando proposición 2.5 se deduce que

$$\frac{\partial (H_1, H_2)}{\partial (u, \lambda)}$$
 es invertible.
$$(u(s_0), \lambda(s_0))$$

Por lo tanto, del Teorema de la función implícita se concluye que podemos resolver (2.26) y (2.27) para únicas funciones R(u), ξ (u) de clase c^1 tales que

$$\dot{\mathbf{u}}(\mathbf{s}_0) = \mathbf{R}(\mathbf{u}(\mathbf{s}_0))$$

$$\lambda(s_0) = \xi(u(s_0)).$$

Además de lo dicho en la proposición anterior, el Teorema de la función implícita asegura que existe δ > 0 tal que la solución de:

$$\dot{\mathbf{u}}(\mathbf{s}) = \mathbf{R}(\mathbf{u})$$

(2.28)

 $\lambda(s) = \xi(u)$ que pasa por $(u(s_0), \lambda(s_0))$, define para $|s-s_0| < \delta$, un arco de soluciones de (2.11), (2.13), la cual puede continuarse hasta que A(s) de (2.20) lleguz a ser no singular. Sin embargo, la proposición 2.5 nos asegura que esto no ocurrirá sobre una tra yectoria regular. Por lo tanto, un procedimiento numérico apropiado pue de usarse sobre (2.11), (2.13) para construir completamente la curva Γ .

Para empezar, la integración se despeja de (2.11) y (2.13) obteniéndose:

(2.29)
$$\dot{\lambda}(0) = \pm [1 + || [f'(u_0)]^{-1} f(u_0) ||^2]^{-1/2}$$

(2.30)
$$\dot{u}(0) = \dot{\lambda}(0) [f'(u_0)]^{-1} f(u_0).$$

El signo en (2.29) se elige de modo que $\dot{\mathbf{u}}(0)$ esté dirigido dentro de Ω desde $\mathbf{u}_0 \in \partial \Omega$. Para efectos prácticos, la elección del signo en (2.29) sólo indica en qué dirección se empezará a recorrer Γ para hallar raíces de f. Por lo tanto, desde un punto de vista computacional, es el usuario quien proporciona la dirección inicial.

Por otro lado, es importante comentar que frecuentemente los métodos numéricos estándar aplicados al sistema (2.11), (2.13), presentan algunas dificultades como errores de truncación y errores de redondeo.

El primer inconveniente puede salvarse si se toman pasos pequeños, pero esto acrecienta el segundo. Un modo simple de solucionar en parte estos problemas es usar (2.12) periódicamente para retornar a la trayectoria Γ y así no dejar que la integración numérica de (2.11), (2.13) nos lleve demasiado lejos de la curva.

En definitiva, puede afirmarse que esta técnica resulta muy efectiva para una gran clase de problemas. El procedimiento que sugiere KE-LLER [6], y que se describe a continuación, pretende ser aún más eficiente para una clase más amplia de funciones.

Específicamente, se supone conocido un punto (u_n, λ_n) sobre la trayectoria Γ y el correspondiente vector tangente (u_n, λ_n) . Estos vecto res deben satisfacer (2.11), (2.12) y (2.13). Entonces, el objetivo es buscar otro punto (v, ξ) sobre Γ resolviendo el siguiente sistema no lineal de orden N + 1:

(2.31)
$$G(v,\xi) = f(v) - \xi f(u_0) = 0$$

(2.32)
$$N(\mathbf{v},\xi;\sigma) = \dot{\mathbf{u}}_{N}^{T} (\mathbf{v}-\mathbf{u}_{n}) + \dot{\lambda}_{n}(\xi-\lambda_{n}) - \sigma = 0.$$

La siguiente proposición muestra que efectivamente v y ξ pueden despejarse de (2.31) y (2.32), para $|\sigma|$ suficientemente pequeño.

Proposición 2.7.

Sean $(u_n, \lambda_n) \in \Gamma$ y $(\dot{u}_n, \dot{\lambda}_n)$ su vector tangente tales que satisfa cen (2.11)-(2.13). Entonces existe $\delta > 0$ tal que v y ξ pueden despejarse de (2.31) y (2.32) en función de σ , para $|\sigma| < \delta$.

Demostración:

Notar que con $\sigma=0$, se obtiene de (2.31), (2.32) (v, ξ) = (u_n, λ_n) . Además, es claro que (v, ξ) \neq (u_n, λ_n) para $\sigma\neq 0$.

Por otro lado, la jacobiana de (2.31)-(2.32) está dada por:

$$\frac{\partial (G,N)}{\partial (v,\xi)} = \begin{bmatrix} f'(v) & -f(u_0) \\ \vdots \\ u_n & \lambda_n \end{bmatrix}$$

Luego:

$$\frac{\partial (G,N)}{\partial (v,\xi)} = \begin{bmatrix} f'(u_n) & -f(u_0) \\ & & \\ \vdots & & \\ \dot{u}_n^T & & \dot{\lambda}_n \end{bmatrix}$$

У

$$\left| \frac{\partial (G,N)}{\partial (v,\xi)} \right|_{(u_n,\lambda_n)} = \frac{1}{2} \left| \begin{array}{ccc} f'(u_n) & -f(u_0) \\ & & \\ 2 & \dot{u}_n^T & 2 & \dot{\lambda}n \end{array} \right| = \frac{1}{2} \left| A(s_n) \right|$$

donde hemos supuesto $(u_n, \lambda_n) = (u(s_n), \lambda(s_n))$.

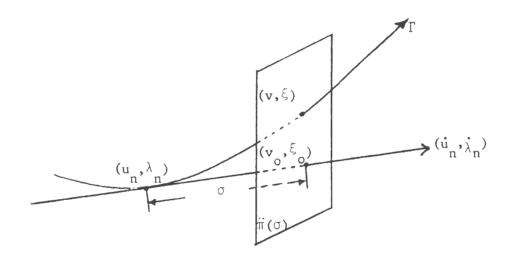
Así, aplicando proposición 2.5 se deduce que

$$\left|\frac{\partial (G,N)}{\partial (v,\xi)}\right| \ \text{es distinto de cero.}$$

$$(u_n,\lambda_n)$$

Por lo tanto, utilizando nuevamente el Teorema de la función implícita se concluye que existe $\delta > 0$ tal que con σ como parámetro, (2.31) y (2.32) definen un arco ($\mathbf{v}(\sigma)$, $\xi(\sigma)$) de Γ para $|\sigma| < \delta$. En otras palabras, existe una vecindad de ($\mathbf{u}_{\mathbf{n}}$, $\lambda_{\mathbf{n}}$) en la cual v y ξ de Γ pueden despejarse de (2.31) y (2.32) en función de σ .

En consecuencia, la regla consiste en generar una sucesión de arcos, como se indica en la proposición 2.7, cambiando cada vez el punto $(\mathbf{u}_n, \lambda_n)$ y el vector tangente asociado $(\dot{\mathbf{u}}_n, \lambda_n)$. Observemos también que (2.32) es la condición que (\mathbf{v}, ξ) permanezca en el plano $\pi(\sigma)$, normal al vector tangente $(\dot{\mathbf{u}}_n, \dot{\lambda}_n)$ y a una distancia $|\sigma|$ del punto de tangencia $(\mathbf{u}_n, \lambda_n)$. Además (2.31) establece que (\mathbf{v}, ξ) sea el punto de intersección de Γ con $\pi(\sigma)$. Una interpretación geométrica de esto se esquematiza en la siguiente figura:



Para calcular (v, ξ) de (2.31)-(2.32) usamos el método de NEWTON-RAPHSON con punto inicial:

(2.33)
$$\begin{cases} v_0 = u_n + \sigma \dot{u}_n \\ \xi_0 = \lambda_n + \sigma \dot{\lambda}_n \end{cases}$$

el que satisface claramente (2.32). La importancia de elegir (v_0,ξ_0) \in π (σ) radica en el hecho que como (2.32) es una restricción lineal, todas las demás iteraciones de Newton permanecen en $\pi(\sigma)$. A este respecto se dispone del siguiente resultado:

Proposición 2.8.

Sea $(v_m, \xi_m)_m$ e $\mathbb N$ la sucesión generada por Newton Raphson para el sistema (2.31)-(2.32) con punto inicial dado por (2.33). Entonces:

i)
$$N(v_m, \xi_m; \sigma) = 0, \forall m \in \mathbb{N}$$

ii)
$$v_{m+1} = v_m + (\Delta v)_m$$

(2.35)
$$\xi_{m+1} = \xi_{m} + (\Delta \xi)_{m}$$

donde:

$$\begin{bmatrix} f'(v_m) & -f(u_0) \\ \vdots \\ u_n^T & \lambda_n \end{bmatrix} \begin{bmatrix} (\Delta v)_m \\ (\Delta \xi)_m \end{bmatrix} = \begin{bmatrix} -G(v_m, \xi_m) \\ 0 \end{bmatrix}$$

Además, si $f'(v_m)$ es no singular, puede escribirse:

(2.36)
$$(\Delta \xi)_{m} = \frac{\dot{\mathbf{u}}_{n}^{T} \mathbf{z}}{\dot{\mathbf{u}}_{n}^{T} \mathbf{y} + \lambda_{n}}$$

$$(2.37) \qquad (\Delta V)_{m} = (\Delta \xi)_{m} Y - Z$$

con:

(2.38)
$$f'(v_m) Y = f(u_0)$$

(2.39)
$$f'(v_m) Z = G(v_m, \xi_m)$$

Demostración:

i) Por inducción sobre m. Para m= se tiene:

$$\begin{split} \mathbf{N}(\mathbf{v}_{0}, \boldsymbol{\xi}_{0}; \sigma) &= \dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}}(\mathbf{v}_{0} - \mathbf{u}_{\mathbf{n}}) + \dot{\lambda}_{\mathbf{n}}(\boldsymbol{\xi}_{0} - \boldsymbol{\lambda}_{\mathbf{n}}) - \sigma \\ &= \dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}}(\sigma \dot{\mathbf{u}}_{\mathbf{n}}) + \dot{\lambda}_{\mathbf{n}}(\sigma \dot{\lambda}_{\mathbf{n}}) - \sigma \\ &= \sigma \left[\dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}} \dot{\mathbf{u}}_{\mathbf{n}} + \dot{\lambda}_{\mathbf{n}}^{2} \right] - \sigma \\ &= \sigma \left[\left\| \dot{\mathbf{u}}_{\mathbf{n}} \right\|^{2} + \left| \dot{\lambda}_{\mathbf{n}} \right|^{2} \right] - \sigma = \sigma \cdot 1 - \sigma = 0. \end{split}$$

Supongamos que N(v_m, $\xi_{\rm m};\sigma$) = 0. El esquema de Newton-Raphson para (2.31)-(2.32) establece:

$$v_{m+1} = v_{m} + (\Delta v)_{m}$$

$$\xi_{m+1} = \xi_{m} + (\Delta \xi)_{m}, \text{ con:}$$

$$\begin{bmatrix} f'(v_m) & -f(u_0) \\ \dot{u}_n^T & \dot{\lambda}_n \end{bmatrix} \begin{bmatrix} (\Delta v)_m \\ (\Delta \xi)_m \end{bmatrix} = \begin{bmatrix} -G(v_m, \xi_m) \\ -N(v_m, \xi_m; \sigma] \end{bmatrix}$$

Puesto que $N(v_m, \xi_m; \sigma) = 0$, la última ecuación queda:

$$\dot{\mathbf{u}}_{n}^{T} \left(\Delta \mathbf{v}\right)_{m} + \dot{\lambda}_{n} \left(\Delta \xi\right)_{m} = 0$$

Luego:

$$N(v_{m+1}, \xi_{m+1}; \sigma) = \dot{u}_{n}^{T}(v_{m+1} - u_{n}) + \dot{\lambda}_{n}(\xi_{m+1} - \lambda_{n}) - \sigma$$

$$= \dot{u}_{n}^{T}(v_{m} + (\Delta v)_{m} - u_{n}) + \dot{\lambda}_{n}(\xi_{m} + (\Delta \xi)_{m} - \lambda_{n}) - \sigma$$

$$= \dot{u}_{n}^{T}(\Delta v)_{m} + \dot{\lambda}_{n}(\Delta \xi)_{m} + \dot{u}_{n}^{T}(v_{m} - u_{n}) + \dot{\lambda}_{n}(\xi_{m} - \lambda_{n}) - \sigma$$

$$= N(v_{m}, \xi_{m}; \sigma) = 0.$$

Esto completa la inducción y prueba i).

ii) El sistema lineal para $(\Delta \mathbf{v})_{m}$ y $(\Delta \xi)_{m}$ puede escribirse:

(2.40)
$$f'(v_m)(\Delta v)_m - f(u_0)(\Delta \xi)_m = -G(v_m, \xi_m)$$

(2.41)
$$\dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}}(\Delta \mathbf{v})_{\mathbf{m}} + \dot{\lambda}_{\mathbf{n}} (\Delta \xi)_{\mathbf{m}} = 0$$

Despejando de (2.40) queda:

$$(\Delta v)_{m} = (\Delta \xi)_{m} [f'(v_{m})]^{-1} f(u_{0}) - [f'(v_{m})]^{-1} G(v_{m}, \xi_{m})$$

esto es:

$$(v)_m = (\Delta \xi)_m Y - Z$$

Reemplazando en (2.41) obtenemos:

$$\dot{\mathbf{u}}_{n}^{T}[(\Delta \xi)_{m} \mathbf{Y} - \mathbf{Z}] + \dot{\lambda}_{n}(\Delta \xi)_{m} = 0$$

o bien:

$$(\Delta \xi)_{m} [\dot{\mathbf{u}}_{n}^{T} Y + \dot{\lambda}_{n}] = \dot{\mathbf{u}}_{n}^{T} Z$$

de donde:

$$(\Delta \xi)_{m} = \frac{\dot{\mathbf{u}}_{n}^{T} \mathbf{z}}{\dot{\mathbf{u}}_{n}^{T} \mathbf{y} + \dot{\lambda}_{n}}$$

lo que completa la demostración.

Nota 2.4. De gran relevancia es la distancia $|\sigma|$ la más grande para la cual (2.34)-(2.35) convergerá a un punto sobre Γ que verifique (2.31)-(2.32). Esto es importante, ya que mientras λ no sea cercano a cero estaremos interesados en avanzar rápido a través de la curva de homotopía. Al respecto, KELLER menciona en [6] una alternativa para σ , la cual requiere del cálculo del número de condición de la matriz Λ (s) definida en (2.20). Puesto que esto es muy caro, nosotros sugerimos en sección 2.3 un Esquema de selección automática del paso σ , el cual se basa en una idea de ALLGOWER [1].

Después que la raíz (v, ξ) de (2.31)-(2.32) es obtenida, se nece sita calcular ($\dot{\mathbf{v}}, \dot{\xi}$) para poder seguir el proceso empezando ahora desde (v, ξ). Para ello recordamos que (v, ξ) pertenece a Γ y por lo tanto debe verificar (2.11) y (2.13); esto es:

$$f'(v)\dot{v} - \dot{\xi} f(u_0) = 0$$

$$\|\dot{\mathbf{v}}\|^2 + |\dot{\xi}|^2 = 1$$

Estas ecuaciones se resuelven fácilmente para el caso f'(v) invertible formando:

(2.42)
$$f'(v) \phi = f(u_0)$$

(2.43)
$$\dot{\xi} = \pm \left[1 + \| \phi \|^2 \right]^{-\frac{1}{2}}$$

$$(2.44) \dot{\mathbf{v}} = \dot{\boldsymbol{\xi}} \ \boldsymbol{\phi}$$

Ahora, el vector $(\dot{\mathbf{v}}, \dot{\boldsymbol{\xi}})$ debe verificar:

$$\langle (\dot{\mathbf{u}}_{\mathbf{n}}, \lambda_{\mathbf{n}}), (\dot{\mathbf{v}}, \xi) \rangle = \dot{\mathbf{u}}_{\mathbf{n}} \dot{\mathbf{v}} + \lambda_{\mathbf{n}} \xi > 0$$

ya que con ello se asegura un avance positivo en la curva de homotopía. Es decir, el ángulo entre los vectores tangente debe variar entre 0 y $\pi/2$. De acuerdo a esto, el signo en (2.43) se elige de modo que (2.45) se satisfaga; esto es:

$$(2.46) \qquad \dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}} \dot{\mathbf{v}} + \dot{\lambda}_{\mathbf{n}} \dot{\xi} = \dot{\xi} (\dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}} \phi + \dot{\lambda}_{\mathbf{n}}^{\mathbf{T}}) > 0.$$

Notemos que la última interacción descrita en proposición 2 implica que Y de (2.38) coincide con ϕ de (2.42); luego, no se requieren cálculos adicionales para obtener ϕ . Además, la expresión $\dot{u}_n^T \phi + \dot{\lambda}_n = \dot{u}_n^T Y + \dot{\lambda}_n$ utilizada en (2.46) se obtiene también de la última iteración en el denominador de (2.36).

Por otro lado, si f'(v) es singular entonces como (v, ξ) e f recurrimos a la demostración de proposición 2.5 o a Nota 2.3 para recordar que en tal caso resulta $\dot{\xi}$ = 0 y $\dot{\mathbf{v}}$ es el $\dot{\mathbf{v}}$ p de f'(v) asociado al vp o que tiene norma 1. Además, el signo de $\dot{\mathbf{v}}$ se elige de modo que (2.45) se satisfaga, esto es: $\dot{\mathbf{u}}_n^T \dot{\mathbf{v}} > 0$.

El comentario anterior visualiza una cualidad más del método de Keller en el sentido que jacobianas singulares no afectan la aplicación del algoritmo; en otras palabras, la curva Γ puede pasar sin problemas a través de regiones con jacobiano de f nulo. Por lo tanto, al contrario del método de Newton, el algoritmo de Keller sí puede hallar raíces u* de f con |f'(u*)| = 0.

Finalmente, después que se obtienen (v,ξ) y $(\dot{v},\dot{\xi})$, se usan para reemplazar (u_n,λ_n) y $(\dot{u}_n,\dot{\lambda}_n)$ en (2.31)-(2.32), y se procede a calcular un nuevo punto de Γ . El proceso se sigue hasta detectar un cambio de signo en λ . Una vez localizada la raíz, KELLER [6] sugiere hallarla exactamente utilizando un esquema de tipo método de la secante. Nosotros proponemos en sección 2.4 un esquema alternativo bastanta simple que se basa sólo en una disminución ponderada del paso σ .

2.2. ALGORITMO.

El objetivo de esta parte es detallar cada uno de los pasos que deben seguirse en la aplicación del método de Keller descrito en sección 2.1.

Sea f: $\mathbb{R}^N \to \mathbb{R}^N$ de clase C^2 . Se desea resolver el sistema no lineal f(u) = 0. Para ellos se considera un vector inicial $\bar{u}_0 \in \mathbb{R}^N$ y se define la homotopía:

$$G(u,\lambda) = f(u) - \lambda f(\bar{u}_0)$$
.

En el esquema de Newton Raphson de proposición 2.8 consideramos una precisión ε que debe alcanzarse en un número máximo de iteraciones dado por M. Además, el paso σ se considera fijo a través de todo el proceso. Sin embargo, si se excede el número de iteraciones M en (2.34)- (2.39) y no se satisface el requerimiento de error ε , entonces se disminuye el paso σ en un 50% hasta una cierta tolerancia σ_{\min} , y se reinicia el proceso iterativo.

También se introduce un parámetro P que indica el número máximo de puntos a determinar sobre la curva Γ hasta localizar una raíz de f.

El algoritmo es el siguiente:

Paso 0. Definir
$$\bar{u}_0 \in \mathbb{R}^N$$
; $e > 0$; Me IN; Pe IN; $\sigma > 0$; $\sigma_{\min} > 0$

Definir una dirección de búsqueda SIG $e \{-1,1\}$.

Definir la función $G(u,\lambda) = f(u) - \lambda f(\bar{u}_0)$.

Paso 1. Definir
$$(u_0, \lambda_0) = (\bar{u}_0, 1)$$
.

Calcular z_0 del sistema lineal $f'(u_0)$ $z_0 = f(\bar{u}_0)$.

Hacer:
$$\dot{\lambda}_0 = \text{SIG} \left[1 + || z_0 ||^2 \right]^{-1/2}$$

$$\dot{u}_0 = \dot{\lambda}_0 z_0$$

Definir n = 0.

Paso 2.

- 2.1) Hacer $\sigma = \sigma$
- 2.2) Calcular $(v_0, \xi_0) = (u_n, \lambda_n) + \hat{\sigma} (\dot{u}_n, \dot{\lambda}_n)$.

 Hacer m = 0.

- Si
$$T_{m+1} \ge \varepsilon$$
, $m+1 = M$ y $\hat{\sigma} > \sigma_{min}$, hacer $\hat{\sigma} = \hat{\sigma}/2$ e ir a PASO 2.2.

- Si
$$T_{m+1} \ge e$$
, $m+1 = M$ y $\sigma \le \sigma_{min}$, ir a PASO 8.

Paso 3.

Definir
$$(u_{n+1}, \lambda_{n+1}) = (v_{m+1}, \xi_{m+1})$$

- Si f'(v_m) es no singular ir a PASO 4.
- Si f'(\mathbf{v}_{m}) es singular, hacer $\dot{\lambda}_{\mathrm{n}+1}$ = 0 y calcular $\dot{\mathbf{u}}_{\mathrm{n}+1}$ como el único $\dot{\mathbf{v}}_{\mathrm{p}}$ de f'(\mathbf{v}_{m}) asociado al \mathbf{v}_{p} 0, que tiene norma 1 y que verifica $\dot{\mathbf{u}}_{\mathrm{n}}^{\mathrm{T}}$ $\dot{\mathbf{u}}_{\mathrm{n}+1}$ > 0.

Paso 4.

Recuperar de PASO 2.4 el vector $\mathbf{Y}_{\mathbf{m}}$ y el escalar $\mathbf{u}_{\mathbf{n}}^{\mathbf{T}}$ $\mathbf{Y}_{\mathbf{m}}$ + $\lambda_{\mathbf{n}}$.

- Si
$$\dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}} \mathbf{Y}_{\mathbf{m}} + \dot{\lambda}_{\mathbf{n}} \geq 0$$
 entonces calcular $\dot{\lambda}_{\mathbf{n}+1} = [1+||\mathbf{Y}_{\mathbf{m}}||^2]^{-1/2}$.

- Si
$$\dot{\mathbf{u}}_{\mathbf{n}}^{\mathbf{T}} \mathbf{Y}_{\mathbf{m}} + \dot{\lambda}_{\mathbf{n}} < 0$$
 entonces calcular $\dot{\lambda}_{\mathbf{n}+1} = -\left[1 + \left\| \mathbf{Y}_{\mathbf{m}} \right\|^{2}\right]^{-1/2}$.

Calcular
$$\dot{u}_{n+1} = \dot{\lambda}_{n+1} Y_{m} e$$
 ir a PASO 5.

Paso 5.

- Si
$$\lambda_n$$
 $\lambda_{n+1} \leq 0$, ir a PASO 7

- Si
$$\lambda_n$$
 $\lambda_{n+1} > 0$, ir a PASO 6

Paso 6.

- Si $n + 1 \le P$, hacer n = n + 1 e ir a PASO 2.

- Si n + 1 = P, ir a PASO 9.

Paso 7.

En el segmento de la curva de homotopía que une los vectores (u_n, λ_n) con (u_{n+1}, λ_{n+1}) existe una raíz de f. Una vez que se ha calculado dicha raíz, existen dos posibilidades: terminar el proceso, o bien reiniciar la búsqueda de una nueva raíz. En este último caso se define:

$$(u_0, \lambda_0) = (u_{n+1}, \lambda_{n+1})$$

 $(\dot{u}_0,\dot{\lambda}_0)=(\dot{u}_{n+1},\dot{\lambda}_{n+1})$, se hace n=0 y se va nuevamente a PASO 2.

Paso 8.

A partir del vector (u_n, λ_n) y con un paso $\sigma \in [\sigma_{\min}, \sigma]$, no ha sido posible hallar un punto (v, ξ) en Γ con una precisión ϵ . El método falla y luego se detiene el proceso.

Paso 9.

Se han localizado P nuevos puntos sobre Γ y no se ha encontrado en ellos ningún cambio de signo en λ . Existen aquí dos posibilidades: terminar sin éxito el proceso o bien incrementar P, hacer n = n + 1 e ir a PASO 2.

Nota 2.5. A modo de observación podemos comentar que en todos los ejemplos numéricos que realizamos, algunos de los cuales se detallan es sección 4, la elección $\varepsilon = 10^{-6}$ y σ ε [0.1,1] tuvo un buen comportamiento. Más aún, en la mayoría de los casos, el número de iteraciones de Newton-Raphson para resolver (2.31)-(2.32) fue inferior o igual a 3.

2.3. SELECCION AUTOMATICA DEL TAMAÑO DEL PASO.

El proceso descrito en PASO 2.6 del algoritmo anterior constituye, en realidad, una especie de selección automática para o. Sin embargo, cada modificación del paso se realiza una vez que se han hecho las Miteraciones permitidas. Además, ello sólo contempla una disminución en su valor. Nosotros proponemos aquí un esquema similar al de ALLGOWER [1], el cual requiere sólo dos iteraciones para detectar si es necesaria una modificación en el tamaño del paso. También permite no sólo una disminución, sino que eventualmente, un incremento en el valor de o. Esto último se justifica, por ejemplo, cuando el vector tangente en un punto de la curva coincide o está muy cerca de dicha trayectoria en un tramo relativamente grande.

Sea σ > 0 un paso inicial y ϵ la tolerancia requerida para puntos de Γ . Consideremos 0 < α_1 < α_2 < 1 dos factores de contracción y σ_{\min} , σ_{\max} los tamaños de paso mínimo y permitidos, respectivamente.

Entonces, el proceso de selección automática de J está implícito en el reemplazo del PASO 2 del algoritmo de sección 2.2 por el siguiente esquema:

Paso 2.

2.1) Hacer $\sigma = \sigma$.

- 2.2) Calcular $(v_0, \xi_0) = (u_n, \lambda_n) + \hat{\sigma}(\dot{u}_n, \dot{\lambda}_n)$. Hacer m = 0. Ir a PASO 2.3.
- 2.3) Realizar PASO 2.4 dos veces hasta obtener (v_1, ξ_1) y (v_2, ξ_2) . Ir a PASO 2.5.
- 2.4)
 - 2.4.1) Si $f'(v_m)$ es no singular ir a PASO 2.4.2.

Si f'(v_m) es singular calcular (Δv) $_m$ y ($\Delta \xi$) $_m$ del sistema lineal siguiente:

$$\begin{bmatrix} f'(v_m) & -f(\bar{u}_0) \\ \vdots \\ u_n^T & \lambda_n \end{bmatrix} \begin{bmatrix} (\Delta v)_m \\ (\Delta \xi)_m \end{bmatrix} = \begin{bmatrix} -G(v_m, \xi_m) \\ 0 \end{bmatrix}$$

Ir a PASO 2.4.3.

2.4.2) Resolver los sistemas lineales:

$$f'(v_m) Y_m = f(\bar{u}_0)$$

$$f'(v_m) Z_m = G(v_m, \xi_m)$$

Calcular:

$$(\Delta \xi)_{m} = \frac{\dot{\mathbf{u}}_{n}^{T} \mathbf{z}_{m}}{\dot{\mathbf{u}}_{n}^{T} \mathbf{y}_{m} + \dot{\lambda}_{n}}; \quad (\Delta \mathbf{v})_{m} = (\Delta \xi)_{m}^{Y} \mathbf{y}_{m} - \mathbf{z}_{m}$$

2.4.3) Hacer:

$$v_{m+1} = v_m + (\Delta v)_m$$

$$\xi_{m+1} = \xi_m + (\Delta \xi)_m$$
.

2.5) Calcular:
$$w_1 = \| (v_1, \xi_1) - (v_0, \xi_0) \|$$

$$w_2 = \| (v_2, \xi_2) - (v_1, \xi_1) \|.$$

Ir a PASO 2.6.

- 2.6) Si $w_2 > \alpha_2$ w_1 y $\hat{\sigma} > \sigma_{min}$, hacer $\hat{\sigma} = \hat{\sigma}/2$ e ir a PASO 2.2.
 - $\sin w_2 > \alpha_2 w_1 y \sigma \leq \sigma_{min}$ ir a PASO 8.
 - Si $w_2 < \alpha_1 w_1 y \hat{\sigma} < \alpha_{max}$, hacer $\sigma = Min\{2\hat{\sigma}, \sigma_{max}\}$ e ir a PASO 2.2.
 - Si $w_2 < \alpha_1 w_1 y \hat{\sigma} \geqslant \sigma_{max}$, o bien, si $\alpha_1 w_1 < w_2 < \alpha_2 w_1$, ir a PASO 2.7.
- 2.7)
 - 2.7.1) Hacer m = 2
 - 2.7.2) Realizar PASO 2.4 y volver.
 - 2.7.3) Calcular $T_{m+1} = \|G(v_{m+1}, \xi_{m+1})\|_{\infty}$
 - 2.7.4) -Si $T_{m+1} < \epsilon$ ir a PASO 3.
 - -Si $T_{m+1} \ge \epsilon$, hacer m=m+1 e ir a PASO 2.7.2.

Nota 2.6. Conviene hacer notar que las modificaciones explicadas en PASO 2.6 sólo se permiten en un sentido al realizarse más de una vez; esto es, si la primera alteración sobre $\hat{\sigma}$ fue una bisección, entonces las restan-

tes, si las hay, también serán disminuciones en el tamaño del paso (De manera análoga si inicialmente se efectuó un aumento del valor). Obviamente se impone esta restricción con el objeto de evitar loops en la selección automática de σ .

2.4. LOCALIZACION EXACTA DE RAICES DE f SOBRE Γ .

Como ya se ha explicado en sección 2.1, el método de Keller genera una sucesión de puntos $(u_n, \lambda_n)_{n \in \mathbb{N}}$ sobre con sus correspondientes vectores tangente $(\dot{u}_n, \dot{\lambda}_n)$. Cuando λ cambia de signo entre dos vectores consecutivos (u_n, λ_n) y (u_{n+1}, λ_{n+1}) , entonces una raíz de f(u) = 0 permanece sobre el arco de Γ entre estos dos puntos. Una primera alternativa para hallar esta raíz es obviamente utilizar un método de tipo convergencia local, por ejemplo Newton-Raphson, con el punto inicial $\frac{u_n + u_n + 1}{2}$. Sin embargo, nada permite asegurar a priori que es te proceso convergerá, más aún si el jacobiano de f es cercano a cero en esa región.

Ahora, si (u_{n+1}, λ_{n+1}) se calculó a partir de (u_n, λ_n) con un paso σ entonces el arco $(v_n, \xi(\sigma))$ definido por las ecuaciones:

(2.47)
$$G(v,\xi) = f(v) - \xi f(\overline{u}_0) = 0$$

(2.48)
$$N(v,\xi;\sigma) = \dot{u}_n^T (v-u_n) + \dot{\lambda}_n(\xi-\lambda_n) - \sigma = 0$$

con $\in [0,\sigma]$, contiene la raíz u* de f. Por lo tanto, necesitamos sólo hallar $\sigma^* \in [0,\sigma]$ tal que $\xi(\sigma^*) = 0$ y así u* = $v(\sigma^*)$.

Para ello presentamos a continuación una técnica alternativa que generaliza el método de la secante sugerido por KELLER [6].

Sean (u_n, λ_n) y $(u_n + 1, \lambda_n + 1)$ los puntos de Γ entregados en el PASO 7 del algoritmo. Suponga que \hat{O} es el paso requerido para hallar el

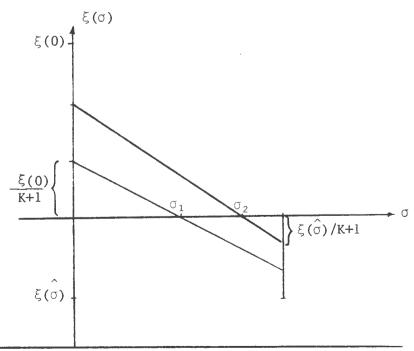
segundo vector a partir del primero. Luego, $\lambda_n \lambda_{n+1} \leq 0$ y para el arco definido en (2.47) y (2.48) se tiene:

$$(v(0), \xi(0)) = (u_n, \lambda_n)$$

 $(v(0), \xi(0)) = (u_n + 1, \lambda_n + 1).$

Con el objeto de localizar más exactamente la raíz se determinan K nuevos tamaños de paso σ_j , $j=\overline{1,K}$, sobre $[0,\sigma]$, y luego se empiezan a calcular los vectores $(\mathbf{v}(\sigma_j),\ \xi(\sigma_j))$ desde j=1 en adelante hasta detectar el primer cambio de signo. El proceso se reinicia con los nuevos vectores que presentan esta característica, hasta satisfacer un requerimiento de error del tipo: $\max\{|\xi(0)|\ ;\ |\xi(\sigma)|\}\ |\|\mathbf{f}(\overline{\mathbf{u}_0})\|_{\infty}<\epsilon.$

Una primera posibilidad para los valores σ_j es una distribución uniforme sobre $[0,\hat{\sigma}]$, esto es, $\sigma_j=j\frac{\hat{\sigma}}{K+1}$. Sin embargo, ello no considera la información proporcionada por los valores $\xi(0)$ y $\xi(\hat{\sigma})$. A objeto de tomar en cuenta esto último, sugerimos elegir σ_j como la raíz de la recta que une $(0,\frac{j\xi(0)}{K+1})$ con $(\hat{\sigma},(K+1-j)\frac{\xi(\hat{\sigma})}{K+1})$. Un ejemplo gráfico de esto, para K=2, se aprecia en la siguiente figura:



Un cálculo fácil revela que:

(2.49)
$$\sigma_{j} = \frac{j \; \xi(0) \; \hat{\sigma}}{j \; \xi(0) \; - \; (K + 1 - j) \; \xi(\hat{\sigma})} \; ; \; j = \overline{1, K}.$$

De aquí se observa que si $\left|\xi\left(0\right)\right|=\left|\xi\left(\sigma\right)\right|$ entonces se obtiene la distribución uniforme para σ_{j} . Además, el caso K = 1 no es más que el método de la secante.

De este modo el algoritmo de localización es el siguiente:

- Paso 0. Defina K \in IN y una tolerancia ε > 0. Indique el valor $\overset{\wedge}{\sigma}$
- Paso 1. Defina $(\mathbf{v}(0), \xi(0)) = (\mathbf{u}_n, \lambda_n)$ $(\mathbf{v}(0), \xi(0)) = (\mathbf{u}_n, \lambda_n)$ $(\dot{\mathbf{v}}(0), \dot{\xi}(0)) = (\dot{\mathbf{u}}_n, \dot{\lambda}_n).$
- Paso 2. Defina $\sigma_0 = 0$ y $\kappa + 1 = .$

Calcule
$$\sigma_{j} = \frac{j\xi(0)\hat{\sigma}}{j\xi(0) - (K+1-j)\xi(\hat{\sigma})}$$
; $j = \overline{1,K}$

Haga j = 1.

Paso 3.

3.1) Encuentre el vector $(v(\sigma_j),\xi(\sigma_j))$ solución del siguiente sistema no lineal en las incógnitas (v,ξ) .

$$G(v,\xi) = f(v) - \xi f(u_0) = 0$$

$$N(v,\xi;\sigma_j) = \dot{v}(0)(v - v(0)) + \dot{\xi}(0)(\xi - \xi(0)) - \sigma_j = 0.$$

Utilice Newton-Raphson con punto inicial dado por

$$(v(0), \xi(0)) + \sigma_{j}(\dot{v}(0), \dot{\xi}(0)).$$
3.2) - Si $\xi(\sigma_{j-1}) \xi(\sigma_{j}) > 0$ y j < K, haga j = j + 1 y vaya a PASO 3.1.

- Si $\xi(\sigma_j - 1)$ $\xi(\sigma_j)$ > 0 y j = K, haga j = k + 1 y vaya a PASO 5.

- Si $\xi(\sigma_{j-1})$ $\xi(\sigma_{j}) \leq 0$, vaya a PASO 4.

Paso 4. La raíz se ubica en el arco de Γ entre los vectores $(\mathbf{v}(\sigma_{j-1}), \xi(\sigma_{j-1}))$ y $(\mathbf{v}(\sigma_{j}), \xi(\sigma_{j}))$.

 $\tilde{\text{Defina T}}_{j} = \text{Max } \{ |\xi(\sigma_{j-1})| ; |\xi(\sigma_{j})| \} || f(\tilde{u}_{0})||_{\infty}$

Si $T_{j} < \epsilon$, terminar

Si $T_i \ge \epsilon$, vaya a PASO 5.

Paso 5. Defina:

$$(v(0), \xi(0)) = (v(\sigma_{j-1}), \xi(\sigma_{j-1}))$$

$$(\dot{\mathbf{v}}(0), \dot{\xi}(0)) = (\dot{\mathbf{v}}(\sigma_{j-1}), \dot{\xi}(\sigma_{j-1}))$$

Calcule:

$$\hat{\boldsymbol{\sigma}} = \dot{\boldsymbol{v}}(\boldsymbol{\sigma}_{j-1}) \, (\boldsymbol{v}(\boldsymbol{\sigma}_{j}) - \boldsymbol{v}(\boldsymbol{\sigma}_{j-1})) + \dot{\boldsymbol{\xi}}(\boldsymbol{\sigma}_{j-1}) \, (\boldsymbol{\xi}(\boldsymbol{\sigma}_{j}) - \boldsymbol{\xi}(\boldsymbol{\sigma}_{j-1})).$$

Defina:

$$(\mathbf{v}(\sigma)\,,\,\,\xi(\sigma))\,=\,(\mathbf{v}(\sigma_{\mathtt{j}})\,,\,\,\xi(\sigma_{\mathtt{j}}))\,.$$

Ir a PASO 2.

Nota 2.7.

i) Es importante observar que el cálculo del nuevo σ en PASO 5 constituy ye la respuesta a la siguiente pregunta: ¿Para qué tamaño de paso obtengo el punto $(\mathbf{v}(\sigma_{\mathbf{j}}), \, \xi(\sigma_{\mathbf{j}})) \in \Gamma$ a partir de $(\mathbf{v}(\sigma_{\mathbf{j}}-1), \, \xi(\sigma_{\mathbf{j}}-1)^2$. En otras palabras, cuál debe ser el valor de σ para que $(\mathbf{v}(\sigma_{\mathbf{j}}), \, \xi(\sigma_{\mathbf{j}}))$ sea la solución del siguiente sistema no lineal:

(2.50)
$$G(v,\xi) = f(v) - \xi f(\overline{u}_0) = 0.$$

(2.51)
$$N(\mathbf{v}, \xi; \hat{\sigma}) = \dot{\mathbf{v}}(\sigma_{j-1})(\mathbf{v} - \mathbf{v}(\sigma_{j-1})) + \dot{\xi}(\sigma_{j-1})(\xi - \xi(\sigma_{j-1})) - \hat{\sigma} = 0.$$

Como $(\mathbf{v}(\sigma_{\mathbf{j}}), \xi(\sigma_{\mathbf{j}}))$ e Γ , la ecuación (2.50) se verifica. Por lo tanto, $\hat{\sigma}$ se despeja de (2.51) al reemplazar (\mathbf{v}, ξ) por $(\mathbf{v}(\sigma_{\mathbf{j}}), \xi(\sigma_{\mathbf{j}}))$.

ii) Finalmente, de nuestra experiencia numérica podemos comentar que una elección conveniente es tomar $K \leq 3$.

METODO ALTERNATIVO.

Debido a que la curva $\Gamma = G^{-1}(\{0\})$ se define utilizando sólo el punto inicial u_0 e $\partial\Omega$ en la función $G(u,\lambda)=f(u)-\lambda f(u_0)$, y puesto que a través de ella se genera una sucesión de puntos $(u_n,\lambda_n)_{n\geq 0}$, podemos tratar de diseñar un algoritmo que utilice esta información para avanzar más rápido hasta un vector (u,λ) e Γ con $\lambda=0$. Es decir, la idea que se plantea es ocupar el punto (u_n,λ_n) para calcular el siguiente vector sobre Γ , no sólo de la manera descrita antes, sino que definiendo además una nue va curva a partir de (u_n,λ_n) de la cual sea muy fácil recuperar puntos sobre Γ .

3.1. DESCRIPCION DE UN NUEVO ALGORITMO.

Como un primer intento de lograr lo comentado recién, se plantea un nuevo algoritmo que se basa en lo desarrollado en las secciones anteriores y en la siguiente proposición:

Proposición 3.1.

Sea $\Omega \subseteq \mathbb{R}^N$, abierto y consideremos una sucesión $\{u_n^*\}_n \in \mathbb{N} \subseteq \Omega$ tal que $\forall n \geq 0$ existe la curva de homotopía $\Gamma_n = G_n^{-1}$ ({0}), donde:

$$(3.1) G_n : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}$$

$$(u,\lambda) \rightarrow G_n(u,\lambda) = f(u) - \lambda f(u_n).$$

y f es de clase C^2 sobre \mathbb{R}^N .

Si
$$(\mathbf{u_k}, \lambda_k) \in \Gamma_{K-1}$$
; $\forall \ K \in \{1, \ldots, n\}$, entonces $(\mathbf{u_n}, \xi_n) \in \Gamma_0$, donde $\xi_n = \lambda_n \lambda_{n-1} \ldots \lambda_1$

<u>Demostración</u>: Por inducción sobre n. Para n = 1 es trivial ya que la hipótesis se traduce en $(u_1, \lambda_1) \in \Gamma_0$.

Supongamos que la tesis es válida para n. Probemos que también lo es para n + 1.

Si $(u_k, \lambda_k) \in \Gamma_{k-1}$, $\forall k \in \{1, \ldots, n, n+1\}$ entonces en particular $(u_k, \lambda_k) \in \Gamma_{k-1}$; $\forall k \in \{1, \ldots, n\}$. Aplicando hipótesis de inducción se deduce que $(u_n, \xi_n) \in \Gamma_0$ con $\xi_n = \lambda_n \lambda_{n-1} \ldots \lambda_1$, es decir: $f(u_n) - \xi_n \ f(u_0) = 0$.

Ahora, $(u_{n+1}, \lambda_{n+1}) \in \mathbb{R}$ es equivalente a escribir:

$$f(u_{n+1}) - \lambda_{n+1} f(u_{n}) = 0.$$

Reemplazando $f(u_n)$ por ξ_n $f(u_0)$ nos queda:

 $f(u_{n+1}) - \lambda_{n+1} \xi_n \ f(u_0) = 0, \ \text{lo cual indica que}$ $(u_{n+1}, \lambda_{n+1} \xi_n) \in \Gamma_0, \ \text{esto es} \ (u_{n+1}, \xi_{n+1}) \in \Gamma_0 \ \text{donde} \ \xi_{n+1} = \xi_{n+1} \xi_n = \lambda_{n+1} \xi_n = \lambda_{n+1} \lambda_n \cdots \lambda_1.$

Esto completa la inducción.

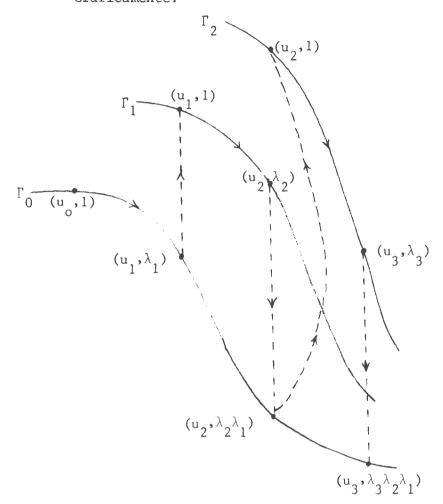
La proposición 3.1 nos sugiere un nuevo método para seguir la curva de homotopía Γ_0 desde $(u_0,1)$ con $u_0\in\partial\Omega$.

Suponiendo que (u_m, ξ_m) $\in \Gamma_0$, se define la nueva curva Γ_m por:

$$\Gamma_{m} = \{(u,\lambda) / G_{m}(u,\lambda) = f(u) - \lambda f(u_{m}) = 0\}$$

A partir de $(u_m,1) \in \Gamma_m$ se calcula un nuevo punto (u_{m+1},λ_{m+1}) e Γ_m siguiendo el procedimiento de Keller descrito en sección 2. Luego, el vector $(u_{m+1},\xi_{m+1}) = (u_{m+1},\lambda_{m+1},\xi_m)$ es el nuevo punto sobre Γ_0 .

Gráficamente:



Nota 3.1. Conviene mencionar aquí que lo que se espera es que la convergencia de (u_m, ξ_m) a un punto $(u^*, 0) \in \Gamma_0$ sea mucho más rápida que la sucesión generada por Keller. Por otro lado, es obvio que una condición suficiente para la existencia de Γ_m es que u_m sea valor regular de G_m . Sin embargo, determinar cuando ocurre esto último constituye un problema abierto.

Es importante comentar también que la obtención de (u_{m+1}, λ_{m+1}) e Γ_m a partir de $(u_m, 1)$ puede hacerse a través de varios puntos; esto es, para K e IN, fijo, la idea es generar sobre Γ_m K nuevos vectores y proyectar el último de ellos sobre Γ_0 .

Además, la dirección en la que se buscarán los puntos sobre $\Gamma_{\rm m}$ está determinada por la dirección del vector tangente $(\dot{\bf u}_{\rm m},\dot{\xi}_{\rm m})$.

4. ENSAYOS NUMERICOS.

En esta parte se muestran algunos ejemplos numéricos que ilustran el comportamiento del algoritmo descrito en sección 2. Todos los resultados fueron obtenidos mediante el programa FORTRAN que aparece en el apéndice y cuyas especificaciones se incluyen en [5]. Los resultados mencionados aquí y otros adicionales están explicados detalladamente en la tesis de HENRIQUEZ [5]. Se utilizó el computador DIGITAL modelo VAX-780 de la Universidad de Concepción, el cual tiene una capacidad memoria aproximada de 3 Mega.

Se considera una función f: $\mathbb{R}^N \to \mathbb{R}^N$ definida por sus N componentes:

$$f_i: \mathbb{R}^N \to \mathbb{R}$$

$$U = (x_1, ..., x_N) \rightarrow f_i(u) ; i = \overline{1,N}.$$

La notación a emplear es la siguiente:

Γ: indica la curva de Homotopía

 $(\mathbf{u_n}, \lambda_n)$: representa el n-ésimo vector sobre Γ .

 $(\dot{\mathbf{u}}_{\mathbf{n}},\dot{\lambda}_{\mathbf{n}}):$ indica el vector tangente a Γ en el punto $(\mathbf{u}_{\mathbf{n}},\lambda_{\mathbf{n}}).$

 σ_n : indica el tamaño de paso que permite calcular el punto (u_n, λ_n) sobre Γ a partir del vector $(u_n - 1, \lambda_n - 1)$.

- In: indica el número de iteraciones de Newton Raphson son requeridas para calcular (u_n, λ_n) a partir de $(u_n 1, \lambda_n 1)$ mediante el sistema no lineal (2.31)-(2.32) con un paso dado por σ_n .
- ϵ : precisión requerida para los vectores sobre Γ en la resolución de cada sistema no lineal (2.31)-(2.32).
- SIG: dirección en la cual se empieza a recorrer [.
- M: número máximo de iteraciones de N-R permitidas antes de modificar el tamaño del paso.
- o: tamaño de paso inicial.
- omin: tamaño de paso mínimo permitido.
- omax: tamaño de paso máximo permitido.
- σ_1, σ_2 : factores de contracción para la selección automática del tamaño del paso.
- K: número de vectores a introducir cada vez sobre la locale nón de la raíz para hallarla en forma más precisa.

En los ejemplos detallados a continuación se empleó ε = 10 $^{-6}$ y una dirección de búsqueda inicial SIG = -1.

Consideremos el siguiente problema de valor de contorno no lineal:

(4.1)
$$\frac{d^2x}{dt^2} = \frac{1}{2} (x(t) + t + 1)^3 ; x(0) = x(1) = 0.$$

Si aplicamos sobre (4.1) una discretización de diferencias finitas de $0(h^2)$ se obtiene un sistema no lineal en las incógnitas $x_k = x(t_k)$, con $t_k = k \ h; \ k = \overline{1,N}; \ h = 1/N + 1$. Si notamos $u = (x_1, \dots, x_N)^T$, entonces dicho sistema puede escribirse:

$$\begin{cases} f_{1}(u) = 2x_{1} - x_{2} + \frac{1}{2} \frac{1}{(N+1)^{2}} (x_{1} + \frac{1}{N+1} + 1)^{3} = 0 \\ f_{1}(u) = 2x_{1} - x_{1-1} - x_{1+1} + \frac{1}{2} \frac{1}{(N+1)^{2}} (x_{1} + \frac{1}{N+1} + 1)^{3} = 0; i = \overline{2, N-1} \\ f_{N}(u) = 2x_{N} - x_{N-1} + \frac{1}{2} \frac{1}{(N+1)^{2}} (x_{N} + \frac{N}{N+1} + 1)^{3} = 0. \end{cases}$$

Conviene observar que (4.2) tiene una única solución $u^* = (x_1^*, \dots, x_N^*)^T$ con $x_i^* \in [-0.5, 0]; \ \forall \ i = \overline{1, N}.$ Este ejemplo ha sido tomado de COSNARD, MORE [4].

La tabla 4.1 resume el comportamiento del algoritmo de Keller aplicado al sistema (4.2) con N = 5. Se utiliza un paso inicial σ = 0.5 y un valor mínimo σ_{\min} = 0.1. La selección del tamaño del paso considera

 $\frac{\text{Tabla 4.1}}{(N = 5, \sigma = 0.5, \sigma_{\min} = 0.1, M = 10, u_0 = (1,1,1,0,0)^T)}$

n	λ _n	λn	[©] n	I _n
0	1.0000	-0.5383		
1	0.7457	-0.4776	0.5	2
2	0.5226	-0.4140	0.5	2
3	0.3314	-0.3504	0.5	2
4	0.1714	-0.2901	0.5	2
5	0.0401	- 0.2363	0.5	2
6	-0.0666		0.5	2

una disminución de él en un 50% sólo si el número de iteraciones de N - R excede de M = 10. El punto inicial elegido es $u_0 = (1,1,1,0,0)^T$. Se observa aquí que no fue necesaria una bisección en el valor de σ ya que el valor I_n no excedió de 2. Por otro lado, el cambio de signo en λ ocurre entre los puntos 5 y 6. Esto indica que sobre la curva Γ , entre los vectores (u_5, λ_5) y (u_6, λ_6) , existe el punto $(u^*, 0)$ donde u^* es la raíz del sistema no lineal.

La tabla 4.2 resume la curva para el sistema (4.2) con N = 20. Se utiliza un paso inicial σ = 0.2 y valores límite σ_{\min} = 0.1, σ_{\max} = 0.4. La selección automática del tamaño del paso considera factores de contracción α_1 = 0.3 y α_2 = 0.7. El : nto inicial es u_0 = $(-1,-1,\ldots,-1)^T$. Puede apreciarse que el tamaño del paso es aumentado al doble en cada etapa. Más aún, la convergencia al correspondiente punto de Γ se logra cada vez en só-

Tabla 4.2

$$(N = 20, \sigma_{\min} = 0.1, \sigma = 0.2, \sigma_{\max} = 0.4, \alpha_{1} = 0.3, \alpha_{2} = 0.7, u_{0} = (-1, -1, ..., -1)^{T})$$

n	$\lambda_{ m n}$	· ^{\lambda} n	σ _n	I n
0	1.00000	-0.37992	_	-
1	0.84384	-0.40142	0.4	1 /
2	0.67871	-0.42499	0.4	1
3	0.50376	-0.45034	0.4	1
4	0.31842	-0.47683	0.4	1
5	0.12235	-0.50387	0.4	1
6	-0.08457		0.4	1

lo una iteración de N - R. Este hecho hace pensar que si se hubiera permitido un mayor valor para σ_{\max} , entonces se habría avanzado más rápido a través de Γ y se habría localizado antes el cambio de signo en λ .

Consideremos ahora el sistema (4.2) con N = 10. La tabla 4.3 mues tra los resultados obtenidos con $\mathbf{u}_0 = (-1,-1,\dots,-1)^T$ y las constantes $\sigma_{\min} = 0.1$, $\sigma_{\max} = 0.5$, $\alpha_1 = 0.3$, $\alpha_2 = 0.7$. Se observa que la convergencia a los puntos de Γ mediante el sistema (2.31)-(2.32) se logra con $\mathbf{I}_n = 1$. En esta circunstancia de rápida convergencia, la selección automática del tamaño del paso mantiene el valor de σ , aún cuando se admite que en tal caso sería más conveniente aumentarlo. Ahora, si se aplica la técnica descrita en sección 2.4 entre los vectores $(\mathbf{u}_{14},\lambda_{14})$ y $(\mathbf{u}_{15},\lambda_{15})$, con K = 3, entonces al cabo de 3 etapas de refinamiento se han calculado 9 vectores sobre Γ y se obtiene una nueva localización para la raíz dada por los vectores

Tabla 4.3

(N = 10, $\sigma_{\min} = 0.1$, $\sigma = 0.2$, $\sigma_{\max} = 0.5$, $\alpha_{1} = 0.3$, $\alpha_{2} = 0.7$, $u_{0} = (-1, -1, ..., -1)^{T}$)

n	λ _n	λ _n	o _n	In
0	1.00000	-0.31498	-	-
1	0.93678	-0.31727	0.2	1
3	0.80884	-0.32255	0.2	1
5	0.67862	-0.32874	0.2	1
7	0.54574	-0.33577	0.2	1
8	0.47821	-0.33959	0.2	1
9	0.40989	-0.34359	0.2	1
10	0.34076	-0.34778	0.2	1
11	0.27077	-0.35213	0.2	1
12	0.19990	-0.35665	0.2	1
13	0.12811	-0.36132	0.2	1
14	0.05537	-0.36613	0.2	1
15	-0.01835		0.2	1

 (u_{23},λ_{23}) y (u_{24},λ_{24}) . Precisamente, tabla 4.4 muestra estos dos vectores, y en ella se aprecia una precisión de 6 dígitos decimales.

Para el mismo sistema (4.2) con N = 10, se considera el vector inicial $u_0 = (5,5,\ldots,5)^T$, el cual está relativamente lejos de la solución exacta. Un resumen de las características de Γ se detalla en tabla 4.5. Allí se observa que con un paso $\sigma = 0.5$ se requieren 33 vectores para detectar el cambio de signo en λ . Además, la obtención de cada uno de ellos ocupa sólo una iteración de N - R.

(u ₂₃ , λ ₂₃)	(u ₂₄ , \lambda_24)
-0.043164982	-0.043164815
-0.081577155	-0.081577000
-0.114485712	-0.114485568
-0.140973574	-0.140973437
-0.159908693	-0.159908561
-0.169877199	-0.169877069
-0.169089980	-0.169089849
-0.155249532	-0.155249397
-0.125355889	-0.125355744
-0.075416532	-0.075416373
1.0 E-10	-1.82 E-7

Tabla 4.5

$$(N = 10, \sigma_{min} = 0.1, \sigma = 0.2, \sigma_{max} = 0.5, \alpha_1 = 0.3, \alpha_2 = 0.7, u_0 = (5,5,...,5)^T)$$

n	λ _n	λ'n	σ _n	n
0	1.00000	-0.11391	-	-
5	0.73935	-0.09491	0.5	1
8	0.60502	-0.08432	0.5	1
11	0.48615	-0.07429	0.5	1
14	0.38187	-0.06485	0.5	1
17	0.29133	-0.05600	0.5	1
20	0.21358	-0.04779	0.5	1
23	0.14764	-0.04028	0.5	1
26	0.09232	-0.03362	0.5	1
29	0.04628	-0.02794	0.5	1
32	0.00794	- 0.02339	0.5	1
33	-0.00344		0.5	1

Esto nos hace meditar nuevamente acerca de la elección de los parámetros σ , σ_{\min} y σ_{\max} . Sin duda que al permitir un mayor aumento en el tamaño del paso se habrían necesitado menos de 33 etapas para hallar una aproximación de la raíz del sistema. Ovbiamente, esta localización habría sido de mayor longitud. Sin embargo, esto no es ningún inconveniente pues to que el proceso de refinamiento descrito en sección 2.4 nos permite calcular la raíz con la precisión requerida. En definitiva, debe quedar claro que la convergencia a la raíz está asegurada a partir de la existencia de la curva de Homotopía regular; el valor del tamaño del paso sólo incide sobre la velocidad con que avanza a través de Γ . Al aplicar el proceso de localización con K = 3, entre vectores (u_{32}, λ_{32}) y (u_{33}, λ_{33}) , se utilizan 4 etapas de refinamiento y se generan 9 vectores más sobre Γ . Finalmente

se sitúa la raíz entre (u_{41', 41}) y (u_{42', 42}), donde 0 < λ_{41} < 1.0 E-9 y 0 > λ_{41} > -1.0 E-9.

Por último, con el objeto de observar el comportamiento del método ante una mala estimación inicial, consideramos nuevamente el sistema (2.4) para N = 40 con $\mathbf{u}_0 = (50,50,\dots,50)^T$. Los resultados numéricos se muestran en tabla 4.6. Allí se observa que se requiere un tamaño de paso bastante grande ($\sigma = 30$) para llegar en pocas etapas a una localización de la raíz. Es indudable que con un valor menor para σ también se habría hallado el cambio de signo en λ , pero para ello se habrían calculado una mayor cantidad de vectores sobre Γ . En este último caso, el desgaste se compensa en parte con el hecho que la localización final es más precisa. No obstante, debe recordarse que el objetivo no es ubicar una gran cantidad de puntos sobre Γ , sino que avanzar a través de ella, detectar rápidamente los cambios de signo en λ y aplicar sobre éstos la técnica de localización descri

Tabla 4.6

$$(N = 40, \sigma_{\min} = 1.0, \sigma = 20, \sigma_{\max} = 30, \alpha_1 = 0.3, \alpha_2 = 0.7, u_0 = (50, ..., 50)^T)$$

n	$\lambda_{\mathbf{n}}$	$\dot{\lambda}_{ m n}$	σ _n	ĭ,
		4.5	11	*1
0	1.00000	-0.00889	_	
1	0.75646	-0.00737	30.0	2
2	0.55641	-0.00599	30.0	2
3	0.39541	-0.00477	30.0	2
4	0.26903	-0.00368	30.0	2
5	0.17294	-0.00274	30.0	2
6	0.10291	-0.00194	30.0	2
7	0.05481	-0.00128	30.0	2
8	0.02456	-0.00075	30.0	2
9	0.00808	-0.00036	30.0	2
10	0.00126	-0.00011	30.0	2
11	-0.00029		30.0	2

ta en sección 2.4. Después de realizar 5 etapas de refinamiento, se obtienen 10 vectores adicionales sobre Γ entre (u_{10},λ_{10}) y (u_{11},λ_{11}) . Al cabo de este proceso, el vector $(u^*,0)$ se encuentra sobre la curva de Homotopía entre los vectores (u_{20},λ_{20}) y (u_{21},λ_{21}) , cuyas componentes se muestran en tabla 4.7. Aquí se aprecia que la localización final entrega una precisión de al menos 5 dígitos decimales. A modo de observación final, podemos comentar que con ninguno de los métodos NEWTON, BROWN o BRENT, se obtiene con vergencia para $u_0=(50,\ldots,50)^{\rm T}$.

Tabla 4.7

(u ₂₀ ,λ ₂₀)	(u ₂₁ , \(\lambda_{21}\)
-0.012040348	-0.012042073
-0.023772311	-0.023775548
-0.035184049	-0.035188706
-0.046263239	-0.046269226
-0.056996920	-0.057004148
-0.067371448	-0.067379828
-0.077372450	-0.077381897
-0.086984780	-0.086995207
-0.096192459	-0.096203783
-0.104978622	-0.104990759
-0.113325455	-0.113338324
-0.121214124	-0.121227644
-0.128624706	-0.128638797
-0.135536105	-0.135550689
-0.141925968	-0.141940966
-0.147770583	-0.147785921
-0.153044786	-0.153060386
-0.157721835	-0.157737624
-0.161773293	-0.161789197
-0.165168891	-0.165184835
-0.167876372	-0.167892285
-0.169861332	-0.169877142
-0.171087034	-0.171102671
-0.171514211	-0.171529603
-0.171100838	-0.171115915
-0.169801892	-0.169816583
-0.167569076	-0.167583312
-0.164350517	-0.164364228

(u ₂₀ , \lambda_20)	(u ₂₁ , ^λ 21)
-0.160090430	-0.160103545
-0.154728739	-0.150103343
-0.148200661	-0.148212373
-0.140436231	-0.140447134
-0.131359776	-0.131369798
-0.120889321	-0.120898387
-0.108935918	-0.108943953
-0.095402890	-0.095409816
-0.080184970	-0.080190707
-0.063167330	-0.063171796
-0.044224466	-0.044227575
-0.023218932	-0.023220593
2.0 E-09	-1.0 E-10

COMENTARIOS FINALES.

Sin duda que el ejemplo desarrollado en sección 4 ha dado suficiente evidencia de las buenas propiedades del método de homotopía. Existen también otras situaciones numéricas, que no se incluyen aquí por falta de espacio, en las cuales se aprecia que el algoritmo de Keller es capaz de hallar secuencialmente varias raíces de un sistema no lineal. Un análisis más detallado y una mayor cantidad de ensayos numéricos pueden encontrarse co. [5].

En resumen, el procedimiento descrito en este artículo, en conjunto con las variantes propuestas, constituye una excelente alternativa para resolver sistemas no lineales de gran tamaño, en los cuales se dispone de malas estimaciones iniciales. Más aún, las regiones con jacobiano nulo no afectan la aplicación de la técnica de seguimiento descrita aquí.

APENDICE

En las 6 páginas siguientes aparece codificado el programa NOLINH, el cual utiliza el método descrito en este artículo para hallar todas las raíces de un sistema no lineal algebraico de ecuaciones.

El usuario debe agregar una subrutina de lectura que proporcione todos los datos iniciales para la ejecución del algoritmo. Además, las funciones que constituyen el sistema no lineal deben definirse en el FUNCTION F según indicaciones dadas ahí.

Con respecto al algoritmo de refinamiento, se deja extendida la invitación a que cualquier lector interesado en el tema pueda solicitar directamente a los autores de este artículo, la codificación del mismo.

```
CALL NRAPH(nn, Vm, Up, PSIm, Ln, Lp, Fo, DV, DPSI, Y, Wnor, In, Fp)
                                                                                 DO Jelenn
                                                                                     Vm(J) = Vm(J) + DV(J)
   DOUBLE PRECISION Un(10).Fo(10).Z(10).Ur(10).DV(10).Wn(2).Fr(11.52).
# SIGMER, SIGMIN, Siz, LP, PSIM, PSIM1, VM(50), VMES1(50), YF(50), Y(50),
                                                                                  END DO
* SIGNO, Ln, GM1, K, ZNOR, AMIN, PROD, Lnmas1, Wnor, G, F, Max11, Max12, SG,
                                                                                 PSIm=PSIm+DPSI
# Eps.DPSI, SUM, Det, SIGe
                                                                                 Wn(I)=Hnor
   INTEGER OF
                                                                              END DO
   DATA EPS/1E-6/
                                                                              DO I=1,nn
  SIGNO=-1.
                                                                                 Vmas1(I)=Vm(I)
   CALL LECTUR(nn.Un.SIGo.SIGMax.SIGMin.Mmax.a1.a2.kn.OP)
                                                                              END DD
   GO TO (15,100), DP
                                                                              PSIm1=PSIm
   K = 0
                                                                              m=2
   n=0
                                                                              IF(Wn(2).GT.a2+Wn(1))THEN
  w=0
                                                                                          Is=1
   In=0
                                                                                          SIG*SIG/2.
   Ind=1
                                                                                          IF(Is.EG.2)THEN
   Ln=1.
                                                                                                      DO I=1.nn
   DO I=1.nn
                                                                                                         Vm(I)=Un(I)+SIG+U=(I)
      Fo(I)=F(I,Un,nn)
                                                                                                      END DO
                                                                                                      PEIm=Ln+SIG#LP
   END DO
                                                                                                      m = 1
   CALL DERIVA(Un.nn.Fe)
   DO I=1.NN
                                                                                                      GO TO 70
                                                                                                      END IF
      F_P(I,nn+1)=F_0(I)
                                                                                          IF(SIG.GE.SIGmin)GOTO 20
      F_P(I,nn+2)=F_O(I)
                                                                                          CALL FALLA
   END DO
                                                                                          CALL EXIT
   CALL ECLIN(Fp.nn, 2, 2, dct)
                                                                                          ELSE
  ZNOR=PROD(Z,Z,nn)
                                                                                          IF(Wn(2).LT.a1+Wn(1))THEN
   SIG=SIGo
   IF (In.EG.1) THEN
                                                                                                       IF(Is.EG.1)GO TO 40
                 LP=O.
                                                                                                       Is=2
                 2(nn)=1.
                                                                                                       SIG=2.*SIG
                 DO J=1.nn-1
                                                                                                       IF (SIG.GT.SIGMax) THEN
                    Lenn-J
                                                                                                                          SIG=SIGmax
                    SUM=0.
                                                                                                                          DO I=1,nn
                    DO K=L+1.nn-1
                                                                                                                             U_{m}(I)=U_{n}(I)+SIG+U_{m}(I)
                        SUM=SUM+F#(L,K)+Z(K)
                                                                                                                          END DO
                    END DO
                                                                                                                          PSIM=Ln+SIG+LP
                    Z(L) = (-Fe(L,nn)-SUM)/Fe(L,L)
                                                                                                                          m = 1
                                                                                                                          GD TO 70
                 END DO
                 R=DSGRT(PROD(Z,Z,nn))
                                                                                                                          END IF
                 DO I=1.nn
                                                                                                       SIG=AMIN(SIG, EIGHEX)
                    Z(I) = Z(I) / R
                                                                                                       GO TO 20
                 END DO
                                                                                                       END IF
                 IF (PROD(Z.Ue.nn).LT.0) THEN
                                                                               END IF
                                                                           40 DO I=1.nn
                                         SG=-1.
                                                                                  Y2(I)=G(Vmas1.PSIm1.Fo,I.nn)
                                         ELSE
                                                                              END DO
                 56=1.
```

```
END IF
                                                                                   GM1=DSGRT(PROD(Y2,Y2,nn))
                                                                                   IF (GM1.LT.EPs) GO TO 50
                   DO I=1.nn
                      Up(I)=SG+2(I)
                                                                                   IF(m.GE.Mmax)THEN
                                                                                                 IF(SIG.GE.SIGmin)THEN
                   END DO
                                                                                                                    SIG = SIG/2.
                   FLSE
                                                                                                                     I \le = 1
     LP=SIGNO/DSGRT(1.+ZNOR)
                                                                                                                     DO I=1.nn
     DG I = 1 . nn
                                                                                                                        U_{m}(I) = U_{n}(I) + SIG + U_{n}(I)
        Ue(I)=Le*Z(I)
     END DO
                                                                                                                     END DO
                                                                                                                     PSIm=Ln+SIG#LF
     END IF
                                                                                                                     m = 1
     CALL IMPRE(k,nn,Un,Ln,Up,Lp,SIG,n,m,Ind,Ie)
     m = 0
                                                                                                                     GD TO 70
                                                                                                                     END IF
     Ind=1
                                                                                                  TYPE +, 'Elemin otro Uo y recomenzar'
     Is=0
20 DO I=1.nn
                                                                                                  CALL EXIT
        V_{m}(I) * U_{n}(I) + SIG*U_{n}(I)
                                                                                                  END IF
     END DO
                                                                                   m=m+1
     PSIm=Ln+SIG+LP
                                                                                   DD I=1.nn
     DO I=1.2
                                                                                       Um(I)=Umas1(I)
     END DO
                                                                                                   CALL ECLIN(GP:n1.Zm.Zm.Det)
     PEIntePEIm1
                                                                                                   DO I=1.nn
                                                                                                      DV(I) = Zm(I)
70 CALL NRAPH(nn.Um.Up.PSIm.Ln.Lp.Fo.DV.DPSI.Y.Wnor,In.Fe)
     DO J=1.nn
                                                                                                      Y(I) = Zm(I)
        U_{\text{MB}} = 1 \text{ (i)} = U_{\text{M}} \text{ (i)} + DU \text{ (i)}
                                                                                                   END DO
     END DC
                                                                                                   DPSI=Zm(nn+1)
     PEIN1 = PEIN+DPEI
                                                                                                   In = 1
     GD TO 40
                                                                                                   ELSE
 50 Lnwas1=PSIm1
                                                                                    DPSI=PROD(Up, T.nn)/(PROD(Up, Y.nn)+Lp)
     IF(Ln+Lnwas1.LT.O)THEN
                                                                                    DO I=1,nn
          n = n + 1
                                                                                       DU(I) = DPSI + Y(I) - T(I)
          CALL IMPRE(K.nn. Vmas1, Lnmas1, UP, LP, SIG, n, m, Ind, Ie)
                                                                                    END DO
          CALL REFINA(Un, Ln, Up, Lp, SIG, Kn, Fo, Lnmasi, nn, Eps, K, n)
                                                                                    END IF
          K = K + 1
                                                                                    W=PROD(DU.DU.nn)
          TYPE 60.char(27).k
                                                                                    w=DSGRT(W+DPSI##2)
 60
          FORMAT('+',A1,'[24;5H',' Raiz W',I2,' Desea sequir? (S/N) :',$) RETURN
          ACCEPT 200,51
                                                                                    END
 200
          FORMAT(A5)
          IF (SI.EG. 'NO'.OR.SI.EG. 'N') GO TO 10
          Ind=2
          END IF
                                                                                    FUNCTION PROD(X,Y,n)
     IF(m.EQ.2.OR.m.LE.O.2*Mmax)THEN
                                                                                    DOUBLE PRECISION X(50),Y(50),PROD
                                   SIG=2.#SIG
                                                                                    PRGD=0.
                                   IF (SIG.GT.SIGMAX) THEN
                                                                                    DO I=1.N
                                                                                    PROD=PROD+X(I)#Y(I)
                                                      SIG=SIGMax
                                                      END IF
                                                                                    END DO
                                                                                    RETURN
                                   END IF
     IF(PRCD(UP,Y,nn)+LP,LT,0) THEN
                                                                                    END
```

SIGNO=-1. ELSE

WRITE (7,30) n

```
30 FORMAT(/.20X,'LA LOCALIZACION SE DETERMIND EN LA ITERACION : 1.13)
                                                                          JUEROUTINE ECLIN(A, n, XX, YY, DET)
                                                                          DOUBLE PRECISION A($1,52).YY(50),XX(50),AMAX,GAK,FAC,Det.SUM
40 FORMAT(/,5X,'La Localizacion se determino en Iteracion :',I3)
   RETURN
                                                                          DET=1.
   END
                                                                          DO I=1:n=1
                                                                             AMAX=0.
                                                                             DO 4=1.n
   SUPROUTINE FALLA
                                                                                IF (AMAX.LT.DAES (A(J.I))) THEN
                                                                                                          AMAX=DAES(A(J.I))
   TYPE *,'NO HAY CONVERGENCIA'
   RETURN
                                                                                                          IND=J
                                                                                                           END IF
   END
                                                                             END DO
                                                                             TF (AMAX, LE, 1E-8) THEN
                                                                                              DET=O.
   SUBSOUTINE LECTURION. UO. SIG. SIGMAX. SIGMAN. HNAX. Alfal. alfa2. kn. CP)
                                                                                              RETURN
   DOUBLE PRECISION UO(50), SIGNAX, SIGNID, SIG
                                                                                              END IF
   Esta EUDROUTINE permite leer los siguientes parametros:
                                                                             TE(IND.NE.I)THEN
           :Orden och sistema
                                                                                          DFT=-DET
.
   U0
           "Vector aproximacion inicial, que permite cefinir la curva
                                                                                          DO K=1.n+2
            de Homotopia.
                                                                                          GAR=A(IND,K)
            :Paso or Homotopia inicial.
                                                                                          A(IND,K)=A(I,K)
   SIGNAX : Paso maximo or Homotopia, cota superior para la selección
                                                                                          A(I,K)=GAR
            automatica del tamano del Paso.
                                                                                          END DO
   SIGmin : Paso minimo de Homotopia, cota inferior para le selección
                                                                                          END IF
            automatica del tamano del paso.
                                                                              DO L=I+1, n
   Mmax
           :Numero maximo de iteraciones de Newton Raphson permitigo
                                                                                 IF (A(L,I), NE.O) THEN
            en cada etara.
                                                                                                 FAC=A(L,I)/A(I,I)
   alfai
           :Factor de contraccion para realizar el refinamiento de la
                                                                                                 DO K=1+1.n+2
                                                                                                    A(L,K)=A(L,K)-FAC+A(I,K)
            localizacion. (O < alfa1).
   alfa2
           :Factor de contracción para realizar el refinamiento de la
                                                                                                 END DO
            localizacion. (alfa1 < alfa2 < 1).
                                                                                                 END IF
           :Numero de tamanos de paso a introducir cada vor en el rom
   Kn.
                                                                              END DO
            finamiento.
                                                                          END DO
           'Variable entera que asume valores 1 c 2, sesun se desec
                                                                           XX(n) = A(n, n+1)/A(n, n)
            esecutar el programa o terminar la esecucion.
                                                                           DG J=1,n-1
                                                                              L=n-J
   Agemas debe aprirse un Archivo en el cual quedara almacenada la
                                                                              5UM = 0.
  informacion de salida.
                                                                              DO K=L+1,n
                                                                                 SUM=SUM+A(L,K) *XX(K)
   RETURN
   END
                                                                              END DO
                                                                              XX(L) = (A(L,n+1) - SUM)/A(L,L)
                                                                           END DC
                                                                           VY(n)=A(n,n+2)/A(n,n)
   SUERCUTINE IMPRE(Kinn, U.C. Up, Cp, SIGMA, n.m. Ind. Ic)
                                                                           DC J=1,n-1
   DOUBLE PRECISION U(50), L. UP (50), LP, SIGMA
                                                                              L = Fr = L
   IF (Ind.EG.2) WAITE (7,60) le
```

60 FORMATION, (Refinantion to, etama 11,12,/5), 19(1H-1)

IF:n.Eb.O.OR.Ind.EQ.2.OR.Ind.EQ.3)THEN

SUM=C.

```
DO K=L+1.n
   SUM # SUM + A (L,K) + YY (K)
   END DO
   YY(L)=(A(L,n+2)-SUM)/A(L,L)
END DO
DC 1=1.0
DET=DET=A(I,I)
END DO
RETURN
END
FUNCTION AMIN(A.I)
DOUBLE PRECISION A.E.AMIN
IF (A.LT.I) THEN
           AMIN=A
           ELSE
OMINER
END IF
RETURN
END
FUNCTION Maxi2(A.E)
DOUBLE PRECISION A.E. Max12
A=DAES(A)
E=DARS(E)
IF (A.LT.E) THEN
           Max12=E
           ELSE
Maxi2=A
END IF
KETURN
END
```

```
FUNCTION F(1,X,N)
DOUBLE PRECISION X(10),F

c

c Esta FUNCTION evalua la 1-esima componente del sistema
c F(x)=0 de orden N, en el punto x de dimension N.
c Esta FUNCTION debe ser implementada por el usuario. A m
se entresa el sipuiente sistema de orden 2.
c

GO TO (10,20),I

40 F=X(1)++2+X(2)++2-5.
RETURN
ZO F=EXP(-X(1))-X(2)
RETURN
END
```

```
FUNCTION Max:1(C,n)
     DOUBLE PRECISION C(50), Maxi1
     Max:1=DAES(C(1))
     DC ImDan
        IF(max11.LT.DAES(C(I)))THEN
                               Maxi1=DADS(C(I))
                               END IF
     END DO
     RETURN
     END
     SUIROUTINE REFINA(Un.Ln.Up.Lp.SIG.K.Fo.Lnmas1.nn.Eps.1.n)
     DOUPLE PRECISION Un(50), Up(50), Fo(50), Lp, SIG, Lnmas1, Eps, SIGMA, S(11),
  Up1(50),Vp(50),SIo,Vpo(50),SIpo,V1(50),SI1,V(50),SI,DV(50),DPSI,
  + F=(51.52).SS.A.W.SIp1.Y(50).T.Maxi1.Maxi2.Y2(50).PROD.T1.T2.
  # SIGNO, Ln.F
     Esta SUEROUTINE realiza el REFINAMIENTO de la localización entresa-
     da por el programa NOLINA utilizando el algoritmo descrito en seccion
C
     2.4
     RETURN
     END
```

REFERENCIAS

- [1] ALLGOWER, E.: "A survey of homotopy methods for smooth mappings".

 In Numerical solution of nonlinear equations, ed.

 por E. Allgower, K. Glashoff y H. Peitgen. Lecture

 Notes in Mathematics 878, 1980.
- [2] BRANIN, F.: "Widely convergent methods for finding multiple solutions of simultaneous nonlinear equations",

 I.B.M.J. Research Develop. 16, (1972), pp 504-522.
- [3] BROWN, K.: "Computer oriented algorithms for solving systems of simultaneous non linear algebraic equations". In Numerical Solution of Systems of Non linear Equations, ed. G. Byrne and C. Hall, Acad. Press, 1973.
- [4] COSNARD, M.; MORE, J.: "Numerical Solution of Nonlinear equations".

 ACM. trans. Mathem. Software., 5, N° 1, (1979) pp 64
 85.
- [5] HENRIQUEZ, E.: "Resolución Numérica de sistemas no lineales algebraicos. Implementación Computacional". Tesis (Ingeniero Matemático), Universidad de Concepción, 1985.
- [6] KELLER, H.: "Global homotopies and Newton methods", in Symposium:

 Recent advances in Numerical Analysis, ed. por C. de

 Boor, 1978, pp 73-94.
- [7] MILNOR, J.: "Topology from the differentiable viewpoint", University Press of Virginia, Charlotte, VA, 1965.
- [8] ORTEGA, J.; RHEINBOLDT, W.: "Iterative solution of nonlinear equations in several variables". Acad. Press 1970.
- [9] PERCELL, P.: "Note on a Global Homotopy". Draft Manuscript, Univ. Houston, Texas, 1978.
- [10] SMALE, S.: "A convergent process of price adjustment and global Newton methods". J. Math. Econ, 3, 1976, pp 107-120.