El problema de las raíces múltiples

La bisección no es aplicable a raíces dobles (f′(s)=0). ¿Y Newton?

Cerca de una raíz doble s una función f(x) es similar a una parábola:

$$f(x) = f\underbrace{(s)}_{0} + f'\underbrace{(s)}_{0}(x-s) + \frac{f''(s)}{2}(x-s)^2 + \ldots$$

Luego $f(x) \approx \frac{f''(s)}{2}(x-s)^2$ y su derivada es $f'(x) \approx f''(s)(x-s)$

f(x) y f′(x) → 0 para x=s, pero f(x) va con $(x-s)^2$ y la derivada con (x-s)

por lo que el cociente $\frac{f(x)}{f'(x)}$ esta bien definido: $\frac{f(x)}{f'(x)} \approx \frac{1}{2}(x-s) \underset{x\to s}{\to} 0$

El método de Newton no "explota" en raíces dobles.

Prob LAB: aplicar Newton a f(x)=1+cos(x)

Iteración: $\left.\begin{matrix} f(x) = 1+\cos(x) \\ f'(x) = -\sin(x) \end{matrix}\right\} \to x = x - \frac{1+\cos(x)}{-\sin(x)} = x + \frac{1+\cos(x)}{\sin(x)}$

s=π es una solución ya que f(π) = 1+cos(π) =1-1=0

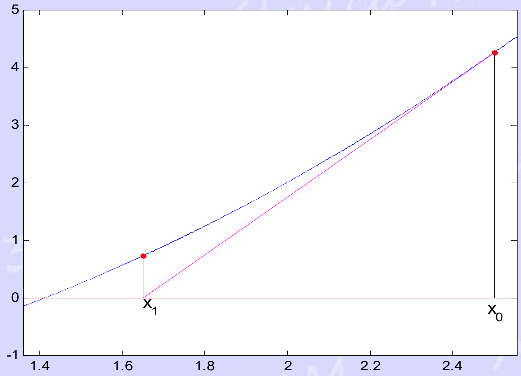
```
Solución correcta s = π = 3.1415926535897931
Arrancando en x0=3.000000
----------------------------------------------
It  1 x=3.0709148443026528  3.1
It  2 x=3.1062684671563341  3.1
It  3 x=3.1239323971628581  3.1
It  4 x=3.1327627548819419  3.1
It  5 x=3.1371777329211668  3.14
It  6 x=3.1393851968410607  3.14
It  7 x=3.1404889256636337  3.14
It  8 x=3.1410407896827648  3.14
```

Newton funciona pero muy lentamente (parece un método lineal)

¿Por qué perdemos la convergencia cuadrática?

s=π es una raíz doble ya que f(π) = 0 y f′(π) = -sin(π) también es 0

¿Por qué “falla” Newton en este caso?

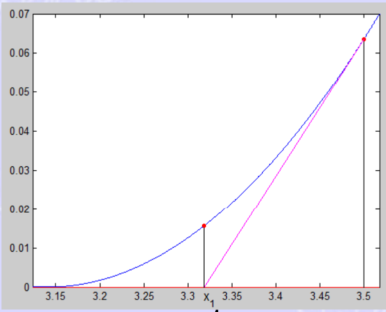


Newton asumía que cerca de la raíz una función es como una recta:

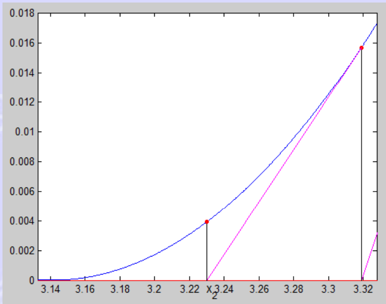
$$f(x) = f'(s)(x-s) + \ldots$$

En raíces dobles f’(s)=0 y f(x) se parece al siguiente término de la serie de Taylor

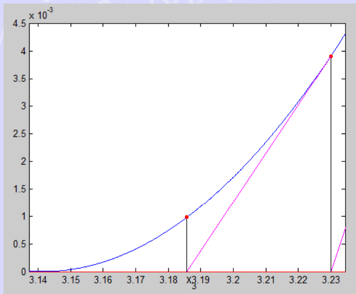
→ ya no es una recta sino una parábola.



x1



x2



x3

Tras 3 iteraciones la recta tangente no consigue “pegarse” a la función

Orden de Newton en raíces dobles

¿De qué orden es Newton en raíces dobles?

Recordando que cerca de una raíz doble se verifica que: $\dfrac{f(x)}{f'(x)} \approx \dfrac{1}{2}(x-s)$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \approx x_n - \frac{1}{2}(x_n - s) \longrightarrow x_{n+1} - s \approx \frac{1}{2}(x_n - s)$$

Por lo tanto $e_{n+1} \sim e_n/2$ → ¡ Newton se ha convertido en lineal !

En raíces dobles Newton se comporta como un método lineal con K=1/2 (en las gráficas anteriores se aprecia como el error se reduce a la mitad)

Se puede demostrar que para una raíz p-ésima el valor de K = 1 - 1/p.

Con raíces de multiplicidad alta K→1 y la convergencia es muy lenta.

Posibles soluciones

Si queremos seguir teniendo convergencia cuadrática:

a) Cambiar a un método de orden superior, que substituya la aproximación de una recta con una aproximación a una parábola: por ejemplo usar 3 puntos e interpolar una parábola (método de Mueller).

La solución de la raíz de la parábola sería nuestro siguiente punto.

b) Modificar Newton para que mantenga convergencia cuadrática:

$$x_{n+1} = x_n - p\frac{f(x_n)}{f'(x_n)}$$

Esta variante mantiene convergencia cuadrática para una raíz con multiplicidad p.

Este "apaño" no es muy útil en la práctica.
Si no se donde está la raíz, menos voy a saber si es múltiple o no.

Más problemas en raíces dobles

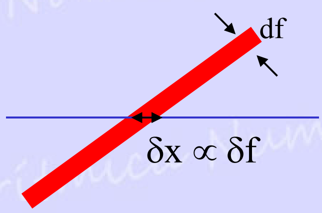
```
It 21 x=3.141592586338427‌8  f(x)= 9.1e-015.  3.1415926
It 22 x=3.1415926193555426  f(x)= 2.2e-015.  3.1415926
It 23 x=3.1415926355706341  f(x)= 5.6e-016.  3.1415926
It 24 x=3.1415926417319815  f(x)= 1.1e-016.  3.1415926
It 25 x=3.1415926510947800  f(x)= 1.1e-016.  3.14159265
It 26 x=3.1415926510947800  f(x)= 0.0e+000.  3.14159265
```

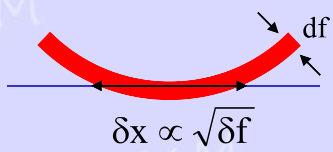
La iteración termina con 8/9 cifras correctas, ¿qué ha sido del resto?

¿Por qué termina la iteración?

Como f(x) se hace 0 se supone que hemos llegado al objetivo

¿Cómo es posible qué f(x)=0 si estamos todavía lejos de s?





Variantes de los métodos vistos

- Representante de métodos robustos: BISECCIÓN.

  No “pierde” la solución: siempre converge pero de forma lenta.

- Representante de métodos rápidos: NEWTON.

  Muy rápido cuando converge.
  Exige poder conocer la derivada de la función.
  Más costoso: requiere 2 evaluaciones por iteración

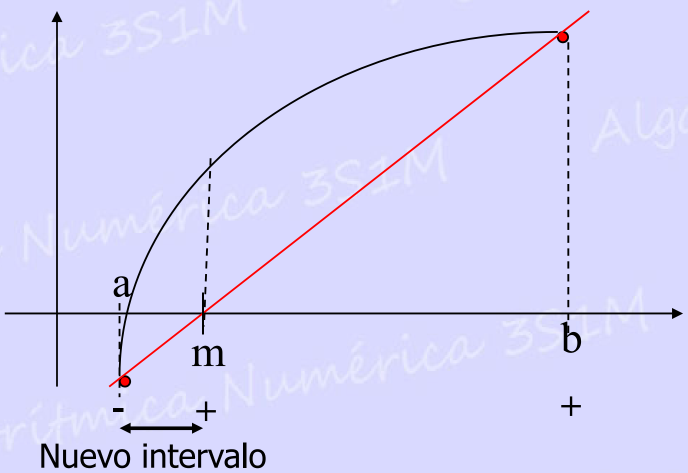
De cada uno de estos métodos se han propuestos diversas variantes que tratan de solucionar o reducir sus problemas, manteniendo en la medida de lo posible sus ventajas.

Método de regula-falsi

Variante de bisección: usa más información de f(a), f(b), no sólo el signo.

Inicio: intervalo [a,b] verificando $f(a)\cdot f(b)<0$ (cambio de signo)

Regla: Solución más probable en la recta que une (a, f(a)) con (b, f(b)).
Quedarse con sub-intervalo que mantiene el cambio de signo.
Repetir el proceso.



$$m = \frac{a\cdot f(b) - b\cdot f(a)}{f(b) - f(a)}$$

La única diferencia en el código es usar este valor de m como el siguiente punto en vez de usar el punto medio m=(a+b)/2

Aproximaciones a Newton

Métodos tipo secante = Newton + Aproximación numérica de la derivada:

$$x_{n+1} = x_n - \frac{f(x_n)}{"f'(x_n)"}$$

Varias posibilidades para aproximar "f'(x)":

$$"f'(x_n)" = \frac{f(x_n + h) - f(x_n)}{h} \longrightarrow$$ Exige una 2ª evaluación de f(x) en cada paso

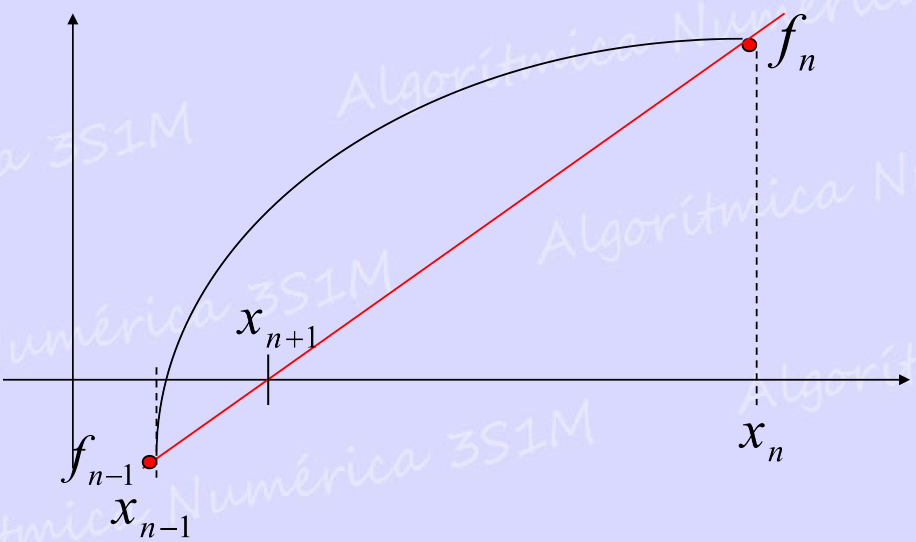
$$"f'(x_n)" = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \longrightarrow$$ **Método de la secante**: usa las dos últimas evaluaciones de f(x) para aproximar f'(x).

¿Orden de secante?

$e_{n+1} \sim K \cdot e_n^{1.62}$ (mejor que lineal, peor que Newton)

Método de la secante (gráficamente)

$$x_{n+1} = x_n - \frac{f_n}{\dfrac{f_n - f_{n-1}}{x_n - x_{n-1}}} = \frac{x_{n-1} \cdot f_n - x_n \cdot f_{n-1}}{f_n - f_{n-1}} \qquad con \quad f_n = f(x_n)$$



Método de Halley

Consiste en "incrementar" el orden de Newton. En vez de la recta tangente en x_0 usa la parábola tangente → exige conocer $f(x_0)$, $f'(x_0)$ y $f''(x_0)$:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}\left[1 - \frac{f(x_n)}{f'(x_n)} \cdot \frac{f''(x_n)}{2f'(x_n)}\right]^{-1}$$

- Con la letra pequeña habitual el método es de orden 3: $e_{n+1} \sim K \cdot e_n^3$
- Usa 3 evaluaciones por iteración y necesita conocer f' y f''
- Salvo aplicaciones específicas no merece la pena incrementar orden
  - 3 iteraciones Newton (6 eval): $e \rightarrow ((e^2)^2)^2) = e^8$ $(10^{-1} \rightarrow 10^{-8})$
  - 2 iteraciones Halley (6 eval): $e \rightarrow ((e^3)^3) = e^9$ $(10^{-1} \rightarrow 10^{-9})$
- El principal problema es saber dónde empezar, tener convergencia, no el hacer una iteración de más o de menos.

Aceleración de métodos lineales

En un método lineal el error se comporta de una forma regular $e_n \sim K \cdot e_{n-1}$

Podemos usar ese decaimiento regular para mejorar nuestra estimación.

Si tenemos 3 iteraciones de un método lineal: x_{n-1}, x_n, x_{n+1} se verificará que:

$$(x_{n+1} - s) \sim K \cdot (x_n - s)$$
$$(x_n - s) \sim K \cdot (x_{n-1} - s)$$

Dividiendo eliminamos K y podemos despejar $s \approx x_{n+1} - \frac{(x_{n+1} - x_n)^2}{(x_{n+1} - 2x_n + x_{n-1})}$

Usando esta nueva aproximación como punto de partida podemos aplicar otras 2 iteraciones del método original (→) y tenemos de nuevo tres valores de la sucesión lineal a los que volver a aplicar la fórmula de "aceleración" (→).

$$x_1 \rightarrow x_2 \rightarrow x_3$$
$$\downarrow$$
$$x_3^+ \rightarrow x_4 \rightarrow x_5$$
$$\downarrow$$
$$x_5^+ \cdots$$

RESUMEN: tipos de problemas de Newton

Newton "sobre la marcha": Ponerse a aplicar Newton : x_0, x_1, x_2, …

A partir de los resultados, estimar errores, decidir si parar, …

Nos basamos en: $e_n = |x_n - s| \approx |x_n - x_{n+1}| \quad y \quad e_{n+1} \approx K \cdot e_n^2$

Newton "a-prori" (sin iterar): probar que hay convergencia a partir de un x_0 dado o perteneciente a un intervalo.

Se trata de acotar M y el error del punto inicial (e_0):

e_0 = máximo valor posible de $|x_0 - s|$ $\quad M = \dfrac{\max\{|f''(x)|\}}{2\min\{|f'(x)|\}}$

Determinados M y e_0 hay convergencia si $M \cdot e_0 < 1$

¿Nº de iteraciones para una cierta tolerancia? $e_n \leq \dfrac{1}{M}(M \cdot e_0)^{2^n} < tol$

Problemas restantes

Problema 2: Bisección, similar al problema. 1

Problema 5: ejemplo típico de problema de Newton "a-priori":
Estimar M, error inicial e0, usar cota para deducir nº de iteraciones.
Similar al problema 6.

Problema 7: Mezcla de ambos tipos.
a,b) existencia de s y demostrar convergencia para todo punto de [0,1]
c) Dar la expresión de Newton y aplicar a x_0=0.5 para obtener x_1, x_2, x_3.
d) Usar valores obtenidos al iterar para estimar errores.

Problema 8:

1ª parte: Newton "a-priori" → estimar M, obtener cota del error

2ª parte: aplicación al cálculo de la raíz cuadrada.

Problema 2

Problema 2: Se quiere resolver aproximadamente la ecuación $x = \log\left(\frac{1}{x}\right)$

Razonar si se puede usar la bisección para hallar una solución s en [1/2,1]

Si se puede, obtener una aproximación con un error menor o igual a 1/16.

$$x = \log\left(\frac{1}{x}\right) = -\log(x) \Rightarrow f(x) = x + \log(x) = 0$$

$f(0.5) = -0.1931 < 0$

$f(1) = 1 > 0$

$\longrightarrow$ Cambio de signo: bisección puede actuar

Para conseguir un error menor o igual que 1/16 basta bajar hasta un intervalo de tamaño 1/8 y dar como estimación su punto medio.

[0.50, 1.00] = [- , +]. s = 0.75 f(0.750)=0.46 (+). Raíz en [0.5, 0.75]

[0.50, 0.75] = [- , +]. s = 0.625, f(0.625)=0.15 (+). Raíz en [0.5 0.625]

[0.50, 0.625] --> este intervalo tiene ancho 1/8. Si tomo el punto medio

$s \cong 0.5625$ tengo asegurado error < 1/16.

Problema 5 (julio 2013)

Sea la ecuación $x = \dfrac{\cos(x)}{2}$, cuya solución está en intervalo [0,1]:

a) Si trabajamos con una representación en coma flotante con una mantisa de 15 bits, ¿cuántas iteraciones como máximo tendría sentido aplicar del método de la bisección?

b) Si utilizamos el método de Newton, ¿cuál sería el mayor error inicial que nos podríamos permitir en la hipótesis inicial para asegurar la convergencia?

c) Si empezamos en x_0=0.5, dar una cota del error inicial e_0.

¿Cuántas iteraciones serían necesarias para alcanzar la precisión de la representación?

Problema 5 (julio 2013)

a) 15 iteraciones (cada iteración la bisección gana un factor 2 = un bit)

b) Recordar: hay convergencia si $e_0 < 1/M$

$$\begin{aligned} & f(x) = x - \cos(x)/2 \\ & f'(x) = 1 + \sin(x)/2 \longrightarrow |f'(x)| \geq 1/2 \\ & f''(x) = \cos(x)/2 \longrightarrow |f''(x)| \leq 1/2 \end{aligned} \Bigg\} \quad M = \frac{\max\{|f''(x)|\}}{2\min\{|f'(x)|\}} = \frac{1/2}{1} = \frac{1}{2}$$

Podemos asegurar convergencia si e_0 es menor que $1/M = 2$

c) Si $x_0=0.5$, ¿error inicial? ¿cuántas iteraciones tiene sentido hacer?

15 bits de mantisa → limite de la precisión es del orden de 2^{-15}

$$e_n \leq \frac{1}{M}(M \cdot e_0)^{2^n} \qquad M = \frac{1}{2}, e_0 \leq \frac{1}{2} \quad \rightarrow e_n \leq 2\left(\frac{1}{4}\right)^{2^n} \leq 2^{-15} \quad ??$$

$$\log_2\left(\frac{1}{4}\right)^{2^n} \leq \log_2\left(2^{-16}\right) \Rightarrow 2^n(-2) \leq -16 \Rightarrow 2^n \geq 8$$

$\boxed{\text{3 iteraciones}}$

Problema 7 (enero 2014)

Sea la ecuación $x = e^{-x}$. Se pide:

a) Demostrar que la ecuación anterior tiene al menos una solución en [0,1].

b) Demostrar que **para cualquier punto inicial** en el intervalo [0,1] el método de Newton puede usarse para hallar la solución.

c) Dar la expresión del método de Newton y obtener las 3 primeras iteraciones si partimos de x0=0.5. Listar los tres valores en una tabla con 8 decimales.

d) Usando **únicamente los valores obtenidos antes** para x1, x2 y x3, estimar los errores para las 3 iteraciones calculadas. Listar dichos errores usando notación científica.

Usar el hecho de que si el método de Newton converge hacia la solución s, se verifica que:

$$\begin{cases} |x_n - s| = e_n \approx |x_{n+1} - x_n| \\ \\ e_{n+1} \approx K e_n^2 \end{cases}$$

Problema 7 (enero 2014)

a) La ecuación dada es equivalente a $f(x) = x - e^{-x} = 0$

Evaluando en 0 y 1: f(0)=-1, f(1)=1-1/e >0

Hay cambio de signo, función continua, al menos una raíz.

b) Newton funciona si $|Me_0| < 1$ con $M = \frac{\max|f''(x)|}{2\min|f'(x)|}$ en el intervalo en cuestión

$$\begin{array}{l} f'(x) = 1 + e^{-x} \\ f''(x) = -e^{-x} \end{array} \longrightarrow \begin{array}{l} |f'(x)| = 1 + e^{-x} > 1 + 1/e = \min|f'(x)| \\ |f''(x)| = e^{-x} < 1 = \max|f''(x)| \end{array} \longrightarrow M = \frac{1}{2(1.368)} = 0.3655$$

Por otra parte, el error inicial e0 para cualquier punto del intervalo no puede superar su ancho: $|e_0| < 1$

Y por lo tanto $|Me_0| = M|e_0| < M = 0.3655 < 1$ y tenemos convergencia.

Problema 7 (enero 2014)

c) Dar la expresión del método de Newton y obtener las 3 primeras iteraciones si partimos de x0=0.5. Listar los tres valores en una tabla con 8 decimales:

$$x = x - \frac{f(x)}{f'(x)} = x - \frac{x - e^{-x}}{1 + e^{-x}} = \frac{1 + x}{1 + e^{x}}$$

Iter	x(n)
1	**0.56631100**
2	**0.56714317**
3	**0.56714329**

d) Para hallar los errores de x1 y x2 podemos usar el hecho de que:

$$|e_1| = |x_1 - s| \approx |x_1 - x_2| = 8.3 \cdot 10^{-4}$$
$$|e_2| = |x_2 - s| \approx |x_2 - x_3| = 1.2 \cdot 10^{-7}$$

Esto no vale para estimar e3 porque no disponemos de x4. La alternativa es usar e1 y e2 para hallar la constante K y luego estimar e3 como $e_3 \sim K \cdot e_2^2$

$$e_2 \approx K \cdot e_1^2 \longrightarrow K \approx \frac{e_2}{e_1^2} \approx 0.17 \longrightarrow e_3 \approx Ke_2^2 = 0.17 \cdot \left(1.2 \cdot 10^{-7}\right)^2 \approx 2.4 \cdot 10^{-15}$$

Problema 8

Para calcular $\sqrt{c}$ vamos a aplicar el método de Newton a f(x) = x^2-c :

a) Verificar que la iteración de Newton corresponde a $x_{n+1} = \dfrac{x_n^2 + c}{2x_n}$

b) Demostrar que para un punto inicial $x_0>0$, todos los puntos de la sucesión se encuentran a la derecha de la raíz: $x_{n+1} - \sqrt{c} > 0$

c) Si c≥1, hallar una cota de M = f''(x)/(2·f'(x)) y demostrar que el método converge si $|e_0|<2$ y que entonces se verifica que:

$$e_n \leq 2 \cdot \left(\frac{e_0}{2} \right)^{2^n}$$

Problema 8

a) Verificar la expresión de la iteración.

$$\begin{array}{l} f(x) = x^2 - c \\ f'(x) = 2x \end{array} \longrightarrow x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - c}{2x_n} = \frac{x_n^2 + c}{2x_n}$$

b) Demostrar que para un punto inicial x0>0, el resto de los puntos de la sucesión se encuentran a la derecha de la raíz, esto es:

Vamos a demostrar que $\left(x_{n+1} - \sqrt{c}\right)$ es positivo para n=0, 1, …

Método Newton Saco factor común Cuadrado diferencia

$$\left(x_{n+1} - \sqrt{c}\right) = \left(\frac{x_n^2 + c}{2x_n} - \sqrt{c} \right) = \frac{1}{2x_n}\left(x_n^2 + c - 2x_n\sqrt{c}\right) = \frac{1}{\underset{+}{2x_n}} \underset{+}{\left(x_n - \sqrt{c}\right)^2} > 0$$

Problema 8

c) Si c>=1, hallar cota M de f″(x)/(2·f′(x)) y demostrar que el método converge si $|e_0|<2$ y se verifica que: $$e_n \le 2\cdot\left(\frac{e_0}{2}\right)^{2^n}$$

Como $x_n > \sqrt{c} \ge 1$ solo tengo que preocuparme de la zona x>1

$$\begin{array}{ll} f'(x) = 2x \ge 2 & x \ge 1 \\ f''(x) = 2 & \forall x \end{array} \longrightarrow \mathrm{M} = \frac{\max\{|f''(x)|\}}{2\min\{|f'(x)|\}} = \frac{2}{2\cdot 2} = \frac{1}{2}$$

Si M=1/2, puedo garantizar convergencia si $M\cdot e_0 < 1$, esto es, $e_0<2$.

La cota del error es entonces: $e_n \le \frac{1}{M}\cdot\left(Me_0\right)^{2^n} \le 2\cdot\left(\frac{e_0}{2}\right)^{2^n}$

2ª parte Problema 8 (aplicación)

¿Usar el método de Newton para calcular raíces cuadradas?

Un número real positivo a se representa en máquina (base 2) con una mantisa m en [1,2) y un exponente entero e:

$$a = m\cdot 2^e \quad\Rightarrow\quad \sqrt{a} = \begin{cases} \sqrt{m}\cdot 2^{\frac{e}{2}} & e = par \\ \sqrt{2m}\cdot 2^{\frac{e-1}{2}} & e = impar \end{cases}$$

ALGORITMO: dado (m,e) construyo $c = \begin{cases} m & e = par \\ 2m & e = impar \end{cases}, \qquad c \in [1,4)$

La nueva mantisa m', que cae dentro del intervalo [1,2) será $\sqrt{c}$.

Hallar la raíz cuadrada de **cualquier número máquina** = hacer la raíz cuadrada de un número c en el intervalo [1,4) (mantisa del resultado).

Estudiar el nº de iteraciones que nos garanticen un error menor que nuestra tolerancia (10^{-6}) y ver cómo influye la elección del x0 inicial.

Problema 8 (aplicación)

a) Al interesarnos las raíces cuadradas de c entre 1 y 4 (cuyo resultado está entre 1 y 2), una idea razonable es tomar como punto inicial x_0=1.5. Usando el resultado previo, demostrar que Newton converge y estimar el número de iteraciones para asegurar error relativo menor que 10^{-6}

¿Convergencia? Como M=1/2 → necesitamos que e_0<2 (para que M·e0< 1)

¿e_0? La solución va a estar entre 1 y 2 y arranco en x_0=1.5, luego $e_0 \leq 0.5$

Tenemos que M·e0 = 0.5·0.5 = 0.25<1 → el método de Newton converge

¿ Número de iteraciones ? $e_n \leq \frac{1}{M}(M \cdot e_0)^{2^n} < 10^{-6}$?

M=1/2 y $(M \cdot e_0) \leq 1/4$

$$¿2\left(\frac{1}{4}\right)^{2^n} < 10^{-6} ?$$ → n>3.39

(mínimo 4 iteraciones)

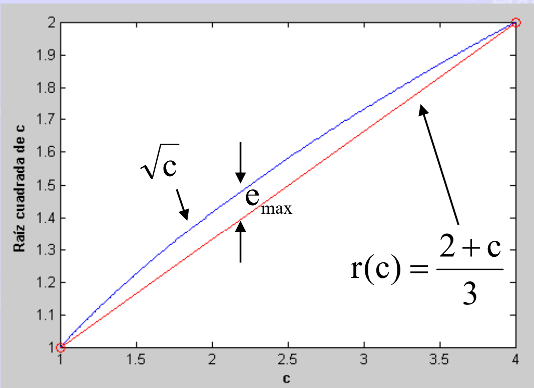
$$2\left(\frac{1}{4}\right)^{2^n}$$

Iter (n)	Cota E
1	1.25 10^{-1}
2	7.81 10^{-3}
3	3.05 10^{-5}
4	4.66 10^{-10}

Problema 8 (aplicación)

b) Para tratar de ahorrar iteraciones usaremos el valor inicial $x_0 = \frac{2+c}{3}$ que es la interpolación de $\sqrt{c}$ con una recta entre c=1 y c=4.

Estimar cota de e_0 y el número de iteraciones para 10^{-6} para este caso.



¿ Máximo error inicial posible e_0 ?

→ Máxima diferencia entre $\sqrt{c}$ y la recta.

$$e(c) = \sqrt{c} - \frac{2+c}{3} \Rightarrow \frac{de}{dc} = \frac{1}{2\sqrt{c}} - \frac{1}{3} = 0 \Rightarrow c = \frac{9}{4}$$

$$e_0 \leq \max\{e\} = error\left(\frac{9}{4}\right) = \frac{3}{2} - \frac{17}{12} = \frac{1}{12}$$

$$e_n \leq \frac{1}{M}(M \cdot e_0)^{2^n} < 10^{-6} ?$$

$$2 \cdot \left(\frac{1}{24}\right)^{2^n} < 10^{-6} ?$$

Iter (n)	x0=1.5	x0=(2+c)/3
1	1.25 e-1	3.47 e-3
2	7.81 e-3	6.03 e-6
3	3.05 e-5	1.82 e-11
4	4.66 e-10	

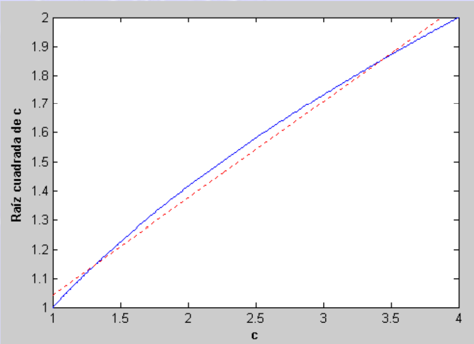
Problema 8 (no pedido)

Intentemos mejorar este resultado (no pedido en problema)

Modificación mínima de la hipótesis x0 que reduce e_0 a la mitad y podría (?) bajar a dos el número de iteraciones necesarias.

$$x_0 = \frac{2+c}{3} + \frac{1}{24}$$

Ahora el error inicial verifica $e_0 \le \frac{1}{24}$



$$e_n \le \frac{1}{M}(M \cdot e_0)^{2^n} < 10^{-6}\,? \Rightarrow 2 \cdot \left(\frac{1}{48}\right)^{2^n} < 10^{-6}\,?$$

Iter (n)	x_0=1.5	x_0=(2+c)/3	x_0= … + 1/24
1	1.25 e-1	3.47 e-3	8.68 e-4
2	7.81 e-3	6.03 e-6	3.77 e -7
3	3.05 e-5	1.82 e-11	
4	4.66 e-10		

Resumen: Algoritmo para raíz cuadrada

Sea un número a>0 ($a = m \cdot 2^e$), con un exponente e y mantisa m

1. Si e es par → e'=e/2 y c=m. Si e es impar e'=(e-1)/2 y c = 2m.
2. Se trata ahora de hallar m' = $\sqrt{c}$ (con c entre 1 y 4):

   a) Inicializar $x = \frac{2+c}{3} + \frac{1}{24}$

   b) Aplicar 2 veces x = 0.5 (x + c/x) (Newton para $f(x)=x^2-c$)

   c) m' = x (tras las 2 iteraciones)

```
>> c=(1:0.001:4);
>> x = (2+c)/3 + 1/24;
>> x = 0.5*(x+c./x);
>> x = 0.5*(x+c./x);
>> plot(c,abs(x-sqrt(c)))
```