



BANCO CENTRAL DE COSTA RICA
DIVISIÓN ECONÓMICA
DEPARTAMENTO DE INVESTIGACIÓN ECONÓMICA

INTRODUCCIÓN A LOS MÉTODOS NUMÉRICOS

Desirée Castrillo Rojas

Nota Técnica

DEC-DIE-014-2009-NT. Marzo 2009

Documento de trabajo del Banco Central de Costa Rica, elaborado por el Departamento de Investigación Económica

Las ideas expresadas en este documento son responsabilidad de la autora y no necesariamente representan la opinión del Banco Central de Costa Rica

Tabla de contenido

1. Introducción.....	3
2. Conceptos Importantes.....	4
• Errores de truncamiento.....	4
• Errores de redondeo.....	5
Otro término importante es el de algoritmo numérico.....	5
Convergencia.....	5
3. Métodos numéricos de aproximación para raíces de ecuaciones no lineales	6
Método de la Bisección.....	7
Método del Punto Fijo	8
Método de Newton.....	9
Método de la Secante	11
4. Métodos numéricos de aproximación para sistemas de ecuaciones lineales	12
Método de Jacobi.....	12
Método Gauss- Seidel	13
5. Métodos de sistemas numéricos de aproximación para sistemas de ecuaciones no lineales	14
Método de Newton en varias variables.....	14
6. Uso de métodos numéricos en Economía	16
7. Consideraciones finales	17
8. Bibliografía	18
9. Anexos.....	19
9.1 Algoritmos de los métodos aquí estudiados.....	19
Algoritmo 1: Método de la Bisección.....	19
Algoritmo 2: Método de Punto Fijo	20
Algoritmo 3: Método de Newton-Raphson	20
Algoritmo 4: Método de la Secante	21
Algoritmo 6: Método de Jacobi.....	22
Algoritmo 7: Método de Gauss-Seidel.....	23
Algoritmo 8: Método de Newton en varias variables.....	24
9.2 Implementación de algunos algoritmos en Mathematica	25

1. Introducción

Muchos de los fenómenos de la vida real son modelados matemáticamente con el fin de poder explicarse, sin embargo en la mayoría de los casos éstos no pueden ser solucionados por medio de algún método exacto¹ y aunque algunas veces se puede lograr su solución ésta puede resultar demasiado laboriosa en términos de tiempo y recursos computacionales. Los métodos numéricos (MN) solucionan este tipo de problema mediante la búsqueda de una solución numérica aproximada y el cálculo del error asociado, el cual se espera que sea lo suficientemente pequeño.

Los MN son herramientas o técnicas, diseñadas mediante algoritmos, que permiten la resolución de problemas matemáticos que tienen como característica un elevado nivel de complejidad y que generalmente no pueden resolverse con los métodos analíticos tradicionales y cuando esto es posible se requiere de un elevado costo.

La matemática numérica es antigua, pero ha sido gracias al desarrollo computacional que se ha logrado desarrollar en forma aplicada. Sus aplicaciones son inmensas, utilizándose intensivamente en ingeniería, economía, ciencias naturales y otros.

Actualmente, el Departamento de Investigación Económica del Banco Central de Costa Rica (DIE) realiza las simulaciones de política de su Modelo Macroeconómico de Proyección Trimestral mediante el uso de estos métodos y a través del paquete econométrico de Eviews, sin embargo, también se está implementando el uso de software WinSolve para igualmente realizar simulaciones ya que se considera existen muchas ventajas en su uso. En aras de lograr un mejor entendimiento del mecanismo por el cual estos software realizan sus aproximaciones se realiza este documento, su objetivo es introducir al uso de algunas de las diferentes técnicas de métodos numéricos usadas en software de aplicación como Eviews y WinSolve, los cuáles son utilizados en el DIE.

El presente documento incluye, además de esta introducción una sección de conceptos importantes. En la sección 3 se explican los MN de aproximación de raíces para ecuaciones no lineales, continuando la siguiente sección con los MN para la resolución de sistemas de ecuaciones lineales, para continuar con los MN de sistemas de ecuaciones no lineales. En la sección 6 se comenta un poco sobre el uso de los MN en la Economía y su

¹ Se conoce como método exacto o analítico

presencia en los “software” de aplicación de Eviews y WinSolve y finalmente se exponen las conclusiones.

2. Conceptos Importantes

En este apartado se explica algunos conceptos importantes de manejar a la hora de utilizar la técnicas numéricas que se exponen en las secciones siguientes.

Los valores o cifras significativas permiten determinar la confianza de un valor numérico, su importancia en los MN radica primero en que dado que éstos son técnicas iterativas, se puede establecer como criterio de parada el hecho de que una solución alcance determinado número de cifras significativas; y segundo, se utilizan para decidir la precisión de un método numérico. Por ejemplo, el número 2.820×10^3 posee cuatro dígitos significativos.

El término de **exactitud**, mide cuán cerca se encuentra el valor calculado con respecto a su valor verdadero y el de **precisión** se refiere a la cercanía de una aproximación a un valor con respecto a las aproximaciones anteriores.

Un valor obtenido mediante el uso de los MN al ser una aproximación difiere de su solución real en una cantidad que generalmente se denomina **error**. Éste término es de vital importancia ya que la estimación obtenida es evaluada dependiendo de qué tan pequeño sea este. El error asociado mide el grado de exactitud e incertidumbre de éstos.

Los errores son clasificados en dos grandes categorías en la mayoría de la literatura y son causados generalmente por el uso de computadores para realizar los cálculos numéricos. Estos se denominan **error de truncamiento** y **error de redondeo**. A continuación se explican cada uno de ellos.

- **Errores de truncamiento:** se originan debido a que se procede a calcular una solución mediante una aproximación y por el empleo de un número finito de términos de la serie para realizar un cálculo que requiere un número infinito. Un ejemplo de error de truncamiento sucede con el valor del número de Euler $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$, el cual

se aproxima con $e^x \approx \sum_{i=0}^k \frac{x^i}{i!}$ $k < \infty$ y por lo tanto se ha acertado el número de términos y por ende se incurre en el error.

- **Errores de redondeo:** Se originan debido a que los cálculos numéricos no pueden realizarse con una precisión infinita y al empleo de un número finito de términos para realizar un cálculo que requiere un número finito de ellos. Las técnicas utilizadas son las que siguen:
 1. Redondeo: se eliminan cierto número de cifras significativas realizando un ajuste sobre la última cifra no descartada.
 2. Corte o poda: igual que con el redondeo se prescinde de cierto número de cifras no significativas sin realizar un ajuste en la última no descartada.

La medición del error se realiza en forma absoluta o relativa. Burden (2001) lo define de la siguiente manera: Si p^* es una aproximación de p , el error absoluto es $|p - p^*|$ y el error relativo es $\frac{|p - p^*|}{|p|}$ siempre que $p \neq 0$.

Otro término importante es el de **algoritmo numérico**, también llamado método constructivo. Como se dijo anteriormente, los MN permiten encontrar soluciones mediante el uso de éstos. Un algoritmo es un procedimiento que describe un número finito de pasos que pueden ejecutarse de manera lógica. El objetivo de un algoritmo es ser la base para implementar un procedimiento que resuelve un problema o que aproxima una solución al problema (Burden, 2001).

Convergencia: La convergencia es la propiedad que tienen algunas sucesiones de tender a un límite. En métodos iterativos como los numéricos, se construye una sucesión S_n de aproximaciones a la solución del problema. S_n se dice convergente si converge a la solución buscada.

Ahora bien, suponga que p_n es una sucesión que converge a p , con $p_n \neq p$ para todo n . Si existen las constantes α y λ y con

$$\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$$

luego, p_n converge a p de orden α , con un error asintótico constante λ . Una técnica iterativa de la forma $p_n = g(p_{n-1})$ se dice de orden α si la secuencia p_n converge a la solución $p = g(p)$ de orden α (Burden, 2001).

La constante α dirá cuán rápido converge una sucesión, siendo las de más alto orden, de convergencia más acelerada. Dos casos especiales se dan cuando $\alpha=1$ y cuando $\alpha=2$, en donde se dice que la convergencia de la sucesión es lineal o cuadrática respectivamente, siendo la segunda de convergencia más rápida.

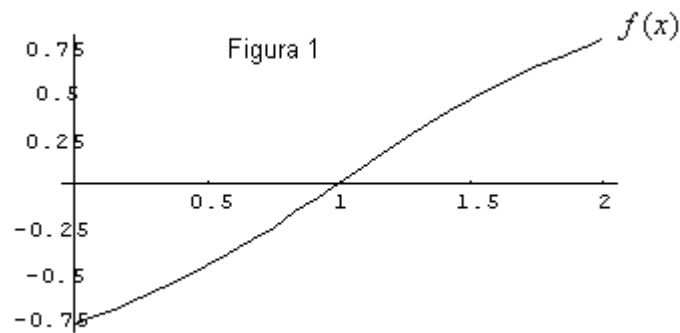
Una vez introducidos algunos términos importantes que ayudarán a comprender el funcionamiento de los MN se procede a continuación a explicar algunas de las diversas técnicas numéricas para la resolución problemas matemáticos. Los algoritmos para cada una de las técnicas aquí expuestas se encuentran en el Anexo 1, así como la implementación de algunos de ellos en Mathematica se encuentran en el Anexo 2.

3. Métodos numéricos de aproximación para raíces de ecuaciones no lineales

El comportamiento de los fenómenos económicos puede modelarse matemáticamente por medio de ecuaciones, por lo tanto, la búsqueda de soluciones es una labor diaria del economista. Las soluciones encontradas son llamadas generalmente raíces o ceros de la función y son básicamente los valores de x que hacen igual a cero la ecuación, o sea, $f(x) = 0$. Los MN estiman valores de x que aproximan a cero la función.

Una función lineal se define como una igualdad en la que intervienen solo sumas y restas de variables a la primera potencia y representan rectas en el plano cartesiano y que debe de cumplir las propiedades de aditividad ($f(x+y) = f(x) + f(y)$) y homogeneidad. Su forma básica es $y = mx + b$ con m igual a la pendiente de la recta y b el intercepto con el eje y . Por lo tanto, una ecuación no lineal viola todas las propiedades anteriores.

La raíz de una función, geoméricamente hablando, es el punto donde la función $f(x)$ cruza el eje x . Por ejemplo, en la siguiente gráfica, se tendría que $x \approx 1$, o sea, esta función encuentra una solución en este punto.



En este documento se hablará de los métodos más conocidos para la solución de este tipo de problemas, dentro de los cuáles se incluyen el de la Bisección, Punto Fijo, Secante y Newton-Raphson.

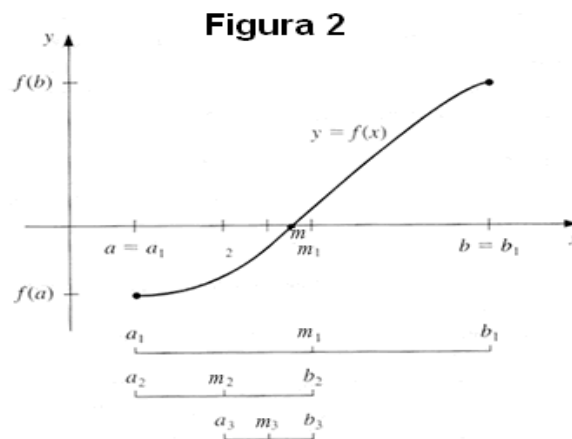
Método de la Bisección

Es uno de los métodos más antiguos y elementales de búsqueda de raíces y se basa en el Teorema del Valor Intermedio², partiendo del supuesto de que dada una función $f(x)$ continua en un intervalo a, b , con $f(a)$ y $f(b)$ con signos opuestos, existe al menos un valor $x \in a, b$ tal que $f(x) = 0$.

El método consiste en que una vez identificados los puntos a y b (con signos opuestos) se construye la primera aproximación de la forma $x_1 = \frac{a+b}{2}$. Luego si $f(a) * f(x_1) < 0$, entonces los puntos tienen signos opuestos y por ende la solución se encuentra en a, x_1 , por lo que para la siguiente iteración se toma $b = x_1$. Si $f(a) * f(x_1) > 0$, quiere decir que la solución se encuentra en x_1, b por lo tanto $a = x_1$. Por último, si $f(a) * f(x_1) = 0$ es la raíz de la ecuación.

² Teorema del Valor Intermedio: Sea f continua en un intervalo a, b y diferenciable en a, b Entonces existe al menos algún punto c en el intervalo abierto tal que la tangente a la curva en c es paralela a la recta secante que une los puntos $(a, f(a))$ y $(b, f(b))$, es decir, $f(b) - f(a) = f'(c)(b - a)$.

Con respecto al criterio de parada, se puede usar ya sea un número máximo de iteraciones o hacer que el error relativo menor a un nivel de error. Su principal crítica es su lenta convergencia.



En la Figura 2 se explica gráficamente el método. Se supone que en el intervalo a_1, b_1 se encuentra la solución, luego $f(a)$ y $f(b)$ deben tener signos opuestos. El método continúa según se explicó anteriormente hasta aproximarse a la solución en el punto m .

Método del Punto Fijo

Un punto fijo c es aquel en donde $y = c$, en otras palabras, donde su imagen coincide con él. Además, debe garantizar la existencia y unicidad³ de ese punto. El problema de encontrar la solución para ecuaciones de la forma $f(x) = 0$ puede solucionarse transformándola de alguna manera en una equivalente $g(x) = x$ para alguna función x , así si p es raíz de $f(x) = 0 \leftrightarrow f(p) = 0 \leftrightarrow p = g(p) \leftrightarrow p$ es raíz de $x = g(x)$

La convergencia del método se logra gracias a la aplicabilidad del Teorema del Punto Fijo⁴. El método consiste en la aproximación de una solución inicial y luego con la ecuación

³ Si $g \in C[a, b]$ y $g(x) \in [a, b]$ para todo $x \in [a, b]$ luego, g tiene un punto fijo en $[a, b]$. Además, $g'(x)$ existe sobre $[a, b]$ y una constante positiva $k < 1$ existe con $|g'(x)| \leq k$ para todo $x \in (a, b)$, luego el punto fijo en $[a, b]$ es único.

⁴ Si g es una función continua en $[a, b]$ y $g(x) \in [a, b]$ para todo $x \in [a, b]$, entonces g tiene por lo menos un punto fijo en $[a, b]$. Si además, $g'(x)$ existe para todo $x \in [a, b]$, y $|g'(x)| \leq K < 1$ para todo $x \in [a, b]$, K constante, entonces g tiene un único punto fijo $x \in [a, b]$. La sucesión $\{x_n\}$, con n definida, se encuentra mediante la fórmula de iteración:

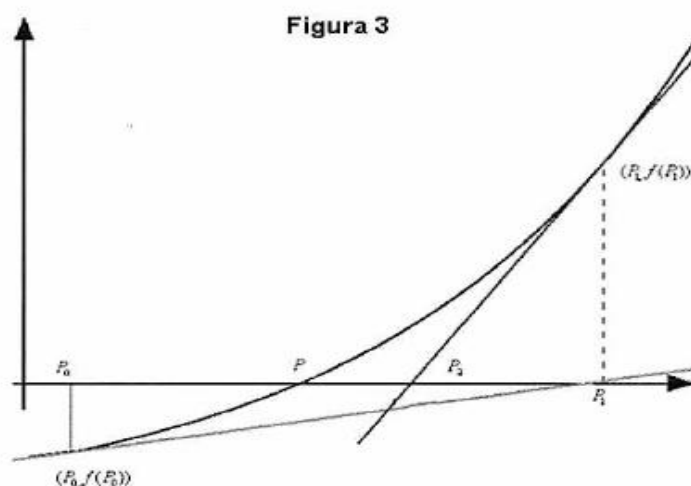
$$x_n = g(x_{n-1}), n = 1, 2, 3, \dots$$

$p_n = g(p_{n-1})$ se genera una sucesión de aproximaciones la cuál converge a la solución de la ecuación $f(x) = 0$. A esta función g se le conoce como función iteradora, la cual converge si y solo si $|g'(x)| < 1$.

Método de Newton

También llamado el método de Newton-Raphson es uno de los más poderosos y mundialmente conocidos para la resolución de problemas de búsqueda de raíces. Se basa en que la función $f(x)$ es derivable sobre un intervalo cerrado a, b , por lo que f tiene una pendiente definida y una única línea tangente en cada punto en a, b .

Con respecto a las formas de obtener el algoritmo se mencionarán aquí dos maneras. La Figura 3 se muestra como se obtienen aproximaciones a partir de rectas tangentes. Se inicia con el punto aproximado inicial P_0 , luego, la aproximación P_1 es la intercepción con el eje x de la recta tangente con el punto $(P_0, f(P_0))$. La siguiente aproximación P_2 se genera por la intercesión de la recta tangente a la curva en el punto formado por la aproximación anterior y la función evaluada en ese valor y el eje x (en este caso $(P_1, f(P_1))$) y así sucesivamente.



La segunda manera de obtener el algoritmo es desarrollando $f(x)$ en una serie de Taylor, para un entorno del punto x_n . Suponga que $f \in C^2$ a, b . Tome $\bar{x} \in a, b$ como una aproximación a x , tal que $f'(\bar{x}) \neq 0$ y $|p - \bar{x}|$ es pequeña. El primer polinomio de Taylor para $f(x)$ expandido alrededor de \bar{x}

$$f(x) = f(\bar{x}) + (x - \bar{x})f'(\bar{x}) + \frac{(x - \bar{x})^2}{2} f''(\xi(x))$$

donde $\xi(x)$ se encuentra entre x y \bar{x} . Como $f(p) = 0$, si $x = p$, se tiene que

$$0 = f(\bar{x}) + (p - \bar{x})f'(\bar{x}) + \frac{(p - \bar{x})^2}{2} f''(\xi(p))$$

y como $|p - \bar{x}|$ es pequeño $(p - \bar{x})^2$ es aún más, por lo tanto

$$0 \approx f(\bar{x}) + (p - \bar{x})f'(\bar{x})$$

$$p \approx \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

Así el Método de Newton inicia con una aproximación inicial y luego genera la secuencia

p_n $n=0$ ∞ , por

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$$

Con respecto al criterio de parada, igualmente se selecciona un nivel de tolerancia del error

($\varepsilon > 0$) y luego construir p_1, \dots, p_N hasta que

$$\begin{aligned} |p_N - p_{N-1}| &< \varepsilon, \\ \frac{|p_N - p_{N-1}|}{p_N} &< \varepsilon, \quad p_N \neq 0 \end{aligned}$$

ó

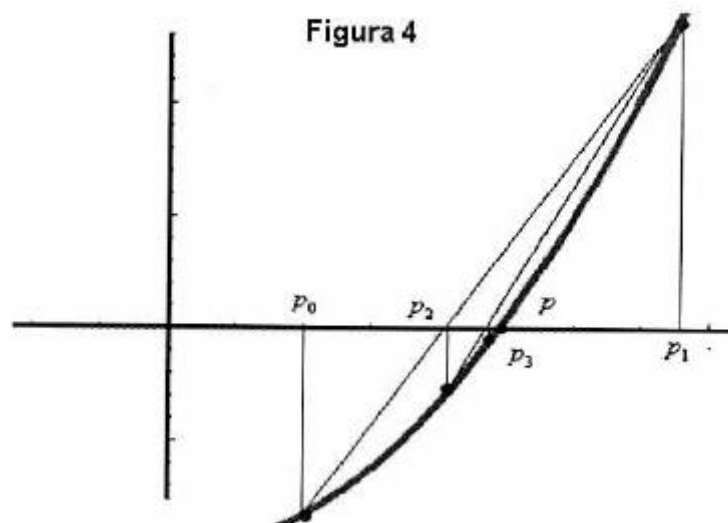
$$f(p_N) < \varepsilon$$

Este método, al igual que cualquier otro, posee sus ventajas y desventajas. Su mayor ventaja, y por la cuál es uno de los preferidos, es que cuando converge a la raíz (el cero de

la función), lo hace de una manera bastante rápida. Sin embargo, entre sus desventajas se puede mencionar el hecho de que al no trabajar sobre un intervalo, en donde se asegure que existe una raíz, no existe ninguna garantía de que se esté aproximando a ésta. Además, es imprescindible que la primera derivada sea diferente de cero para que el método pueda utilizarse y que la aproximación inicial sea bastante buena para que pueda converger.

Para resolver el problema que conlleva el cálculo de la derivada en el método anterior, existe la técnica llamada Método de la Secante.

Método de la Secante



Como se muestra en la Figura 4, el Método de la Secante inicia, a diferencia de la técnica anterior, con una aproximación de dos puntos (p_0, p_1) , luego el punto p_2 es calculado como el intercepto de la línea que une $(p_0, f(p_0))$ y $(p_1, f(p_1))$. La aproximación p_3 es el resultado de la intersección de la línea que une $(p_1, f(p_1))$ y $(p_2, f(p_2))$ con el eje x , y así sucesivamente.

Al igual que el método de Newton, éste no asegura la convergencia la que dependerá en gran medida de las aproximaciones iniciales que se tengan, por lo que un análisis gráfico siempre es importante a la hora de calcular los valores de aproximación inicial.

4. Métodos numéricos de aproximación para sistemas de ecuaciones lineales

Los sistemas lineales del tipo $Ax = b$ son resueltos por métodos directos e indirectos. Dentro de los primeros se puede encontrar el método de Gauss y el de descomposición L-U⁵. Sin embargo, muchas veces a estos no les es posible resolver sistemas de gran dimensión y se hace necesario usar otras herramientas más poderosas. Los métodos indirectos dan soluciones a sistemas lineales más complejos. Dentro de estos están el Método de Jacobi y el de Gauss-Seidel, ambas técnicas iterativas.

Método de Jacobi

Inicia con un vector de aproximación $x^{(0)}$ a la solución del sistema (x), generándose la secuencia $x^{(k)}$ que converge a x . Esta técnica convierte el sistema $Ax = b$ en un sistema equivalente de la forma $x = Tx + c$ para una matriz fija T y un vector c . Resuelve la i -ésima ecuación en $Ax = b$ para x_i ($a_{ii} \neq 0$) y se obtiene

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^n \left(-\frac{a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \text{ para } i = 1, 2, \dots, n$$

luego genera cada $x_i^{(k)}$ de los componentes de $x^{(k-1)}$ para $k \geq 1$ de la siguiente manera:

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j \neq i}}^n (-a_{ij}x_j^{(k-1)}) + b_i}{a_{ii}}, \text{ para } i = 1, 2, \dots, n$$

El sistema se escribe de la forma $x^{(k)} = Tx^{(k-1)} + c$, descomponiendo la matriz A en su diagonal (D), en su estricta triangular inferior (L) y en su estricta triangular superior (U), reescribiéndose el problema en

$$(D-L-U)x=b$$

⁵ Para un mayor desarrollo de estos métodos, consultar Burden, R. (1996)

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ -a_{21} & \ddots & 0 & \ddots \\ \vdots & \ddots & 0 & 0 \\ -a_{n1} & \cdots & \cdots & -a_{n,n-1} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ 0 & \ddots & 0 & \ddots \\ \vdots & \ddots & 0 & -a_{n-1,n} \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$A = D - L - U$$

La ecuación $Ax = b$, o $(D - L - U)x = b$ es transformada en

$$Dx = (L + U)x + b$$

si D^{-1} existe ($a_{ii} \neq 0$) para cada i

$$x = D^{-1}(L + U)x + D^{-1}b$$

Los resultados en forma matricial de la técnica de Jacobi serían

$$x^{(k)} = D^{-1}(L + U)x^{(k-1)} + D^{-1}b$$

$$\text{si } T_j = D^{-1}(L + U) \text{ y } c_j = D^{-1}b$$

$$x^{(k)} = T_j x^{(k-1)} + c_j$$

Con respecto al criterio de parada del método, se tiene el de un número de iteraciones máximo, otro criterio sería el de la tolerancia, el cual se describe de la siguiente manera:

$$\frac{\|x^{(k)} - x^{(k-1)}\|}{\|x^{(k)}\|} \leq Tol$$

claro está, usando la norma adecuada (se recomienda la norma infinita para este método).

Método Gauss- Seidel

El método de Gauss-Seidel es muy similar al de Jacobi, con la diferencia de que el primero actualiza los valores más recientes en cuanto se van obteniendo, de manera que converge más rápido. Esto es, los componentes $x^{(k-1)}$ son usados para calcular x^k , de tal manera que

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^n (a_{ij}x_j^{k-1}) + b_i}{a_{ii}}$$

La mayoría de las veces los resultados de este método son mejores a los de Jacobi debido a la actualización de valores. Sin embargo, existen sistemas para los cuales Gauss-Seidel no converge, pero si lo hace Jacobi.

5. Métodos de sistemas numéricos de aproximación para sistemas de ecuaciones no lineales

Como se mencionó anteriormente, una ecuación no lineal no se puede expresar en el plano cartesiano como una línea (como se hace en una lineal). Dada sus características la mayoría de las veces es casi imposible su cálculo mediante métodos algebraicos, por lo que se recurre a los MN.

La mayoría de los métodos de resolución del tipo de sistemas no lineales son modificaciones de algunos de los métodos para la resolución de ecuaciones no lineales. Dentro de los métodos utilizados se encuentran el de Gauss-Seidel para sistemas no lineales, el de la Máxima Pendiente, el de la Secante, el del Punto Fijo y el de Newton en varias variables, entre otros. En el presente documento se expondrá este último ya que hereda la propiedad de una rápida convergencia del método de Newton-Raphson explicado anteriormente.

El método de Gauss-Seidel para sistemas de ecuaciones no lineales, es muy similar al usado en sistemas lineales, con excepción de que se necesita despejar en forma manual las ecuaciones y escribirlas en forma de función.

Método de Newton en varias variables

El método de Newton en varias variables se espera converja cuadráticamente, y se utiliza para sistemas de ecuaciones de pequeña dimensión. Un sistema no lineal tiene la siguiente forma

$$\begin{aligned}f_1(x_1, x_2, \dots, x_n) &= 0, \\f_2(x_1, x_2, \dots, x_n) &= 0, \\&\vdots \\f_n(x_1, x_2, \dots, x_n) &= 0\end{aligned}$$

se puede definir además, la función F de la siguiente manera

$$F(x_1, x_2, \dots, x_n) = (f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n))^t$$

donde las funciones f_1, f_2, \dots, f_n son coordenadas de la función F .

Con esto, se puede interpretar el sistema de ecuaciones no lineales como el cero del campo vectorial F , o sea, $F(x) = 0$. Además, para garantizar la convergencia cuadrática del sistema se debe definir

$$G(x) = x - A(x)^{-1}F(x),$$

Asumiendo $A(x)$ como una matriz no singular y un punto fijo $p \in \mathbb{R}^n$, el cuál es solución desconocida de $F(x) = 0$.

Se puede construir el polinomio de Taylor que aproxima a f_i alrededor de p de tal manera que

$$f_i(x) = f_i(p) + \nabla f_i(p)(x - p) + r_2(x)$$

Además, si se define J como la matriz jacobiana⁶ de F (arreglo de gradientes) y con $r_2(x)$ que tiende a 0, se puede sustituir el sistema no lineal por el siguiente lineal

$$F(p) + J(x^{(ap)})(x - x^{(ap)}) = 0,$$

con

$$J(x^k) = \begin{pmatrix} \nabla f_1(x^k) \\ \nabla f_2(x^k) \\ \dots \\ \nabla f_n(x^k) \end{pmatrix}$$

Luego, manipulando un poco algebraicamente se obtiene finalmente

$$x^{(k+1)} = x^{(k)} - \left[J(x^k)^{-1} \right] F(x^{(k)})$$

La debilidad de este método radica en el cálculo e inversión de la matriz jacobiana en cada iteración. Es debido a lo anterior, que muchas veces se evita el segundo paso encontrando un vector y de manera que $J(x^{(k)})y = -F(x^{(k)})$ y luego la nueva aproximación es obtenida

⁶ La matriz Jacobiana deberá ser no singular.

sumando al vector “y” $x^{(k)}$. El inconveniente de este procedimiento es que necesita más pasos y por ende la convergencia llega en un tiempo mayor que en el método sin modificar.

6. Uso de métodos numéricos en Economía

Como se dijo anteriormente son muchas las aplicaciones de los métodos numéricos en diferentes ámbitos, en este apartado interesa mencionar un poco sobre sus aplicaciones en economía. Su uso puede iniciarse con técnicas de optimización en las que se pueden mencionar por ejemplo algoritmos heurísticos de optimización, tales como el Recalentamiento recosido (Simulated Annealing en inglés), Búsqueda Tabú (Tabu search), algoritmos genéticos y de optimización multiobjetivo, entre otros. En el campo de la economía dinámica, como el cálculo de variaciones y de control óptimo, se puede mencionar más específicamente el modelo de agentes con vidas infinitamente largas (el modelo básico de ciclos económicos reales, modelos con fricciones, economías abiertas y agentes heterogéneos) y el modelo de generaciones traslapadas, entre muchas otras aplicaciones.

En el DIE se utiliza una serie de paquetes especializados entre los que se encuentran Eviews y WinSolve. Estas herramientas usan análisis numérico tanto para la resolución de sistemas de ecuaciones como para procesos de optimización.

Con respecto a Eviews, éste utiliza por defecto el método de Gauss-Seidel a la hora del cálculo de sistema de ecuaciones no lineales. También se encuentran las opciones del Método de Newton y el de Broyden⁷.

En el caso de WinSolve, posee cuatro métodos para la resolución de sistemas de ecuaciones, a saber, el de Gauss-Seidel, el rápido Gauss-Seidel⁸, Jacobi y el de Newton. Se recomienda el uso de éste último método en el caso de sistemas pequeños y el uso del método de Gauss-Seidel y Jacobi con sistemas de más grande escala.

⁷ El método de Broyden es una modificación del Método de Newton, el cuál trata de disminuir el costo de cada iteración usando aproximaciones para las derivadas del sistema de ecuaciones (Eviews: User guide II, 2007)

⁸ El cuál es una variante del método de Gauss-Seidel que permite una convergencia más rápida

7. Consideraciones finales

Del análisis de los métodos numéricos expuestos en este documento se concluye que existe una gran variedad de métodos para los diferentes problemas de resolución de ecuaciones y sistemas matemáticos, según su naturaleza.

Muchas veces se tiene que para un determinado problema, el método utilizado no converja, por lo que es recomendable probar con otro con características similares en busca de una aproximación a la solución de la ecuación o sistema que se busca resolver

La importancia de tener conocimientos sobre las características de cada uno radica en saber cuál método aplicar a determinado problema y encontrar así la mejor aproximación a la solución buscada.

Las técnicas numéricas de aproximación son utilizadas en las simulaciones del Modelo Macroeconómico de Proyección Trimestral, por lo que el conocerlas más a fondo ayudará a un mayor entendimiento del proceso en su conjunto para sus ejecutores.

Igualmente, en la construcción de un modelo dinámico estocástico de equilibrio general, el entendimiento de estas técnicas será de mucho beneficio en su construcción.

Este documento solo pretende dar una explicación de los fundamentos del análisis numérico, por lo tanto sus aplicaciones en la Economía no se exponen de manera detallada.

8. Bibliografía

Bravo, Juan (2005) "El método de Newton-Raphson-la alternativa del ingeniero para resolver sistemas de ecuaciones no lineales". Scientia et Technica. Año XI. N°27

Burden, R. Faires, D. (2001). Numerical Analysis. Sétima Edición. Thomson Learning, Inc.

Díaz, Wladimiro (1998). "Método numéricos". Universidad de Valencia.

Montero, Gustavo (2007) "Métodos iterativos para la resolución de ecuaciones no lineales". Universidad de las Palmas de Gran Canaria.

Pérez, Justo. "Material didáctico métodos matemáticos VI" Universidad de la Laguna.

Pierse, Richard (2007). "WinSolve Version 4: An introductory guide" Department of Economics, University of Surrey.

Quantitative Micro Software (2007). "Eviews 6: User guide II"

Saenz, Carlota (2006) "Resolución numérica de sistema de ecuaciones no lineales" Universidad Pontificia Comillas.

Vadillo, F. (2006). "Ecuaciones no lineales". Universidad del País Vasco

castrillord@bccr.fi.cr

9. Anexos

9.1 Algoritmos de los métodos aquí estudiados

Algoritmo 1: Método de la Bisección

Inicio: Sea $g(x)$ una función continua.

Parámetros iniciales : N =máximo número de iteraciones

Tol=nivel de precisión con respecto a la solución exacta

a, b =solución inicial

Defina $n=0$.

$FA=f(a)$

Mientras $n \leq N$

$$p = a + \frac{(b-a)}{2};$$

$$FP = f(p)$$

$$\text{Si } FP = 0 \text{ ó } \frac{(b-a)}{2} < Tol$$

Salida p

Parar

Incremente $n = n + 1$

Si $FA * FB > 0$, entonces $b = p$

Salida "El método falló después de N iteraciones"

Parar

Algoritmo 2: Método de Punto Fijo

Inicio: Sea $g(x)$ una función continua.

Parámetros iniciales : N =máximo número de iteraciones

Tol=nivel de precisión con respecto a la solución exacta

X_0 =solución inicial

Defina $n=0$.

Mientras $n \leq N$

$$x_{n+1} = g(x_n)$$

$$\text{Si } |x_{n+1} - x_n| < Tol$$

Salida x_{n+1}

Parar

Incremente $n = n + 1$

Salida "El método falló después de N iteraciones"

Parar

Algoritmo 3: Método de Newton-Raphson

Parámetros iniciales : N =máximo número de iteraciones

Tol=nivel de precisión con respecto a la solución exacta

p_0 =solución inicial

Defina $n=0$.

Mientras $n \leq N$

$$p = p_0 - \frac{f(p_0)}{f'(p_0)} \text{ (cálculo de } p_1 \text{)}$$

$$\text{Si } |p - p_0| < Tol$$

Salida p (El proceso fue exitoso)

Parar

Incremente $n = n + 1$

$p_0 = p$ (actualizar)

Salida "El método falló después de N iteraciones"

Parar

Algoritmo 4: Método de la Secante

Parámetros iniciales: N=máximo número de iteraciones

Tol=nivel de precisión con respecto a la solución exacta

p_0 y p_1 =soluciones iniciales

Tome $n=2$

$$q_0 = f(p_0)$$

$$q_1 = f(p_1)$$

Mientras $n \leq N$

$$\text{Tome } p = p_1 - \frac{q_1(p_1 - p_0)}{q_1 - q_0} \quad (\text{Cálculo de } p_i)$$

$$\text{Si } |p - p_1| < Tol$$

Salida p (El procedimiento fue exitoso)

Parar

Incremente $n = n + 1$

Tome

$$p_0 = p_1;$$

$$q_0 = q_1;$$

$$p_1 = p;$$

$$q_1 = f(p);$$

Salida "El método falló después de N iteraciones"

Parar

Algoritmo 6: Método de Jacobi

Resolver $Ax=b$, dada una solución inicial $x^{(0)}$

Parámetros iniciales: dimensión del sistema

A=entradas a_{ij} , con $i \leq j, j \leq n$ de la matriz A

b=entradas b_i

Tol=nivel de precisión con respecto a la d solución exacta

$x^{(0)}$ =solución inicial

Tome $k=1$

Mientras $k \leq N$

 Para $i=1, \dots, n$

$$x_i^{(k)} = \frac{-\sum_{\substack{j=1 \\ j \neq i}}^n (a_{ij}x_j^{(0)}) + b_i}{a_{ii}}$$

 Si $\|x - x^{(0)}\| < Tol$

 Salida (x_1, \dots, x_n) (El procedimiento fue exitoso)

 Parar

 Incremente $k = k + 1$

 Para $i=1, \dots, n$

 Tome $x_i^{(0)} = x_i$

Salida "El método falló después de N iteraciones"

Parar

Algoritmo 7: Método de Gauss-Seidel

Resolver $Ax=b$, dada una solución inicial $x^{(0)}$

Parámetros iniciales : dimensión del sistema

A =entradas a_{ij} , con $i \leq j, j \leq n$ de la matriz A

b =entradas b_i

Tol =nivel de precisión con respecto a la d solución exacta

$x^{(0)}$ =solución inicial

Tome $k=1$

Mientras $k \leq N$

 Para $i=1, \dots, n$

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j^{(0)} + b_i}{a_{ii}}$$

 Si $\|x - x^{(0)}\| < Tol$

 Salida (x_1, \dots, x_n) (El procedimiento fue exitoso)

 Parar

Incremente $k = k + 1$

Para $i=1, \dots, n$

 Tome $x_i^{(0)} = x_i$

Salida "El método falló después de N iteraciones"

Parar

.....

Algoritmo 8: Método de Newton en varias variables

Aproximar el sistema no lineal $F(x)=0$, dada una solución inicial $x^{(0)}$

Parámetros iniciales : dimensión del sistema

 Tol=nivel de precisión con respecto a la d solución exacta

$x = (x_1, x_2, \dots, x_n)$ =solución inicial

Tome $k=1$

Mientras $k \leq N$

 "Calcular $F(x)$ y $J(x)$ " $J(x)_{i,j} = \left(\frac{\partial f_i(x)}{\partial x_j} \right)$ para $1 \leq i, j \leq n$

 Resuelva el sistema lineal $n \times n$ $J(x)y = -F(x)$

 Haga $x=x+y$

 Si $\|y\| < Tol$,

 Salida "x" (El procedimiento fue exitoso)

 Parar

Incremente $k = k + 1$

Salida "El método falló después de N iteraciones"

Parar

9.2 Implementación de algunos algoritmos en Mathematica

Método de la Bisección

```
Clear[Biseccion];
h[x_] := e-x + 4*x3 - 5;
Print["El grafico de ", h[x_], " " "es:"];
Tol = 10-10;
Plot[h[x], {x, -4, 2}, PlotStyle -> {Red}, AxesStyle -> {Blue, Thickness[0.005]}, PlotPoints -> 20];

Biseccion[f_, x0_, x1_, n_, Tol_] := Module[{i, xu = x0, S, p0 = x0, p1 = x1, p},
  S = {{x0, x1}};
  For[i = 0, i < n && Abs[p0 - p1] > Tol, i++,
    p = p0 + (p1 - p0) / 2;
    If[f[p] f[p0] > 0, p0 = p;
      f[p] = f[p0], p1 = p];
    S = Append[S, {p0, p1}];
  ];

  Print["Los resultados finales de convergencia son:"];
  Print["x0= ", p0, " ", "x1= ", p1];
  Print["Los resultados de las iteraciones son:"]; S]
Biseccion[h, 1.0`20, 2.0`20, 30, Tol] // TableForm
```

Método de Newton-Raphson

```
Clear[Newtonla];
h[x_] := e-x + 4*x3 - 5;
Print["El grafico de ", h[x_], " " "es:"];
Plot[h[x], {x, -4, 2}, PlotStyle -> {Red}, AxesStyle -> {Blue, Thickness[0.005]}, PlotPoints -> 20];
Tol = 10-10;

Newtonla[f_, x0_, n_, Tol_] := Module[{i, xu = x0, S, g},
  S = {x0};
  g[x_] = (x - f[x] / f'[x]);
  xm = g[xu];
  For[i = 0, i < n && Abs[xm - xu] > Tol, i++,
    xu = xm;
    xm = g[xu];
    S = Append[S, xu];
  ];
  Print["El resultado final de convergencia es:"];
  Print["xm= ", xu];
  Print["Los resultados de las iteraciones son:"];
  S]
Newtonla[h, 1.0`20, 40, Tol] // TableForm
```

Método Jacobi

El sistema lineal:

$$2x_1 - x_2 + x_3 = -1$$

$$2x_1 + 2x_2 + 2x_3 = 4$$

$$-x_1 - x_2 + 2x_3 = -5$$

tiene la solución $(1, 2, -1)^T$

```
In[127]:= A =  $\begin{pmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}$ ; b = {7, 2, 5};
```

```
(*A:Matriz de coeficientes de las variables de cada ecuación del sistema*)
(*b: vector del resultado de cada ecuación*)
aDLU[A_] := Module[{D, L, U, n},
  n = Length[A];
  U = Append[Table[Join[Table[0, {i}], -Take[A[[i]], i - n]], {i, 1, n - 1}], Table[0, {n}]]; (*matriz triangular superior*)
  L = Prepend[Table[Join[-Take[A[[i]], i - 1], Table[0, {n + 1 - i}]], {i, 2, n}], Table[0, {n}]]; (*matriz triangular inferior*)
  D = DiagonalMatrix[Table[A[[i, i]], {i, 1, n}]]; (*matriz diagonal*)
  {D, L, U};

NormaInf[x_] := Max[Abs[x]]; (*norma infinita*)
Tol = 10-5; (*nivel máximo de tolerancia*)
(*x0: solución inicial*)

Jacobi[A_, b_, x0_, norma_, Tol, iter_] := Module[{i, x1 = x0, x2, D, L, U, T, c, S},
  {D, L, U} = aDLU[A];
  T = Inverse[D].(L + U);
  c = Inverse[D].b;
  x2 = T.x1 + c;
  S = {x1, x2};
  For[i = 1, i < iter && norma[x2 - x1] > Tol, i++,
    x1 = x2;
    x2 = T.x1 + c;
    S = Append[S, x2];
  ];
  Print["El resultado final de convergencia es:"];
  Print["El vector de solución {x1,x2,x3} es: ", x2];
  Print["El resultado de cada iteración realizada:"];
  S]

Jacobi[A, b, {0.0, 0.0, 0.0}, NormaInf, Tol, 100] // TableForm
```

Método Gauss-Seidel

El sistema lineal:

$$2x_1 - x_2 + x_3 = -1$$

$$2x_1 + 2x_2 + 2x_3 = 4$$

$$-x_1 - x_2 + 2x_3 = -5$$
 tiene la solución $(1, 2, -1)^T$

In[3]:=
$$A = \begin{pmatrix} 2 & -1 & 1 \\ 2 & 2 & 2 \\ -1 & -1 & 2 \end{pmatrix}; b = \{-1, 4, -5\};$$

```
(*A:Matriz de coeficientes de las variables de cada ecuación del sistema*)
(*b: vector del resultado de cada ecuación*)
sDLU[A_] := Module[{D, L, U, n},
  n = Length[A];
  U = Append[Table[Join[Table[0, {i}], -Take[A[[i]], i-n]], {i, 1, n-1}], Table[0, {n}]]; (*matriz triangular superior*)
  L = Prepend[Table[Join[-Take[A[[i]], i-1], Table[0, {n+1-i}]], {i, 2, n}], Table[0, {n}]]; (*matriz triangular inferior*)
  D = DiagonalMatrix[Table[A[[i, i]], {i, 1, n}]]; (*matriz diagonal*)
  {D, L, U};

NormaInf[x_] := Max[Abs[x]]; (*norma infinita*)
Tol = 10^-5; (*nivel máximo de tolerancia*)
(*s0:solución inicial*)

Clear[GaussSeidel];
GaussSeidel[A_, b_, xo_, norma_, tol_, n_] := Module[{i, xl, x2, D, L, U, T, c, S = {xo}},
  {D, L, U} = sDLU[A];
  T = Inverse[D - L].U;
  c = Inverse[D - L].b;
  xl = T.xl + c;
  S = {xl, x2};
  For[i = 1, i < n && norma[x2 - xl] > Tol, i++,
    xl = x2;
    x2 = T.xl + c;
    S = Append[S, x2];
  ];
  Print["El resultado final de convergencia es:"];
  Print["x= ", x2];
  Print["El resultado de cada iteración realizada:"];
  S];

GaussSeidel[A, b, {0.0, 0.0, 0.0}, NormaInf, Tol, 30] // TableForm
```

Método de Newton en Varias Variables

```
Options[MNewton] = {IterMax -> 30, Norma -> NormaInf, Tolerancia -> 10^-15};
MNewton[F_List, x_List, x0_List_, Opc_] :=
  Module[{J, y0, F0, y1, J0},
    n = IterMax /. {Opc} /. Options[MNewton];
    Nrm = Norma /. {Opc} /. Options[MNewton];
    Tol = Tolerancia /. {Opc} /. Options[MNewton];
    J = D[F, {x}];
    y1 = x0; y0 = x0 + 2 Tol;
    For[i = 0, (i < n) && Nrm[y0 - y1] > Tol, i++,
      y0 = y1;
      J0 = J /. Thread[x -> y0];
      F0 = -F /. Thread[x -> y0];
      y1 = LinearSolve[J0, F0];
      y1 = y1 + y0;
      Print["i = ", i, " ", y0];
    ];
    Print["El resultado final de convergencia es:"];
    Print["{x, y, w, z} = ", y1];
  ]
```