

**Estudio numérico del método de Newton  
inexacto para resolver sistemas de ecuaciones  
no lineales**

Oscar Andrés Pérez Guerrero

**Universidad del Cauca  
Facultad de Ciencias Naturales, Exactas y de la Educación  
Programa de Matemáticas  
Departamento de Matemáticas  
Popayán  
2018**

# Estudio numérico del método de Newton inexacto para resolver sistemas de ecuaciones no lineales

Estudiante:

Oscar Andrés Pérez Guerrero

Director

Favián Enrique Arenas Aparicio

Universidad del Cauca  
Facultad de Ciencias Naturales, Exactas y de la Educación  
Programa de Matemáticas  
Departamento de Matemáticas  
Popayán  
2018

## Nota de aceptación

---

---

---

---

Director \_\_\_\_\_

**Mg. Favián Arenas Aparicio**

Jurado \_\_\_\_\_

**Dra. Rosana Pérez**

Jurado \_\_\_\_\_

**Mg. Mauricio Macías Caicedo**

**Fecha de sustentación: 10 de septiembre de 2018, Popayán.**

# Índice general

<b>Índice general</b>	<b>4</b>
<b>1. Introducción</b>	<b>6</b>
<b>2. Métodos iterativos</b>	<b>8</b>
2.0.1. Método de <i>Jacobi</i> . . . . .	8
2.0.2. Método de <i>Gauss-Seidel</i> . . . . .	10
2.0.3. Método de los gradientes conjugados . . . . .	11
<b>3. Método de <i>Newton</i> inexacto</b>	<b>14</b>
3.0.1. Criterio de parada . . . . .	15
3.0.2. Teoría de convergencia . . . . .	16
<b>4. Pruebas Numéricas</b>	<b>20</b>
<b>5. Conclusiones</b>	<b>31</b>

<b>ÍNDICE GENERAL</b>	<b>5</b>
-----------------------	----------

---

<b>Bibliografía</b>	<b>32</b>
---------------------	-----------

# Capítulo 1

## Introducción

Dada la función  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , definida por  $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))^T$  continuamente diferenciable.

Un sistema de ecuaciones no lineales es el conjunto de ecuaciones,

$$\begin{cases} f_1(\mathbf{x}) = 0 \\ f_2(\mathbf{x}) = 0 \\ \vdots \\ f_n(\mathbf{x}) = 0, \end{cases}$$

o de manera compacta

$$F(\mathbf{x}) = 0. \tag{1.1}$$

Resolver un sistema de ecuaciones no lineales consiste en encontrar, de ser posible, un vector  $\mathbf{x} \in \mathbb{R}^n$  que sea solución de la ecuación (1.1).

Estabilidad de aeronaves, ecuaciones de transferencia radiactiva, problemas elípticos de valor de frontera, potencia de flujos, discretización de problemas de evolución, equilibrio de plantas químicas, entre otros [7], son los diferentes campos donde podemos encontrar sistemas de ecuaciones no lineales, de aquí la importancia de encontrar una forma eficiente de resolver estos sistemas de ecuaciones no lineales.

Métodos como el de *Newton* y *cuasi-Newton* [1] son muy utilizados por su eficiencia y

precisión a la hora de encontrar la solución partiendo de un punto inicial  $\mathbf{x}_0$  cercano a la solución; sin embargo resolver un sistema de ecuaciones lineales en cada etapa del proceso es muy costoso cuando el número de variables es grande y puede suceder que la sucesión de vectores que genera el método no se aproxime a la solución, por esto consideraremos el método de *Newton* inexacto [2], que también resuelve este tipo de sistemas. Este tipo de método puede ser tan eficiente como los métodos antes mencionados y puede disminuir el costo computacional. En esto se enfoca el estudio numérico que presento a continuación.

En el capítulo siguiente se definen los métodos iterativos y cuáles son los que se usaron en este trabajo para realizar las comparaciones, en el tercer capítulo explicaré en qué consiste el método de *Newton* inexacto, las condiciones que deben satisfacer los sistemas no lineales para garantizar que converja. En el capítulo cuatro se realizarán pruebas numéricas con el fin de verificar su eficiencia y hacer un comparativo con el método *Newton*, que soluciona de manera precisa estos sistemas (1.1) y así establecer las condiciones en las que se debe usar estos métodos para encontrar la mejor solución. Las conclusiones.

# Capítulo 2

## Métodos iterativos

En la solución de sistemas de ecuaciones lineales los métodos iterativos tratan de resolverlo mediante aproximaciones sucesivas, partiendo de un punto o aproximación inicial. Estos métodos contrastan con los métodos directos que buscan la solución de los sistemas de ecuaciones lineales en un número finito de pasos.

Los métodos iterativos que consideramos en este trabajo son los métodos de *Jacobi*, *Gauss-Seidel* y el método de los gradientes conjugados los cuales describo a continuación.

### 2.0.1. Método de *Jacobi*

El método de *Jacobi* es un método iterativo que nos permite resolver sistemas de ecuaciones lineales  $A\mathbf{x} = \mathbf{b}$  y consiste en generar una sucesión  $\{\mathbf{x}_n\}$  que en determinadas condiciones los puntos mejoran la aproximación a la solución del sistema.

El método de *Jacobi* converge siempre que la matriz  $A$  sea estrictamente diagonal dominante es decir, el valor absoluto del elemento de la diagonal de esa fila es estrictamente mayor que la norma del resto de elementos de la misma fila, por lo tanto, si cumple esta condición se garantiza que la matriz  $A$  no tenga elementos nulos en la diagonal ( $D$ ), notemos que por la afirmación anterior,  $D$  es una matriz no singular o es invertible.

Para comenzar se debe descomponer la matriz  $A$  en dos matrices  $D$  y  $R$ , donde la primera representa la matriz conformada por los elementos de la diagonal y  $R$  es la matriz

resultante de la resta  $A - D$ , es decir  $A = D + R$  al remplazar dicha descomposición se obtiene lo siguiente,

$$D\mathbf{x} + R\mathbf{x} = \mathbf{b},$$

luego,

$$\mathbf{x} = D^{-1}(\mathbf{b} - R\mathbf{x})$$

Así, las iteraciones de este método son de la forma

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - R\mathbf{x}_k)$$

Veamos el siguiente sistema de ecuaciones lineales,

$$\begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix} \quad (2.1)$$

Resolveremos el sistema (2.1) por el método de *Jacobi* con los siguientes datos,

$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}, \quad A = \begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix} \quad (2.2)$$

Por el método de *Jacobi* tenemos que la primera iteración es la siguiente

$$\mathbf{x}_1 = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.090909 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.125 \end{pmatrix} \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 2.27273 \\ -1.1 \\ 1.875 \end{pmatrix}$$

Ahora, la segunda iteración por el método de *Jacobi* es:

$$\mathbf{x}_2 = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.090909 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.125 \end{pmatrix} \begin{pmatrix} 8.9455 \\ -12.25 \\ 5.8955 \\ -15.8364 \end{pmatrix} = \begin{pmatrix} 0.89455 \\ -1.97955 \\ 0.58 \\ -1.97955 \end{pmatrix}$$

Para la tercer iteración por el método de *Jacobi* tenemos que:

$$\mathbf{x}_3 = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.090909 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.125 \end{pmatrix} \begin{pmatrix} -7.5309 \\ 52.0955 \\ -26.66 \\ -38.6973 \end{pmatrix} = \begin{pmatrix} -0.75309 \\ 4.73595 \\ -2.666 \\ 4.83716 \end{pmatrix}$$

Continuando con el proceso encontramos la solución del sistema (2.1) en la iteración 18,

$$\mathbf{x}_{18} = \begin{pmatrix} -1.0 \\ -2.0 \\ 1.0 \\ -1 \end{pmatrix}.$$

## 2.0.2. Método de *Gauss-Seidel*

El método de *Gauss-Seidel* es un método iterativo que al igual que el método de *Jacobi*, nos permite resolver sistemas de ecuaciones lineales  $A\mathbf{x} = \mathbf{b}$ . A partir de una aproximación inicial, este proceso se repite hasta llegar a la solución del sistema, con un margen de error pequeño, tanto como se quiera [1].

De manera similar que en el método de *Jacobi*, se debe descomponer la matriz  $A$  de la siguiente manera  $A = D - L - U$ , donde  $D$  es la matriz diagonal,  $L$  es la matriz triangular inferior y  $U$  es la matriz triangular superior, al igual que en el método de *Jacobi*  $D$  es no singular y  $D - L$  también lo es, de aquí puede deducirse que

$$\mathbf{x}_{k+1} = (D - L)^{-1}U\mathbf{x}_k + (D - L)^{-1}\mathbf{b}.$$

Si llamamos  $M = (D - L)^{-1}U$  y  $\mathbf{c} = (D - L)^{-1}\mathbf{b}$  el sistema anterior se escribe de la siguiente manera,

$$\mathbf{x}_{k+1} = M^{-1}\mathbf{x}_k + \mathbf{c}.$$

La convergencia del método se garantiza siempre que  $A$  sea diagonalmente dominante en el sentido estricto filas<sup>1</sup> o si la matriz es simétrica<sup>2</sup> y definida positiva<sup>3</sup>.

<sup>1</sup>una matriz es diagonalmente dominante en el sentido estricto por filas si cumple que, el valor absoluto del elemento de la diagonal de esa fila es estrictamente mayor que la suma de las normas del resto de elementos de esa fila.

<sup>2</sup>Una matriz es simétrica si es una matriz cuadrada, la cual tiene la característica de ser igual a su traspuesta.

<sup>3</sup>Una matriz definida positiva es una matriz hermitiana que en muchos aspectos es similar a un

Resolvamos el sistema (2.1) con  $\mathbf{x}_0$ ,  $\mathbf{b}$  y  $A$  descritos en (2.2), la primera iteración mediante el método de *Gauss Seidel* es:

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.6 \\ 2.21818 \\ -1.20182 \\ 2.85705 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 2.21818 \\ -1.20182 \\ 2.85705 \end{pmatrix}.$$

Ahora, la segunda iteración por el método de *Jacobi* es:

$$\mathbf{x}_2 = \begin{pmatrix} -0.46218 \\ 0.93047 \\ -0.47119 \\ 0.40782 \end{pmatrix} + \begin{pmatrix} 0.46218 \\ -0.82138 \\ 0.66391 \\ -2.35510 \end{pmatrix} = \begin{pmatrix} -1.11e - 007 \\ 1.0909e - 001 \\ 1.9273e - 001 \\ -1.9473 + 000 \end{pmatrix}.$$

El proceso termina en la séptima iteración,

$$\mathbf{x}_7 = \begin{pmatrix} -1.0 \\ -2.0 \\ 1.0 \\ -1 \end{pmatrix}.$$

### 2.0.3. Método de los gradientes conjugados

Es un método iterativo usado para resolver sistemas de ecuaciones lineales  $A\mathbf{x} = \mathbf{b}$ , para aplicar este método la matriz  $A$  debe ser simétrica y definida positiva y se hace necesario construir una sucesión  $\{\mathbf{r}_k\}$  de vectores llamados vectores residuo  $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$  para formar un conjunto *A-ortogonales*<sup>4</sup>, según el producto interno  $\langle x, y \rangle = x^T Ay$ .

El método de los gradientes conjugados comienza con una aproximación inicial  $\mathbf{x}_0$  cualquiera y definimos el primer residuo como  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  y llamamos  $d_0 = \mathbf{r}_0$  donde  $d$  son las direcciones de búsqueda para la solución del sistema. En el paso *k-ésimo* se han

---

número real positivo, también puede tratarse de una matriz simétrica real cuyos menores principales son positivos (Criterio de Sylvester)

<sup>4</sup>Dada una matriz  $A$  de dimensión  $n \times n$ ,  $u_i$  y  $u_j$  dos vectores tales que  $u_i^T Au_j = 0$  entonces se dice que  $u_i, u_j$  son ortogonales respecto a la matriz  $A$  o *A-ortogonal*

definido  $\mathbf{x}_k, \mathbf{r}_k$  y  $d_k$  para construir el  $k+1$ -ésimo término así,

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_k^T A d_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k d_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A d_k$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}$$

$$d_{k+1} = \mathbf{r}_{k+1} + \beta_k d_k$$

Si el residuo es cero, se tiene la solución del sistema, es decir,  $\mathbf{r}_k = 0$  si, y solo si  $\mathbf{x}_k = \mathbf{x}_*$ .

El método de los gradientes conjugados nos permite obtener la solución exacta en un máximo de  $n$  iteraciones, siempre y cuando el sistema tenga solución, donde  $n$  es el número de ecuaciones del sistema.

Resolveremos el sistema (2.1) mediante el método de los gradientes conjugados. A continuación tenemos los resultados de la primera iteración.

$$\mathbf{r}_0 = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}.$$

Luego, hacemos para el primer paso,  $\mathbf{r}_0 = d_0$

$$d_0 = \begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}.$$

Ahora calculamos  $\alpha_0, \mathbf{x}_1, \mathbf{r}_1, \beta_0, d_1$

$$\alpha_0 = 0.078604$$

$$\mathbf{x}_1 = \begin{pmatrix} 0.47163 \\ 1.96511 \\ -0.8645 \\ 1.17906 \end{pmatrix}.$$

$$\mathbf{r}_1 = \begin{pmatrix} 4.97814 \\ -0.54641 \\ -0.1526 \\ -1.19249 \end{pmatrix}.$$

$$\beta_0 = 0.026341$$

$$\mathbf{d}_1 = \begin{pmatrix} 5.13619 \\ 0.11213 \\ -0.44236 \\ -0.79737 \end{pmatrix}.$$

En este caso el método de los gradientes conjugados encontró la solución en solo un paso, como se dijo anteriormente una ventaja de este método es que si el sistema converge, la solución se logra en no más de  $n$  pasos, donde  $n$  es la dimensión del problema.

# Capítulo 3

## Método de *Newton* inexacto

El método de *Newton* es un algoritmo que genera una sucesión  $\{\mathbf{x}_k\}$  que bajo ciertas hipótesis converge a un vector  $\mathbf{x}_*$  el cual es solución de  $F(\mathbf{x}) = 0$ .

Para deducir este algoritmo, consideremos la aproximación de *Taylor* de primer orden de  $F$  en un punto  $\mathbf{x}$

$$F(\mathbf{x}) \approx L_k(\mathbf{x}) = F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k),$$

donde  $J(\mathbf{x})$  es la matriz jacobiana de  $F$  en  $\mathbf{x}$ , la cual se define por

$$J(\mathbf{x}) = \begin{bmatrix} \nabla f_1(\mathbf{x})^T \\ \vdots \\ \nabla f_n(\mathbf{x})^T \end{bmatrix}. \quad (3.1)$$

Resolver

$$L_k(\mathbf{x}) = 0,$$

es equivalente a

$$F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) = 0.$$

Así, una iteración del método de *Newton* se describe por:

$$J(\mathbf{x}_k)\mathbf{s}_k = -F(\mathbf{x}_k) \quad (3.2)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \quad (3.3)$$

En el método de *Newton* resolver el problema descrito en (1.1) se reduce a solucionar en cada paso un sistema de ecuaciones lineales, el cual se resuelve por eliminación gaussiana. Se ha demostrado que el método de *Newton* converge y si es  $F$  *Lipschitz* continua lo hace cuadráticamente [3].

En el método de *Newton* inexacto buscamos como alternativa para resolver sistemas de ecuaciones lineales los métodos iterativos, que como mencionamos anteriormente no buscan la solución de forma directa sino que la buscan de forma inexacta, aun así, son bastante eficientes en cuanto almacenamiento y a tiempo de computo se refiere; generalmente los métodos iterativos para resolver sistemas de ecuaciones lineales basados en subespacios de *Krylov*<sup>1</sup> en la búsqueda de vectores y valores propios. También en la búsqueda de soluciones de un sistema de ecuaciones lineales [4] [5] [10], en el desarrollo de las pruebas numéricas que mas adelante mostraremos se usará los métodos iterativo anteriormente mencionados.

### 3.0.1. Criterio de parada

Los métodos iterativos usados para resolver sistemas de ecuaciones lineales no satisfacen necesariamente la condición (3.2), por lo tanto necesitan un criterio de parada para decidir que se tiene una solución aproximada del sistema lineal [2].

Para el método de *Newton* inexacto se requiere encontrar un  $\mathbf{s}_k$  que satisfaga lo siguiente,

$$\| J(\mathbf{x}_k)\mathbf{s}_k + F(\mathbf{x}_k) \| \leq \theta_k \| F(\mathbf{x}_k) \| \quad (3.4)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k \quad (3.5)$$

con  $\theta_k \in (0, 1)$ .

Para lograr la convergencia del método de *Newton* inexacto generamos una sucesión de puntos  $\{\mathbf{x}_k\}$  que satisfaga la condición (3.4), si bien no cumple la condición del método de *Newton* tradicional, este criterio nos permite garantizar que los puntos que se generan se

<sup>1</sup>un subespacio de *Krylov* de orden  $r$  generado por una matriz cuadrada  $A$  de orden  $n$  y un vector  $\mathbf{v}$ , es el subespacio vectorial generado por  $A^k\mathbf{v}$  donde  $k < r$  [11].

acerquen a la solución del sistema. A continuación veremos un teorema que nos permite garantizar la convergencia del método de *Newton* inexacto.

### 3.0.2. Teoría de convergencia

En [2] se demuestra que bajo ciertas hipótesis, un algoritmo basado en el método de Newton inexacto es convergente, esta demostración la desarrollaré con más detalle en el **Teorema 3.1**, pero antes es necesario plantear algunos lemas auxiliares. Supongamos (1.1) con las siguientes hipótesis:

**H1:** Existe un  $\mathbf{x}_*$  tal que  $F(\mathbf{x}_*) = 0$ .

**H2:**  $F$  es continuamente diferenciable en un vecindad de  $\mathbf{x}_*$ .

**H3:**  $J(\mathbf{x}_*)$  es no singular.

**Lema 3.0.1.** *Supongamos **H1**, **H3** y que para todo  $\varepsilon > 0$ , existe  $\delta > 0$  tal que  $J(\mathbf{y})$  es no singular y además, si  $\|\mathbf{y} - \mathbf{x}_*\| < \delta$ , entonces  $\|J(\mathbf{y})^{-1} - J(\mathbf{x}_*)^{-1}\| < \varepsilon$  [2].*

En el **lema 3.0.1** si los puntos de la sucesión  $\{\mathbf{x}_k\}$  se acercan a la solución  $\mathbf{x}_*$  entonces sus respectivos jacobianos también lo hacen.

**Lema 3.0.2.** *Supongamos **H1** y para todo  $\varepsilon > 0$ , existe  $\delta > 0$  tal que, si  $\|\mathbf{y} - \mathbf{x}_*\| < \delta$ , entonces  $\|F(\mathbf{y}) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{y} - \mathbf{x}_*)\| \leq \varepsilon \|\mathbf{y} - \mathbf{x}_*\|$  [2].*

En el **lema 3.0.2** tenemos que bajo la hipótesis que los puntos están cerca de la solución, entonces la aproximación de la función es casi lineal.

**Lema 3.0.3.** *Supongamos **H1** y **H3**; sea  $\|\cdot\|$  la norma euclidiana y consideremos la norma  $\|\cdot\|_*$  dada por  $\|\mathbf{y}\|_* := \|J(\mathbf{x}_*)\mathbf{y}\|$  entonces existe  $\mu$  constante, tal que verifiquen la desigualdad*

$$\frac{1}{\mu} \|\mathbf{y}\| \leq \|\mathbf{y}\|_* \leq \mu \|\mathbf{y}\| .$$

*Demostración.* Sea  $\mathbf{y} \in \mathbb{R}^n$ , como  $J(\mathbf{x}_*)$  es no singular y por la definición de  $\|\cdot\|_*$  obtenemos la siguiente desigualdad y con  $\|J(\mathbf{x}_*)\| = \mu_1$

$$\begin{aligned} \|\mathbf{y}\|_* &= \|J(\mathbf{x}_*)\mathbf{y}\| \\ &\leq \|J(\mathbf{x}_*)\| \|\mathbf{y}\| \\ &= \mu_1 \|\mathbf{y}\| . \end{aligned} \tag{3.6}$$

Por otra parte, supongamos  $\|J(\mathbf{x}_*^{-1})\| = \mu_2$ ,

$$\begin{aligned}\|\mathbf{y}\| &= \|J(\mathbf{x}_*)^{-1}J(\mathbf{x}_*)\mathbf{y}\| \\ &\leq \|J(\mathbf{x}_*)^{-1}\| \|J(\mathbf{x}_*)\mathbf{y}\| \\ &= \|J(\mathbf{x}_*)^{-1}\| \|\mathbf{y}\|_* \\ &= \mu_2 \|\mathbf{y}\|_*.\end{aligned}\tag{3.7}$$

Luego,

$$\frac{1}{\mu_2} \|\mathbf{y}\| \leq \|\mathbf{y}\|_* \leq \mu_1 \|\mathbf{y}\|.$$

Ahora solo basta tomar  $\mu := \max\{\mu_1, \mu_2\}$  tal que,

$$\frac{1}{\mu} \|\mathbf{y}\| \leq \frac{1}{\mu_2} \|\mathbf{y}\| \leq \|\mathbf{y}\|_* \leq \mu_1 \|\mathbf{y}\| \leq \mu \|\mathbf{y}\|.$$

Esto significa:

$$\frac{1}{\mu} \|\mathbf{y}\| \leq \|\mathbf{y}\|_* \leq \mu \|\mathbf{y}\|.\tag{3.8}$$

◆

A continuación veremos el teorema que nos permite garantizar la convergencia del método bajo ciertas condiciones.

**Teorema 3.1.** *Supongamos **H1**, **H2** y **H3**; Sea  $0 < \theta_k \leq \theta_{\max} < \theta < 1$  entonces existe un  $\varepsilon > 0$  tal que si  $\|\mathbf{x}_0 - \mathbf{x}_*\| \leq \varepsilon$  la sucesión obtenida en (3.2) y (3.4) converge a  $\mathbf{x}_*$  y satisface*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_* \leq \theta^k \|\mathbf{x}_k - \mathbf{x}_*\|_*\tag{3.9}$$

*Demostración.* (ver [2])

La siguiente demostración se realiza mediante inducción matemática para verificar (3.9) y con **lema 3.0.1** y **lema 3.0.2** se garantiza la existencia de un número tan pequeño como se desea para verificar la desigualdad y así demostrar el **Teorema 3.1**, pero para todos los casos el proceso es similar, por lo tanto asumiremos como hipótesis inductiva que, para todo  $n \leq k$  se tiene la siguiente desigualdad,

$$\|\mathbf{x}_n - \mathbf{x}_*\|_* \leq \theta \|\mathbf{x}_{n-1} - \mathbf{x}_*\|_*$$

Supongamos que  $\| \mathbf{x}_0 - \mathbf{x}_* \| \leq \varepsilon$ , notemos que por la parte izquierda de la desigualdad dada en (3.8)

$$\| \mathbf{x}_k - \mathbf{x}_* \| \leq \mu \| \mathbf{x}_k - \mathbf{x}_* \|_*.$$

Por la hipótesis inductiva aplicada  $k$ -veces,

$$\| \mathbf{x}_k - \mathbf{x}_* \|_* \leq \theta^k \| \mathbf{x}_0 - \mathbf{x}_* \|_*.$$

Por lo tanto tenemos que,

$$\mu \| \mathbf{x}_k - \mathbf{x}_* \|_* \leq \mu \theta^k \| \mathbf{x}_0 - \mathbf{x}_* \|_*.$$

Dado que  $\theta < 1$  y además por (3.6), entonces tenemos que,

$$\mu \theta^k \| \mathbf{x}_0 - \mathbf{x}_* \|_* \leq \mu^2 \| \mathbf{x}_0 - \mathbf{x}_* \|.$$

Así, por el supuesto de  $\| \mathbf{x}_0 - \mathbf{x}_* \| \leq \varepsilon$  podemos concluir que,

$$\| \mathbf{x}_k - \mathbf{x}_* \| \leq \mu \| \mathbf{x}_k - \mathbf{x}_* \|_* \leq \mu \theta^k \| \mathbf{x}_0 - \mathbf{x}_* \|_* \leq \mu^2 \| \mathbf{x}_0 - \mathbf{x}_* \| \leq \mu^2 \varepsilon.$$

Luego para todo  $\varepsilon > 0$ , existe un  $\gamma > 0$  tal que, si  $\| \mathbf{x}_k - \mathbf{x}_* \| \leq \mu^2 \varepsilon$ , entonces por los **Lemas 3.0.1** y **3.0.2** se cumple las siguientes desigualdades

$$\| J(\mathbf{x}_k) - J(\mathbf{x}_*) \| \leq \gamma \tag{3.10}$$

$$\| J(\mathbf{x}_k)^{-1} - J(\mathbf{x}_*)^{-1} \| \leq \gamma \tag{3.11}$$

$$\| F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*) \| \leq \gamma \| \mathbf{x}_k - \mathbf{x}_* \| \tag{3.12}$$

Supongamos,  $J(\mathbf{x}_k)\mathbf{s}_k + F(\mathbf{x}_k) = \mathbf{r}_k$  y notemos que,

$$\begin{aligned} J(\mathbf{x}_*)(\mathbf{x}_{k+1} - \mathbf{x}_*) &= J(\mathbf{x}_*)[\mathbf{s}_k + \mathbf{x}_k - \mathbf{x}_*] \\ &= J(\mathbf{x}_*)J(\mathbf{x}_k)^{-1}[J(\mathbf{x}_k)(\mathbf{s}_k + \mathbf{x}_k - \mathbf{x}_*)] \\ &= J(\mathbf{x}_*)J(\mathbf{x}_k)^{-1}[J(\mathbf{x}_k)\mathbf{s}_k + J(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_*)] \\ &= [J(\mathbf{x}_*)J(\mathbf{x}_k)^{-1}] \cdot [J(\mathbf{x}_k)\mathbf{s}_k + F(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_*) - F(\mathbf{x}_k) - F(\mathbf{x}_*)] \\ &= [J(\mathbf{x}_*)J(\mathbf{x}_k)^{-1}] \cdot [\mathbf{r}_k + [J(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_*) - F(\mathbf{x}_k) - F(\mathbf{x}_*)]] \\ &= [I + J(\mathbf{x}_*)[J(\mathbf{x}_k)^{-1} - J(\mathbf{x}_*)^{-1}]] \cdot [\mathbf{r}_k + [J(\mathbf{x}_k) - J(\mathbf{x}_*)](\mathbf{x}_k - \mathbf{x}_*) \\ &\quad - [F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)]]. \end{aligned}$$

Por lo tanto,

$$\begin{aligned} J(\mathbf{x}_*)(\mathbf{x}_{k+1} - \mathbf{x}_*) &= [I + J(\mathbf{x}_*)[J(\mathbf{x}_k)^{-1} - J(\mathbf{x}_*)^{-1}]] \cdot [\mathbf{r}_k + [J(\mathbf{x}_k) - J(\mathbf{x}_*)](\mathbf{x}_k - \mathbf{x}_*)] \\ &\quad - [F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)]. \end{aligned} \quad (3.13)$$

De igual forma tenemos la siguiente igualdad,

$$F(\mathbf{x}_k) = [J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)] + [F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)]. \quad (3.14)$$

Tomando normas en (3.15) y haciendo uso de (3.12) tenemos que,

$$\begin{aligned} \|F(\mathbf{x}_k)\| &\leq \|\mathbf{x}_k - \mathbf{x}_*\|_* + \|F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)\| \\ &\leq \|\mathbf{x}_k - \mathbf{x}_*\|_* + \gamma \|\mathbf{x}_k - \mathbf{x}_*\|, \end{aligned} \quad (3.15)$$

Ahora de la igualdad (3.13) y usando la definición de  $\mu$ , (3.10), (3.11), (3.12) y (3.15),

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_*\|_* &\leq [1 + \|J(\mathbf{x}_*)\| \|J(\mathbf{x}_k)^{-1} - J(\mathbf{x}_*)^{-1}\|] \cdot [\|\mathbf{r}_k\| + \|J(\mathbf{x}_k) - J(\mathbf{x}_*)\| \\ &\quad \cdot \|\mathbf{x}_k - \mathbf{x}_*\| + \|F(\mathbf{x}_k) - F(\mathbf{x}_*) - J(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)\|] \\ &\leq (1 + \mu\gamma)[\theta_k \|F(\mathbf{x}_k)\| + \gamma \|\mathbf{x}_k - \mathbf{x}_*\| + \gamma \|\mathbf{x}_k - \mathbf{x}_*\|] \\ &\leq (1 + \mu\gamma)[\theta_k \|\mathbf{x}_k - \mathbf{x}_*\|_* + \gamma \|\mathbf{x}_k - \mathbf{x}_*\|] + 2\gamma \|\mathbf{x}_k - \mathbf{x}_*\|. \end{aligned}$$

Ahora tomando de la sucesión  $\{\theta_k\}$  el valor máximo podemos concluir que,

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\|_* \leq (1 + \mu\gamma)[\theta_{max}(1 + \mu\gamma) + 2\gamma\mu] \|\mathbf{x}_k - \mathbf{x}_*\|_*.$$

Por la elección de  $\gamma$  y la propiedad arquimediana, podemos garantizar que existen

$$(1 + \gamma\mu)[\theta_{m\acute{a}x}(1 + \mu\gamma) + 2\mu\gamma] \leq \theta$$

tal que cumple las hipótesis del teorema, por lo tanto queda demostrado la convergencia lineal de la sucesión.  $\blacklozenge$

El teorema anterior, asegura que existe una vecindad en la que al elegir el  $\mathbf{x}_0$  se puede garantizar que el método converge al menos linealmente. Convirtiendo el método de *Newton* inexacto en una alternativa local, algo diferente.

Ahora, ¿Cómo es el comportamiento numérico de este método comparado con *Newton* tradicional? el siguiente capítulo realizamos la comparación.

# Capítulo 4

## Pruebas Numéricas

En este capítulo analizaremos la eficiencia del método de *Newton* inexacto y realizamos una comparación con el método de *Newton* local para encontrar la solución de los sistemas de ecuaciones no lineales. La comparación se realizará entre la eliminación gaussiana (método directo) y *Jacobi*, *Gauss-Seidel*, gradiente conjugado (métodos iterativos), descritos en el capítulo anterior.

A continuación describo el algoritmo de *Newton*, que usamos para realizar las pruebas numéricas y las comparaciones.

**Algoritmo 4.1.** (*Newton*)

Dado  $\mathbf{x}_0$  y  $\theta \in (0, 1)$ , para  $k=1, 2, \dots$

Mientras  $\|F(\mathbf{x})\| > 10^{-5}$  y  $k < N$

Calculamos  $\mathbf{b} = -F(\mathbf{x}_k)$

Calculamos  $A = F'(\mathbf{x}_k)$

Resolver el sistema lineal  $A\mathbf{s}_k - \mathbf{b} = 0$  (*Eliminación Gaussiana*)

Calcular la iteración

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$$

$$k \leftarrow k + 1$$

*fin*

A continuación presentamos el algoritmo de *Newton* inexacto que implementamos en este trabajo, el algoritmo *solución lineal* se presenta más adelante en el **algoritmo 4.3**.

**Algoritmo 4.2.** (*Newton Inexacto*)

Dado  $\mathbf{x}_0$  y  $\theta \in (0, 1)$ , para  $k=1,2,\dots$

Mientras  $\|F(\mathbf{x})\| > 10^{-5}$  y  $k < N$

$$\text{Calculamos } \mathbf{b} = -F(x_k)$$

$$\text{Calculamos } A = F'(x_k)$$

Resolver el sistema lineal  $A\mathbf{s}_k - \mathbf{b} = 0$  (*solución lineal*)

Calcular la iteración

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$$

$$k \leftarrow k + 1$$

*fin*

A continuación describo el algoritmo de cada uno de los métodos iterativos que usamos *Newton* inexacto para compararlo con el método de *Newton* tradicional.

**Algoritmo 4.3.** (*solución lineal*)

Dado  $A$ ,  $\mathbf{b}$  y  $tol$ ,  $\mathbf{x}_0$  para  $k=1,2,\dots$

**Caso 1** (*Jacobi*)

Calculamos  $D$  (*diagonal de A*),  $R = A - D$

Mientras  $\|A\mathbf{x}_k - \mathbf{b}\| > tol$  y  $k < N$

$$\mathbf{x}_{k+1} = D^{-1}(\mathbf{b} - R\mathbf{x}_k)$$

$$k \leftarrow k + 1$$

*fin*

**Caso 2** (*Gauss Seidel*)

Calculamos  $D$  (diagonal de  $A$ ),  $L$  (triangular inferior de  $A$ ),  $U$  (triangular superior de  $A$ )

Mientras  $\|A\mathbf{x}_k - \mathbf{b}\| > \text{tol}$  y  $k < N$

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

$$\text{Definimos } M = (D - L)^{-1}U \text{ y } \mathbf{c} = (D - L)^{-1}\mathbf{b}$$

$$\mathbf{x}_{k+1} = M^{-1}\mathbf{x}_k + \mathbf{c}$$

$$k \leftarrow k + 1$$

*fin*

**Caso 3** (*Gradiente Conjugado*)

Calculamos  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  y  $d_0 = \mathbf{r}_0$

Mientras  $\|A\mathbf{x}_k - \mathbf{b}\| > \text{tol}$  y  $k < N$

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{d}_k^T A \mathbf{d}_k}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k d_k$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A d_k$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}$$

$$d_{k+1} = \mathbf{r}_{k+1} + \beta_k d_k$$

$$k \leftarrow k + 1$$

*fin*

Para escribir los códigos de los algoritmos y de las funciones de prueba, usamos el lenguaje

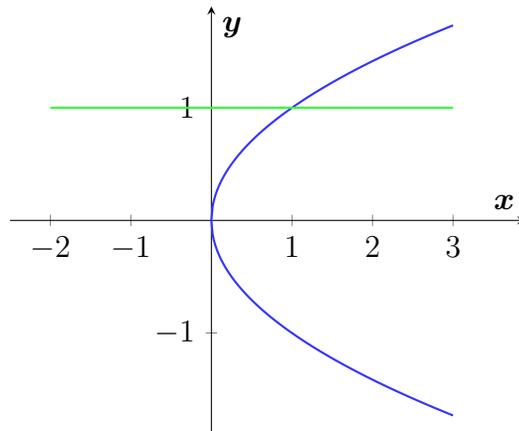
MATLAB. Las pruebas numéricas se realizaron en un computador con procesador intel CORE i5 de 2.6 GHz.

Se utilizaron 10 problemas para probar el algoritmo de *Newton* inexacto; *Rosenbrock*, *Valle Helicoidal*, *Singular Powell* y *Freundenstein y Roth* se tomaron de una lista de funciones [8], las demás fueron tomados de los ejercicios propuestos y ejemplos en [1], [9]. Para cada problema describimos su punto inicial ( $\mathbf{x}_0$ ) y también la solución ( $\mathbf{x}_*$ ) de cada sistema.

1. **Función Rosenbrock** Sea  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

$$\begin{bmatrix} 10(x_2 - x_1^2) \\ 1 - x_1 \end{bmatrix} \quad (4.1)$$

con vector inicial  $\mathbf{x}_0 = (-1.2, 1)^T$  y  $\mathbf{x}_* = (1, 1)^T$ .



2. **Función Valle Helicoidal** Sea  $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  definida por

$$\begin{bmatrix} 10(x_3 - \frac{5}{\pi} \arctan \frac{x_2}{x_1} - \theta) \\ 10(\sqrt{x_1^2 + x_2^2} - 1) \\ x_3 \end{bmatrix} \quad (4.2)$$

con vector inicial  $\mathbf{x}_0 = (-1, 0, 0)^T$ . y  $\mathbf{x}_* = (1, 0, 0)^T$ .

3. **Función Singular Powell Sea**  $F : \mathbb{R}^4 \rightarrow \mathbb{R}^4$  definida por

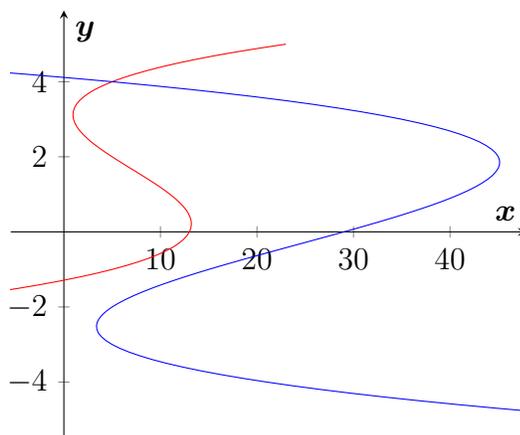
$$\begin{bmatrix} x_1 + 10x_2 \\ \sqrt{5}(x_3 - x_4) \\ (x_2 - 2x_3)^2 \\ \sqrt{10}(x_1 - x_4)^2 \end{bmatrix} \quad (4.3)$$

con vector inicial  $\mathbf{x}_0 = (3, -1, 0, 1)^T$  y  $\mathbf{x}_* = (0, 0, 0, 0)^T$ .

4. **Función Freudenstein y Roth Sea**  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

$$\begin{bmatrix} x_1 - x_2(2 - x_2(5 - x_2)) - 13 \\ x_1 - x_2(14 - x_2(1 + x_2)) - 29 \end{bmatrix} \quad (4.4)$$

con vector inicial  $\mathbf{x}_0 = (0.5, -2)^T$  y  $\mathbf{x}_* = (5, 4)^T$ .



5. **Función Ejercicio 1 Sea**  $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  definida por

$$\begin{bmatrix} 3x_1 - \cos(x_2x_3) - \frac{1}{2} \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 \\ e^{-x_1x_2} + 20x_3 + \left(\frac{10\pi-3}{3}\right) \end{bmatrix} \quad (4.5)$$

con vector inicial  $\mathbf{x}_0 = (0.1, 0.1, -0.1)^T$  y  $\mathbf{x}_* = (5.00e-001, 1.37e-010, -5.23e-001)^T$ .

6. **Función Ejercicio 2** Sea  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

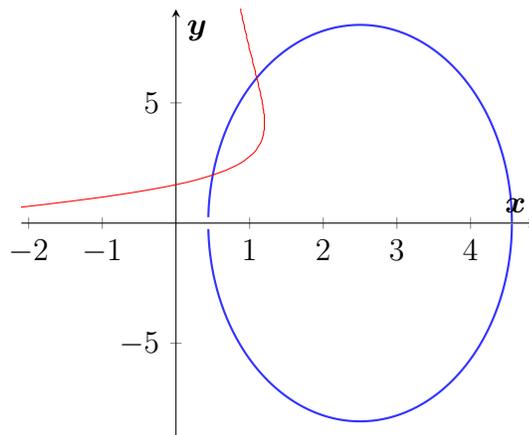
$$\begin{bmatrix} 4x_1^2 - 20x_1 + \frac{1}{4}x_2^2 + 8 \\ \frac{1}{2}x_1x_2^2 + 2x_1 - 5x_2 + 8 \end{bmatrix} \quad (4.6)$$

con vector inicial  $\mathbf{x}_0 = (0.1, 0.1, -0.1)^T$  y  $\mathbf{x}_* = (0.5, 2)^T$ .

6. **Función Ejercicio 2** Sea  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

$$\begin{bmatrix} 4x_1^2 - 20x_1 + \frac{1}{4}x_2^2 + 8 \\ \frac{1}{2}x_1x_2^2 + 2x_1 - 5x_2 + 8 \end{bmatrix} \quad (4.7)$$

con vector inicial  $\mathbf{x}_0 = (0.1, 0.1, -0.1)^T$  y  $\mathbf{x}_* = (0.5, 2)^T$ .



7. **Función Ejercicio 3** Sea  $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  definida por

$$\begin{bmatrix} 15x_1 + x_2^2 - 4x_3 - 13 \\ x_1^2 + 10x_2 - x_3 - 11 \\ x_2^3 - 25x_3 + 22 \end{bmatrix} \quad (4.8)$$

con vector inicial  $\mathbf{x}_0 = (0, 0, 0)^T$  y  $\mathbf{x}_* = (1.036, 1.085, 0.931)^T$ .

8. **Función Ejercicio 4** Sea  $F : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  definida por

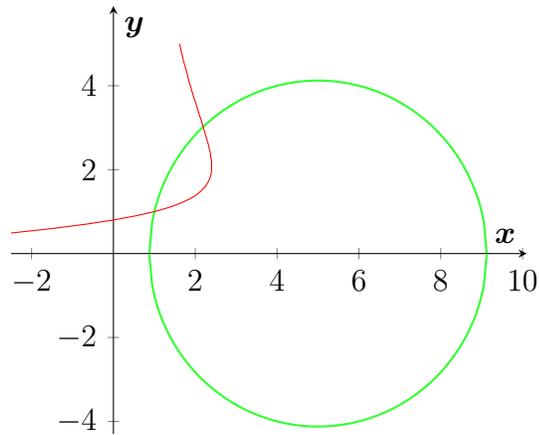
$$\begin{bmatrix} 3x_1 - \cos(x_2x_3) - 0.5 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 \\ e^{-x_1x_2} + 20x_3 + \frac{10\pi-3}{3} \end{bmatrix} \quad (4.9)$$

con vector inicial  $\mathbf{x}_0 = (0.1, 0.1, -0.1)^T$  y  $\mathbf{x}_* = (5.00e - 001, 7.75e - 010, -5.23e - 001)^T$ .

9. **Función Ejercicio 5** Sea  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

$$\begin{bmatrix} x_1^2 - 10x_1 + x_2^2 + 8 \\ x_1x_2^2 + x_1 - 10x_2 + 8 \end{bmatrix} \quad (4.10)$$

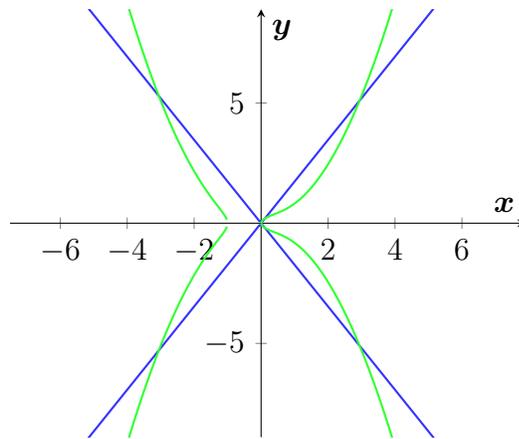
con vector inicial  $\mathbf{x}_0 = (0, 0)^T$  y  $\mathbf{x}_* = (1, 1)^T$ .



10. **Función Ejercicio 6** Sea  $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  definida por

$$\begin{bmatrix} 3x_1^2 - x_2^2 \\ 3x_1x_2^2 - x_1^3 - 1 \end{bmatrix} \quad (4.11)$$

con vector inicial  $\mathbf{x}_0 = (1, 2)^T$  y  $\mathbf{x}_* = (0.5, 0.866)^T$ .



A continuación muestro los resultados obtenidos mediante el método de *Newton* inexacto haciendo usos de los métodos iterativos (**J**, **G.S**, **G.C**) y el método de *Newton* con eliminación gaussian (**E.G**).

**n**: dimensión del problema.

**E.G**: método de eliminación gaussiana.

**J**: método de Jacobi.

**G.S**: método de Gauss Seidel.

**G.C**: método de gradiente conjugado.

Función	n	Método	Iteraciones	$\ F(x)\ $	Tiempo
Rosenbrock	2	E.G	2	0.0000000000	0.0067729950
		J	2	0.0000000000	0.0020051003
		G.S	2	0.0000000000	0.0026731491
		G.C	---	---	---
Valle Helicoidal	3	E.G	12	0.0000000427	0.0080480576
		J	---	---	--
		G.S	---	---	---
		G.C	---	---	---
Singular Powell	4	E.G	11	0.0000030252	0.0105290413
		J	---	---	---
		G.S	---	---	---
		G.C	---	---	---
Freudenstein y Roth	2	E.G	42	0.0000000002	0.0150411129
		J	---	---	---
		G.S	---	---	---
		G.C	---	---	---
Ejercicio 1	3	E.G	5	0.0000000502	0.0040071011
		J	5	0.0000000506	0.0035710335
		G.S	5	0.0000000506	0.0055141449
		G.C	5	0.0000000513	0.0135359764
Ejercicio 2	2	E.G	4	0.0000000030	0.0015048981
		J	4	0.0000000033	0.0035099983
		G.S	4	0.0000000033	0.0040121078
		G.C	18	0.0000091936	0.1639089584
Ejercicio 3	3	E.G	3	0.0000001806	0.0015060902
		J	4	0.0000000000	0.0020351410
		G.S	3	0.0000037218	0.0029900074
		G.c	3	0.0000008872	0.0034971237
Ejercicio 4	3	E.G	4	0.0000000125	0.0020351410
		J	4	0.0000000124	0.0030078888
		G.S	4	0.0000000123	0.0035388470
		G.C	4	0.0000000106	0.0140709877
Ejercicio 5	2	E.G	7	0.0000026874	0.0030419827
		J	7	0.0000026971	0.0060160160
		G.S	7	0.0000026878	0.0075500011
		G.C	---	---	---
Ejercicio 6	2	E.G	5	0.0000008787	0.0030090809
		J	5	0.0000009026	0.0045490265
		G.S	5	0.0000008793	0.0065510273
		G.C	---	---	---

En la tabla anterior se muestra nombre de la función que se utilizó, la dimensión de cada problema, las iteraciones en cada método, el resultado y el tiempo en cada caso, a partir de estos datos nos permitimos hacer la comparación del método de *Newton* y el *Newton* inexacto.

Para las funciones *Valle Helicoidal*, *Singular Powell* y *Freundenstein y Roth*, tomadas del documento [8], notamos que el método de *Newton* es el más apropiado para este tipo de problemas, esto se debe a que el número de condición es demasiado grande, en otras palabras, el sistema es mal condicionando y por ende el método de *Newton* inexacto falló.

En el caso de la función *Rosenbrock*, el método de *Newton* inexacto funcionó bastante bien, excepto el método del gradiente conjugado (**G.C**) que al igual que en las funciones anteriores el número de condición es demasiado grande y por esta razón falla, pero a pesar de esto podemos considerar los otros métodos como una alternativa para buscar la solución del sistema, pues como podemos notar en términos de aproximación e iteraciones es similar al método de *Newton*, pero mejoró en cuanto a tiempo se refiere, es decir que el método de *Newton* inexacto funcionó mejor con *Jacobi* (**J**) y *Gauss Seidel* (**G.S**).

En los resultados obtenidos con las funciones *Ejercicio 5* y *Ejercicio 6* al igual que con la función *Rosenbrock* el método iterativo del gradiente conjugado (**G.C**) falla, pero podemos notar como con los demás métodos la aproximación es bastante buena en comparación al método tradicional de eliminación gaussiana.

Para finalizar este análisis de resultados, las funciones *Ejercicio 1*, *Ejercicio 2*, *Ejercicio 3* y *Ejercicio 4* nos permitieron realizar el comparativo con todos los métodos descritos en este trabajo, antes que nada logramos notar que el método de *Newton* inexacto es el mejor para resolver los sistemas de ecuaciones no lineales, pero no debemos quitar mérito al método de *Newton* inexacto, pues con las pruebas anteriores logramos ver la aproximación tan precisa a las soluciones y con gran eficiencia en cuanto tiempo nos referimos. También es importante que el número de iteraciones en ambos métodos son casi iguales, pues notamos que en la mayoría de los datos observados, lo resuelve en la misma cantidad de pasos.

Realizamos otro experimento adicional para ver si la dimensión del problema influye en los resultados de esta comparación, las pruebas realizadas en una función tomada de [6], la función *Sistema Tridiagonal*, estas pruebas se realizaron con diferentes números de variables para verificar su eficiencia en tiempo de cómputo y en precisión.

11. **Función Sistema Tridiagonal** Sea  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  definida por

$$\begin{cases} 4(x_k - x_{k+1}^2) & , k = 1 \\ 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + 4(x_k - x_{k+1}^2) & , 1 < k < n \\ 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) & , k = n \end{cases}$$

con vector inicial  $\mathbf{x}_0 = (0, 0, 0, \dots, 0)^T$  y  $\mathbf{x}_* = (0.999, 1, 1, \dots, 1)^T$ .

Funcion	n	método	iteraciones	$\ f(x)\ $	tiempo
Sistema Tridiagonal	10	E.G	53	0.0000081441	0.0691900253
		J	53	0.0000081417	0.2567400932
		G.S	53	0.0000081442	0.6172008514
		G.C	- - -	- - -	- - -
Sistema Tridiagonal	100	E.G	53	0.0000083774	0.6140239239
		J	53	0.0000082994	1.7892389297
		G.S	53	0.0000082929	33.3194999695
		G.C	53	0.0000083699	1.8305399418
Sistema Tridiagonal	200	E.G	53	0.0000083774	1.2670710087
		J	53	0.0000081741	3.5320858955
		G.S	52	0.0000099899	226.7349050045
		G.C	53	0.0000083631	4.2891490459
Sistema Tridiagonal	500	E.G	53	0.0000083774	3.5662629604
		J	52	0.0000087689	16.8605470657
		G.S	51	0.0000093331	817.1732509136
		G.C	52	0.0000086423	14.0375490189

Como se puede observar en los datos de la anterior tabla, las iteraciones realizadas en cada caso no varían de manera significativa pero, los tiempos obtenidos cambian a medida que aumentamos el número de variables, en cuanto a la aproximación mediante el método de *Newton* inexacto es muy buena en comparación al método de *Newton*.

# Capítulo 5

## Conclusiones

El método de *Newton* inexacto resulta bastante bueno a la hora de resolver sistemas de ecuaciones no lineales, como se logró observar con los datos recogidos por las pruebas numéricas, nos queda por decir que el método *Newton* sigue siendo el mejor, hasta el momento, para encontrar la solución de los sistemas de ecuaciones no lineales, con esto no pretendemos quitar mérito al método de *Newton* inexacto, pues demostró ser competitivo cuando las condiciones del sistema son óptimas para aplicarlo.

Los resultados obtenidos en las pruebas numéricas, nos permitieron observar el comportamiento de los sistemas cuando se toman puntos iniciales muy lejanos y también diferentes tolerancias; estos datos son muy importantes cuando buscamos una solución del sistema que sea óptima, pues la convergencia del método es local, es decir en una vecindad al rededor de  $\mathbf{x}_*$  la solución del sistema.

Finalmente, este estudio numérico del método de *Newton* inexacto comparamos tres métodos iterativos, *Jacobi* (**J**), *Gauss-Seidel* (**G.S**), gradiente conjugado (**G.C**) y logramos comprobar que en los casos donde el método iterativo de (**G.C**) convergía, resultaba el más eficiente para llegar a la solución y en los que no converge es posible que el sistema este mal condicionado.

Para ampliar este estudio sería bueno hacer una comparación entre los métodos *Newton* global y *Newton* inexacto globalizado.

# Bibliografía

- [1] Richard L Burden, J Douglas Faires, and Simon Mochon, *Análisis numérico*, Grupo Editorial Iberoamericana, 1985.
- [2] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug, *Inexact newton methods*, SIAM Journal on Numerical analysis **19** (1982), no. 2, 400–408.
- [3] John E Dennis Jr and Robert B Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, vol. 16, Siam, 1996.
- [4] Gene H Golub and Charles F Van Loan, *Matrix computations. 1996*, Johns Hopkins University, Press, Baltimore, MD, USA (1996), 374–426.
- [5] Magnus Rudolph Hestenes and Eduard Stiefel, *Methods of conjugate gradients for solving linear systems*, vol. 49, NBS, 1952.
- [6] L Lukšan, *Inexact trust region method for large sparse systems of nonlinear equations*, Journal of Optimization Theory and Applications **81** (1994), no. 3, 569–590.
- [7] J. J. Moré, *A collection of nonlinear model problems*, Lectures in Applied Mathematics **26** (1990), 723–762.
- [8] H. B. Nielsen, *Uctp - test problems for unconstrained optimization*, Tech. report, IMM, nov 2000.
- [9] L Richard and J Burden, *Douglas faires, numerical analysis*, 1988.
- [10] Youcef Saad and Martin H Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on scientific and statistical computing **7** (1986), no. 3, 856–869.

- [11] Yousef Saad, *Iterative methods for sparse linear systems*, vol. 82, siam, 2003.