Universidad de La Laguna

Búsqueda de raíces mediante el método de Newton-Raphson

Alba Crespo Pérez, Raquel Espino Mantas y Robbert Jozef Michiels

12 de mayo de 2014

Facultad de Matemáticas
Universidad de La Laguna

Índice

Índice

Índice

Índice

Índice

1. Motivación y objetivos

Objetivos generales y específicos

El *objetivo principal* del presente trabajo reside en la implementación, mediante el uso del lenguaje de programación Python, de un algoritmo basado en el método de Newton - Raphson que nos permita llevar a cabo el **cálculo de las raíces de la función $\mathbf{f(x) = \cos(\pi x)}$**.

1. Motivación y objetivos

Objetivos generales y específicos

El *objetivo principal* del presente trabajo reside en la implementación, mediante el uso del lenguaje de programación Python, de un algoritmo basado en el método de Newton - Raphson que nos permita llevar a cabo el **cálculo de las raíces de la función $\mathbf{f(x) = \cos(\pi x)}$**.

Además, se pretende lograr la consecución de los siguientes *objetivos específicos*:

1. Efectuar un **análisis numérico y gráfico** de la evolución en el **número de iteraciones** requeridas por este método para la obtención de una raíz, **en función del error absoluto** de la estimación inicial tomada como punto de partida del método respecto a la solución real determinada tras su aplicación.

1. Motivación y objetivos

Objetivos generales y específicos

El *objetivo principal* del presente trabajo reside en la implementación, mediante el uso del lenguaje de programación Python, de un algoritmo basado en el método de Newton - Raphson que nos permita llevar a cabo el **cálculo de las raíces de la función $\mathbf{f(x) = \cos(\pi x)}$**.
Además, se pretende lograr la consecución de los siguientes *objetivos específicos*:

1. Efectuar un **análisis numérico y gráfico** de la evolución en el **número de iteraciones** requeridas por este método para la obtención de una raíz, **en función del error absoluto** de la estimación inicial tomada como punto de partida del método respecto a la solución real determinada tras su aplicación.
2. **Analizar el costo computacional**, en términos de tiempo de uso de CPU, **asociado a la ejecución** del algoritmo implementado, así como su sensibilidad ante modificaciones en los parámetros iniciales.

Origen del método y sus descubridores

Este método constituye una *vía algorítmica de análisis numérico* destinada a la **determinación de los ceros o raíces de una función** matemática.

Origen del método y sus descubridores

Este método constituye una *vía algorítmica de análisis numérico* destinada a la **determinación de los ceros o raíces de una función** matemática.

- Su primera descripción relevante fue realizada por **Isaac Newton** en su obra *De analysi per aequationes numero terminorum infinitas*, publicado en **1711**, así como en su tratado *De metodis fluxionum et serierum infinitarum*, editado en 1736 por John Colson bajo el título de *Método de las fluxiones*.

Origen del método y sus descubridores

Este método constituye una *vía algorítmica de análisis numérico* destinada a la **determinación de los ceros o raíces de una función** matemática.

- Su primera descripción relevante fue realizada por **Isaac Newton** en su obra *De analysi per aequationes numero terminorum infinitas*, publicado en **1711**, así como en su tratado *De metodis fluxionum et serierum infinitarum*, editado en 1736 por John Colson bajo el título de *Método de las fluxiones*.
- Sin embargo, si bien su reconocimiento no alcanzó en su momento la repercusión otorgada a los trabajos de Newton, dicho método encuentra mención previa en el libro *Aequationum Universalis* (**1690**), cuya publicación conllevó para su autor, el matemático inglés **Joseph Raphson**, la consecución del ingreso en la *Royal Society* de Londres.

2.2. Descripción teórica del método

Análisis del procedimiento

Para utilizar el método de Newton-Raphson es necesario comenzar con un **valor inicial** suficientemente **cercano a la raíz buscada**, que a medida que avance la ejecución del algoritmo progresará acercándose al valor real de esta raíz.

2.2. Descripción teórica del método

Análisis del procedimiento

Para utilizar el método de Newton-Raphson es necesario comenzar con un **valor inicial** suficientemente **cercano a la raíz buscada**, que a medida que avance la ejecución del algoritmo progresará acercándose al valor real de esta raíz.

Una vez fijada una estimación inicial $\mathbf{x_0}$, calcularemos la expresión de la **recta tangente** a la gráfica de la función **en el punto** $\mathbf{x = x_0}$. La **coordenada de corte** de dicha recta **con el eje horizontal** será considerada como un **valor más próximo** que x_0 a la raíz buscada, por lo que este paso podrá aplicarse de forma *recursiva* tomando como suposición al nuevo valor obtenido hasta lograr una aproximación aceptable de la raíz, dentro de un *margen de tolerancia* fijado de antemano.

2.2. Descripción teórica del método

Obtención del algoritmo recursivo de Newton-Raphson

Sabemos que la expresión de una recta de pendiente m y que pasa por el punto (a, b) viene por $y - b = m(x - a)$. Por tanto, la expresión de la recta tangente a una función $f(x)$ en el punto $x = x_0$ queda como:

$$\mathbf{y - f(x_0) = f'(x_0)(x - x_0)}$$

2.2. Descripción teórica del método

Obtención del algoritmo recursivo de Newton-Raphson

Sabemos que la expresión de una recta de pendiente m y que pasa por el punto (a, b) viene por $y - b = m(x - a)$. Por tanto, la expresión de la recta tangente a una función $f(x)$ en el punto $x = x_0$ queda como:

$$\mathbf{y - f(x_0) = f'(x_0)(x - x_0)}$$

Hallamos el punto de corte de la recta anterior con el eje de abscisas [a]:

$$y = 0 \quad \rightarrow \quad -f(x_0) = f'(x_0)(x_{n+1} - x_0) \quad \rightarrow \quad x_0 f'(x_0) - f(x_0) = x_{n+1} f'(x_0) \quad \rightarrow \quad \mathbf{x_{n+1} = x_0 - \frac{f(x_0)}{f'(x_0)}}$$

2.2. Descripción teórica del método

Obtención del algoritmo recursivo de Newton-Raphson

Sabemos que la expresión de una recta de pendiente m y que pasa por el punto (a, b) viene por $y - b = m(x - a)$. Por tanto, la expresión de la recta tangente a una función $f(x)$ en el punto $x = x_0$ queda como:

$$\mathbf{y - f(x_0) = f'(x_0)(x - x_0)}$$

Hallamos el punto de corte de la recta anterior con el eje de abscisas [a]:

$$y = 0 \quad \rightarrow \quad -f(x_0) = f'(x_0)(x_{n+1} - x_0) \quad \rightarrow \quad x_0 f'(x_0) - f(x_0) = x_{n+1} f'(x_0) \quad \rightarrow \quad \mathbf{x_{n+1} = x_0 - \frac{f(x_0)}{f'(x_0)}}$$

Vemos así que las funciones con amplias variaciones de pendiente en el entorno de sus raíces dificultarán la convergencia del algoritmo, ante lo cual el establecimiento de una *cota máxima de iteraciones* nos permitirá detectar la aparición de patrones de divergencia.

[a] La notación $\mathbf{x_{n+1}}$ indica que el valor resultante de x corresponderá a la estimación tomada como partida para la siguiente ejecución del algoritmo.

2.3. Análisis de la función de estudio

Cálculo teórico de raíces

Con el objetivo de lograr una mejor comprensión de los resultados obtenidos en la aproximación de las raíces de la función $f(x) = cos(\pi x)$ mediante el algoritmo de Newton-Raphson, procedemos a **calcular** dichas raíces **de modo teórico**, mediante la resolución de una sencilla **ecuación trigonométrica**:

2.3. Análisis de la función de estudio

Cálculo teórico de raíces

Con el objetivo de lograr una mejor comprensión de los resultados obtenidos en la aproximación de las raíces de la función $f(x) = cos(\pi x)$ mediante el algoritmo de Newton-Raphson, procedemos a **calcular** dichas raíces **de modo teórico**, mediante la resolución de una sencilla **ecuación trigonométrica**:

$$\mathbf{cos(\pi x) = 0} \quad \rightarrow \quad \pi x = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad \rightarrow \mathbf{x = \frac{1}{2} + k}, \ k \in \mathbb{Z}$$

Cálculo teórico de raíces

Con el objetivo de lograr una mejor comprensión de los resultados obtenidos en la aproximación de las raíces de la función $f(x) = cos(\pi x)$ mediante el algoritmo de Newton-Raphson, procedemos a **calcular** dichas raíces **de modo teórico**, mediante la resolución de una sencilla **ecuación trigonométrica**:

$$\mathbf{cos(\pi x) = 0} \quad \rightarrow \quad \pi x = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad \rightarrow \mathbf{x = \frac{1}{2} + k}, \;\; k \in \mathbb{Z}$$

Vemos así que se verifica lo siguiente:

$$cos(\pi x) = 0 \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

2.3. Análisis de la función de estudio

Cálculo teórico de raíces

Con el objetivo de lograr una mejor comprensión de los resultados obtenidos en la aproximación de las raíces de la función $f(x) = cos(\pi x)$ mediante el algoritmo de Newton-Raphson, procedemos a **calcular** dichas raíces **de modo teórico**, mediante la resolución de una sencilla **ecuación trigonométrica**:

$$\mathbf{cos(\pi x) = 0} \quad \rightarrow \quad \pi x = \frac{\pi}{2} + k\pi, k \in \mathbb{Z} \quad \rightarrow \mathbf{x = \frac{1}{2} + k}, \ k \in \mathbb{Z}$$

Vemos así que se verifica lo siguiente:

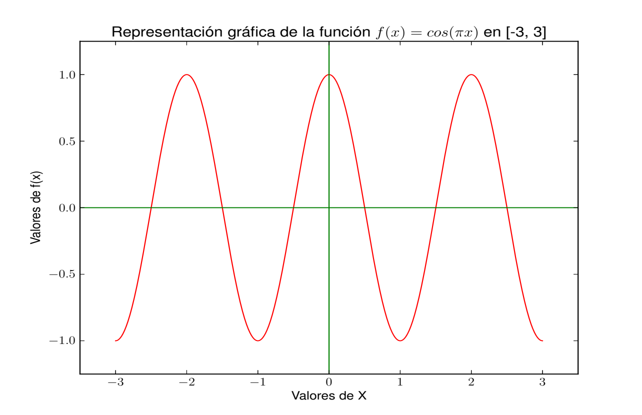
$$cos(\pi x) = 0 \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

Concluimos de este modo que la función estudiada es **periódica**, con periodo $\mathbf{P = 2}$.

2.3. Análisis de la función de estudio

Representación gráfica

El carácter periódico de la función $\mathbf{f(x) = cos(\pi x)}$, así como el valor de sus raíces reales, quedan reflejados de modo ilustrativo a través de la siguiente gráfica:



Puntos críticos y valores iniciales del algoritmo

Debido a la naturaleza del método de Newton-Raphson, es preciso señalar que su aplicación no podrá llevarse a cabo en caso de alcanzarse una **anulación de la derivada** para el punto de partida del algoritmo.

2.3. Análisis de la función de estudio

Puntos críticos y valores iniciales del algoritmo

Debido a la naturaleza del método de Newton-Raphson, es preciso señalar que su aplicación no podrá llevarse a cabo en caso de alcanzarse una **anulación de la derivada** para el punto de partida del algoritmo.

Analizaremos por tanto los **puntos críticos de la función** de estudio, para así conocer qué valores **deben ser evitados como suposiciones iniciales** a la hora de efectuar una invocación al método:

2.3. Análisis de la función de estudio

Puntos críticos y valores iniciales del algoritmo

Debido a la naturaleza del método de Newton-Raphson, es preciso señalar que su aplicación no podrá llevarse a cabo en caso de alcanzarse una **anulación de la derivada** para el punto de partida del algoritmo. Analizaremos por tanto los **puntos críticos de la función** de estudio, para así conocer qué valores **deben ser evitados como suposiciones iniciales** a la hora de efectuar una invocación al método:

$$\boldsymbol{(\cos(\pi x))' = 0} \;\rightarrow\; -\pi sen(\pi x) = 0 \;\rightarrow\; sen(\pi x) = 0 \;\rightarrow$$
$$\rightarrow\; \pi x = k\pi,\; k \in \mathbb{Z} \;\rightarrow\; \mathbf{x = k},\; k \in \mathbb{Z} \;\rightarrow x = ..., -2, -1, 1, 2, ...$$

2.3. Análisis de la función de estudio

Puntos críticos y valores iniciales del algoritmo

Debido a la naturaleza del método de Newton-Raphson, es preciso señalar que su aplicación no podrá llevarse a cabo en caso de alcanzarse una **anulación de la derivada** para el punto de partida del algoritmo. Analizaremos por tanto los **puntos críticos de la función** de estudio, para así conocer qué valores **deben ser evitados como suposiciones iniciales** a la hora de efectuar una invocación al método:

$$\mathbf{(\cos(\pi x))' = 0} \quad \rightarrow \quad -\pi sen(\pi x) = 0 \quad \rightarrow \quad sen(\pi x) = 0 \quad \rightarrow$$
$$\rightarrow \quad \pi x = k\pi,\ \ k \in \mathbb{Z} \quad \rightarrow \quad \mathbf{x = k},\ \ k \in \mathbb{Z} \quad \rightarrow x = ..., -2, -1, 1, 2, ...$$

Observamos así que los puntos críticos de la función $f(x) = cos(\pi x)$ se encuentran separados una distancia de *dos unidades*, lo que respalda las conclusiones obtenidas acerca de su periodicidad.

3.1. Descripción de los experimentos

Síntesis de los experimentos realizados

1. Para llevar a cabo el cálculo de las raíces de la función objeto de estudio, se ha efectuado la **implementación de un algoritmo recursivo** basado en el método **de Newton-Raphson**, en cuya ejecución hemos utilizado *distintos valores como parámetros iniciales* del método para así observar los efectos producidos por la variabilidad de éstos sobre la solución final.

3.1. Descripción de los experimentos

Síntesis de los experimentos realizados

1. Para llevar a cabo el cálculo de las raíces de la función objeto de estudio, se ha efectuado la **implementación de un algoritmo recursivo** basado en el método **de Newton-Raphson**, en cuya ejecución hemos utilizado *distintos valores como parámetros iniciales* del método para así observar los efectos producidos por la variabilidad de éstos sobre la solución final.
2. Ha sido diseñado un programa Python destinado a permitir el **recuento de las iteraciones** requeridas para el cálculo de una raíz en función del error absoluto de la estimación inicial, *almacenando los resultados en un fichero* para su posterior lectura y **representación gráfica** por parte de un programa creado a tal efecto.

3.1. Descripción de los experimentos

Síntesis de los experimentos realizados

1. Para llevar a cabo el cálculo de las raíces de la función objeto de estudio, se ha efectuado la **implementación de un algoritmo recursivo** basado en el método **de Newton-Raphson**, en cuya ejecución hemos utilizado *distintos valores como parámetros iniciales* del método para así observar los efectos producidos por la variabilidad de éstos sobre la solución final.
2. Ha sido diseñado un programa Python destinado a permitir el **recuento de las iteraciones** requeridas para el cálculo de una raíz en función del error absoluto de la estimación inicial, *almacenando los resultados en un fichero* para su posterior lectura y **representación gráfica** por parte de un programa creado a tal efecto.
3. Con el fin de analizar el costo computacional del algoritmo, hemos realizado un diagrama que refleja el **tiempo total de CPU** empleado en distintas ejecuciones, **frente al volumen de iteraciones** requerido durante las mismas.

3.2. Descripción del material

Características de hardware y software

Seguidamente se detallan las especificaciones técnicas de la computadora empleada para llevar a cabo los experimentos, así como las versiones de los programas utlizados:

3.2. Descripción del material

Características de hardware y software

Seguidamente se detallan las especificaciones técnicas de la computadora empleada para llevar a cabo los experimentos, así como las versiones de los programas utlizados:

1. **HARDWARE:**
   - **Tipo de CPU:** Pentium(R) Dual-Core CPU, GenuineIntel, T4500.
   - **Velocidad de la CPU:** 1200 Hz.
   - **Tamaño del caché:** 1024 KB.
   - **Memoria RAM:** 8 GB.

3.2. Descripción del material

Características de hardware y software

Seguidamente se detallan las especificaciones técnicas de la computadora empleada para llevar a cabo los experimentos, así como las versiones de los programas utlizados:

1. **HARDWARE:**
   - **Tipo de CPU:** Pentium(R) Dual-Core CPU, GenuineIntel, T4500.
   - **Velocidad de la CPU:** 1200 Hz.
   - **Tamaño del caché:** 1024 KB.
   - **Memoria RAM:** 8 GB.
2. **SOFTWARE:**
   - **Versión de Python:** 2.7.3.
   - **Compilador Python:** GCC 4.7.2.
   - **Sistema operativo:** Linux-3.5.0-17-genérico con Ubuntu-12.10-quantal.
   - **Fecha de creación de la versión de Python:** Sep 26 2012 21:53:58.

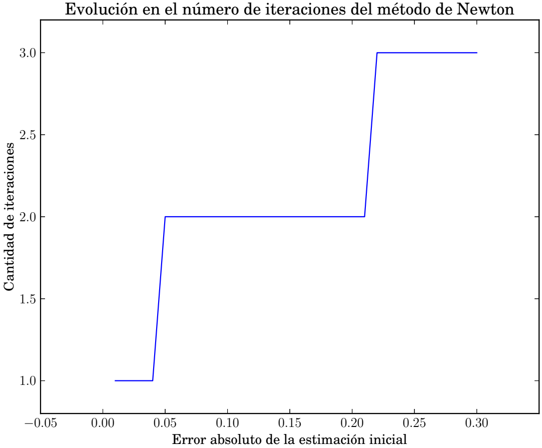
3.3. Resultados obtenidos

Tabla ejemplificativa de raíces y parámetros de inicio

INICIO	TOLERANCIA	ITERACIONES	RESULTADO
-7.08	10^{-7}	3	-8.5
-7.1	10^{-7}	4	-9.5
-7.2	10^{-7}	3	-7.5
-5.8	10^{-5}	2	-5.49999728877
-5.8	10^{-6}	3	-5.5
-0.23	10^{-7}	3	-0.5
-0.23	10^{-9}	3	-0.5
1.24	10^{-6}	2	1.50000001507
1.24	10^{-8}	3	1.5
2.17	10^{-12}	4	2.5
3.45	10^{-4}	1	3.49999999976
3.45	10^{-10}	2	3.5

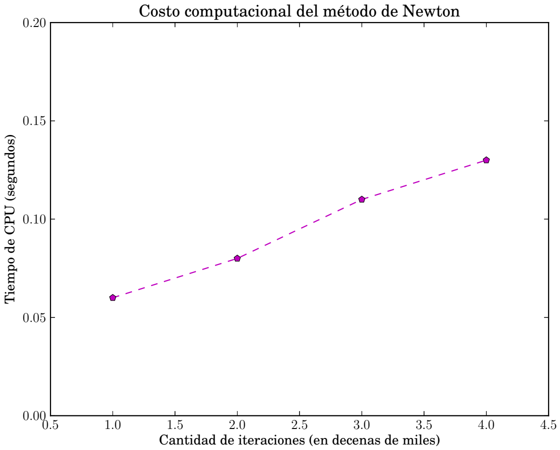
3.3. Resultados obtenidos

Incremento en la cantidad de iteraciones frente a la variación de los errores absolutos iniciales



3.3. Resultados obtenidos

Tiempo total de uso de CPU por volumen de iteraciones



3.4. Análisis de los resultados

Ejecución del algoritmo de Newton-Raphson

- En caso de proporcionar como suposición de partida un valor cercano a un **extremo relativo** de la función, **la iteración inicial** diverge respecto a la raíz más próxima a dicho valor, puesto que la acentuada pendiente de la tangente origina que su corte con la abscisa se encuentre notablemente alejado del valor de partida.

3.4. Análisis de los resultados

Ejecución del algoritmo de Newton-Raphson

- En caso de proporcionar como suposición de partida un valor cercano a un **extremo relativo** de la función, **la iteración inicial** diverge respecto a la raíz más próxima a dicho valor, puesto que la acentuada pendiente de la tangente origina que su corte con la abscisa se encuentre notablemente alejado del valor de partida.
- **Todas las ejecuciones** que no comienzan con una anulación de la derivada **resultan convergentes**. Esto es así puesto que la periodicidad de la función permite que los puntos originados en iteraciones divergentes propicien el inicio de una recursión convergente.

3.4. Análisis de los resultados

Ejecución del algoritmo de Newton-Raphson

- En caso de proporcionar como suposición de partida un valor cercano a un **extremo relativo** de la función, **la iteración inicial** diverge respecto a la raíz más próxima a dicho valor, puesto que la acentuada pendiente de la tangente origina que su corte con la abscisa se encuentre notablemente alejado del valor de partida.
- **Todas las ejecuciones** que no comienzan con una anulación de la derivada **resultan convergentes**. Esto es así puesto que la periodicidad de la función permite que los puntos originados en iteraciones divergentes propicien el inicio de una recursión convergente.
- Observamos que el método de Newton-Raphson presenta una **elevada velocidad de convergencia** para la función de estudio, puesto que la cantidad de iteraciones requerida hasta lograr la obtención de una raíz válida no llega a superar las cuatro iteraciones, aún en caso de seleccionar estrechos márgenes de tolerancia.

3.4. Análisis de los resultados

Evolución de las iteraciones y costo computacional

- Queda patente que el **incremento en el volumen de iteraciones** recursivas en función del error absoluto de la estimación inicial muestra un **carácter suave**, en conexión con la alta eficiencia y velocidad de convergencia del algoritmo.

3.4. Análisis de los resultados

Evolución de las iteraciones y costo computacional

- Queda patente que el **incremento en el volumen de iteraciones** recursivas en función del error absoluto de la estimación inicial muestra un **carácter suave**, en conexión con la alta eficiencia y velocidad de convergencia del algoritmo.
- Sin embargo, la **separación constante** de una unidad existente **entre raíces imposibilita el análisis** del comportamiento del método **para errores absolutos superiores a 0.3** en la suposición de partida.

3.4. Análisis de los resultados

Evolución de las iteraciones y costo computacional

- Queda patente que el **incremento en el volumen de iteraciones** recursivas en función del error absoluto de la estimación inicial muestra un **carácter suave**, en conexión con la alta eficiencia y velocidad de convergencia del algoritmo.
- Sin embargo, la **separación constante** de una unidad existente **entre raíces imposibilita el análisis** del comportamiento del método **para errores absolutos superiores a 0.3** en la suposición de partida.
- La información relativa al costo computacional del algoritmo nos permite corroborar una vez más la **notable eficiencia** del método, puesto que es preciso reiterar decenas de miles de veces su invocación hasta obtener valores significativos para los tiempos de uso de CPU.

Evolución de las iteraciones y costo computacional

- Queda patente que el **incremento en el volumen de iteraciones** recursivas en función del error absoluto de la estimación inicial muestra un **carácter suave**, en conexión con la alta eficiencia y velocidad de convergencia del algoritmo.
- Sin embargo, la **separación constante** de una unidad existente **entre raíces imposibilita el análisis** del comportamiento del método **para errores absolutos superiores a 0.3** en la suposición de partida.
- La información relativa al costo computacional del algoritmo nos permite corroborar una vez más la **notable eficiencia** del método, puesto que es preciso reiterar decenas de miles de veces su invocación hasta obtener valores significativos para los tiempos de uso de CPU.
- Por último, observamos que **la variabilidad en los parámetros** iniciales **tan sólo afecta** a la utilización de recursos computacionales **cuando** su modificación **altera el volumen de iteraciones requeridas** en la ejecución.

4. Conclusiones

Síntesis de conclusiones finales

- Las **raíces** de la función $f(x) = cos(\pi x)$ vienen dadas de la forma:

  $$\mathbf{x = \frac{1}{2} + k},\ k \in \mathbb{Z} \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

4. Conclusiones

Síntesis de conclusiones finales

- Las **raíces** de la función $f(x) = cos(\pi x)$ vienen dadas de la forma:

  $$\mathbf{x = \frac{1}{2} + k},\ k \in \mathbb{Z} \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

- Para el caso de estudio, **la convergencia** del algoritmo de Newton-Raphson se encuentra prácticamente **garantizada**, debido a la reducida separación entre raíces y a su carácter de **periodicidad**.

4. Conclusiones

Síntesis de conclusiones finales

- Las **raíces** de la función $f(x) = cos(\pi x)$ vienen dadas de la forma:

  $$\mathbf{x = \frac{1}{2} + k},\ k \in \mathbb{Z} \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

- Para el caso de estudio, **la convergencia** del algoritmo de Newton-Raphson se encuentra prácticamente **garantizada**, debido a la reducida separación entre raíces y a su carácter de **periodicidad**.
- La **eficiencia del algoritmo** implementado resulta considerablemente **elevada**, al requerir ínfimos tiempos de CPU para la determinación de una raíz y no superar en ningún caso el margen de cuatro iteraciones.

4. Conclusiones

Síntesis de conclusiones finales

- Las **raíces** de la función $f(x) = cos(\pi x)$ vienen dadas de la forma:
  $\mathbf{x = \frac{1}{2} + k}$, $k \in \mathbb{Z} \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$
- Para el caso de estudio, **la convergencia** del algoritmo de Newton-Raphson se encuentra prácticamente **garantizada**, debido a la reducida separación entre raíces y a su carácter de **periodicidad**.
- La **eficiencia del algoritmo** implementado resulta considerablemente **elevada**, al requerir ínfimos tiempos de CPU para la determinación de una raíz y no superar en ningún caso el margen de cuatro iteraciones.
- El método de Newton-Raphson demuestra caracterizarse por una **notable velocidad de convergencia** para la función considerada, llegando a proporcionar hasta cinco cifras decimales correctas a lo largo de una única iteración.

4. Conclusiones

Síntesis de conclusiones finales

- Las **raíces** de la función $f(x) = cos(\pi x)$ vienen dadas de la forma:

  $$\mathbf{x = \frac{1}{2} + k},\ k \in \mathbb{Z} \iff x = ..., -2.5, -1, 5, -0.5, 0.5, 1.5, 2.5, ...$$

- Para el caso de estudio, **la convergencia** del algoritmo de Newton-Raphson se encuentra prácticamente **garantizada**, debido a la reducida separación entre raíces y a su carácter de **periodicidad**.
- La **eficiencia del algoritmo** implementado resulta considerablemente **elevada**, al requerir ínfimos tiempos de CPU para la determinación de una raíz y no superar en ningún caso el margen de cuatro iteraciones.
- El método de Newton-Raphson demuestra caracterizarse por una **notable velocidad de convergencia** para la función considerada, llegando a proporcionar hasta cinco cifras decimales correctas a lo largo de una única iteración.
- El reducido valor de los tiempos de CPU implica que las diferencias globales ante alteraciones en los parámetros iniciales pueden ser consideradas despreciables.

5. Bibliografía

- ALEX MARTELLI, *Python: guía de referencia*, Madrid, España, Anaya, 2003.
- A.A. SAMARSKI, *Introducción a los métodos numéricos*, Moscú, Rusia, MIR Editorial, 1986.
- STEVEN C. CHAPRA, *Métodos numéricos para ingenieros*, México, McGraw-Hill Interamericana de México, 2007.
- Desarrollo histórico del método de Newton-Raphson y calculadora virtual de raíces: *http://www.uam.es/personal_pdi/ciencias/barcelo/cnumerico/recursos/newton.html*