

Capítulo 4

La resolución de ecuaciones y de sistemas de ecuaciones

Cuando en el instituto aprendemos a resolver ecuaciones solemos enfrentarnos a ecuaciones cuadráticas de la forma $ax^2 + bx + c = 0$, para cuyas soluciones existe una conocida fórmula general. Sin embargo, en el ámbito práctico de la física la mayoría de ecuaciones no son tan sencillas. Podemos encontrarnos con multitud de situaciones más complicadas como las siguientes:

- i) La ecuación a resolver contiene polinomios de grado mayor que 2, o funciones trascendentes (trigonométricas, exponenciales, etc.) con lo que no disponemos de una fórmula para su resolución. Por ejemplo, podemos tratar de comprender un muelle con rozamiento cuya ecuación (simplificada) viene dada por $\ddot{x} = -c\dot{x} - kx/m$ donde c es una constante de rozamiento. La solución de esta ecuación depende de los valores de c, k, m . Si por ejemplo se tiene $c^2 = 4k/m$ entonces la solución es $x(t) = e^{-ct/2}(A + Bt)$ para algunas (cualesquiera) constantes A, B . Si suponemos que partimos del reposo con elongación de 1 metro, esto es, $x(0) = 1, \dot{x}(0) = 0$ la solución es $x(t) = e^{-ct/2}(1 + ct/2)$. Pongamos que hemos medido $c = 0,01$ ¿Podríamos decir en qué momento o momentos se alcanza la elongación 1/2? Nos enfrentamos a la ecuación

$$e^{-0,005t}(1 + 0,005t) = 0,5,$$

que no es un polinomio cuadrático ni se le parece. Siempre podemos intentar dibujar la función con un paquete de software numérico y tratar de localizar visualmente el punto de corte de la función con el eje de abscisas pero ¿es eficaz este método cuando es posible que tengamos que resolver el problema un número grande de veces para distintos parámetros? Por ejemplo, pongamos que quiero comprender, en función del rozamiento c , el valor de t para el que la elongación es 1/2. Necesito resolver una ecuación parecida a la anterior para muchos valores de c (pongamos para 100 valores de c entre 0 y 1). ¿Estás dispuesto a dibujar 100 gráficas y mirar a ojo el punto en el que pasan por $x = 1/2$ o prefieres aprender a resolver esta clase de problemas numéricamente con un método eficaz?

- ii) Otro ejemplo nos viene dado por una partícula que parte del reposo en un plano inclinado uniforme sin rozamiento, cuyo ángulo θ cambia con el tiempo a velocidad constante ω . Se puede demostrar que después de t segundos la partícula ha recorrido

$$x(t) = -\frac{g}{2\omega^2} (\sinh \omega t - \sin \omega t) \text{ metros,}$$

donde aparece una función trigonométrica y otra hiperbólica. Si medimos que en 1 segundo ha recorrido 0,5 metros, ¿cuál es la velocidad de cambio de ángulo ω ? De nuevo nos vemos obligados a resolver una ecuación para la que no hay fórmulas sencillas.

- III) No tenemos una fórmula de la ecuación a resolver, sino que sabemos que tal función existe y solo podemos calcularla en algunos pocos puntos de nuestra elección. Por ejemplo, nos preguntamos por el peso que ha de añadirse a un globo de helio en un laboratorio para hacerlo caer al suelo desde 1 metro de altura de forma que la caída dure 2 segundos. En un contexto real en que tenemos una fuerza de rozamiento (desconocida), no existe una fórmula que en función del peso nos devuelva el tiempo que tarda el globo en caer, pues el rozamiento depende de la forma concreta del globo. ¿Hay una opción mejor que “ir probando”? Realizar experimentos es costoso en tiempo (y por lo tanto en dinero) y en materiales (o sea, en dinero), de modo que mejor realizamos el menor número posible de experimentos. ¿Cómo lo hacemos para no perder nuestros recursos innecesariamente?

Ilustramos este punto con otro ejemplo: en el diseño de los coches modernos se utiliza la simulación computacional para tratar de comprender el comportamiento del habitáculo frente a un choque lateral, frontal, etc. Posteriormente, se fabrican varios prototipos y se realizan pruebas “reales” en las que, con maniqués en lugar de seres humanos, se evalúan los efectos que unas u otras decisiones sobre el diseño hayan tenido en la cantidad de daño recibido por los ocupantes del vehículo. Pongamos que todos los componentes de un vehículo se han decidido ya y que nos preguntamos a qué velocidad máxima puede producirse un choque frontal sin que pase algo concreto, pongamos a qué velocidad máxima la cabeza del maniquí no se separa de su cuerpo. Realizaremos una serie de pruebas incluyendo algunas velocidades bajas y otras altas, observando cómo en las bajas todo va bien y en las altas no. Cada prueba lleva obviamente un gasto muy elevado, ¿qué podemos hacer para tratar de encontrar la velocidad máxima sin hacer más pruebas de las estrictamente necesarias? Nótese que aquí tenemos que encontrar el mayor x en un intervalo tal que $f(x) = 0$, donde $f(x)$ vale 0 hasta un momento y 1 de ahí en adelante. Se trata pues de encontrar la mayor solución de $f(x) = 0$. Este problema (así como otros muchos) se resuelve con los métodos estudiados en este capítulo.

- IV) La función que define nuestra ecuación puede calcularse en cualquier punto de nuestra elección pero solo a través de un proceso numérico. Este es un caso mezclado de los dos anteriores, y es difícil entenderlo sin estudiar antes la resolución numérica de ecuaciones diferenciales ordinarias que se verá en este curso. Por ejemplo, en el estudio del Sistema Solar comprendemos muy bien el movimiento de los planetas, que se pueden tabular con gran precisión con años de anterioridad, pero los asteroides pequeños son más difíciles de comprender. En la estupenda novela “El martillo de Dios” de Arthur C. Clarke, un equipo de astronautas trata de desviar un asteroide cuyo curso parece ser directo hacia nuestro querido planeta. Dado un “vector” aplicado al asteroide, los programas pueden calcular qué efectos tendrá el cambio en esa velocidad en su posición meses más tarde (cuando se encuentre próximo a la Tierra), esto es, dado un vector aplicado al asteroide se puede predecir con precisión a qué distancia pasará de la Tierra. Considerando 40,000 kilómetros una distancia de seguridad, ¿cómo podríamos calcular la magnitud del vector que debemos aplicar? Nuevamente nos encontramos ante el mismo problema que antes, solo que en lugar de experimentación en laboratorio tenemos resultados computacionales, que (podemos suponer) son más fáciles y baratos hoy en día que los que nos exigen experimentación en laboratorio. El impacto de un asteroide potencialmente devastador es una certeza, no una posibilidad, a la que la humanidad habrá de enfrentarse en el futuro si no es precedida por otra catástrofe comparable. ¡Afortunadamente muchas de las matemáticas necesarias para saber lo que hay que hacer (que no para poder hacerlo, harina de otro costal) son relativamente elementales!

4.1. Ecuaciones de una variable real

En esta sección suponemos que nos es dada una función $f : [a, b] \rightarrow \mathbb{R}$ de la que hemos de encontrar una raíz, esto es un ζ tal que $f(\zeta) = 0$. Comenzamos con el método más sencillo.

4.1.1. El método de bisección

Sea $f \in C[a, b]$ tal que $\alpha = f(a)$ y $\beta = f(b)$ tienen signos diferentes. El Corolario 1.1.8 nos garantiza que existe una solución de $f(x) = 0$ en el intervalo $[a, b]$, y es nuestra voluntad encontrarla. Para ello, calculemos el punto medio c y el valor de la función en dicho punto:

$$c = \frac{a+b}{2}, \quad \gamma = f(c).$$

Si γ y α tienen distinto signo, entonces de nuevo por el Corolario 1.1.8 hay una solución de $f(x) = 0$ en $[a, c]$. En caso contrario, por el mismo argumento, sabemos que hay una solución en $[c, b]$. Puede parecer que no hemos avanzado, pero si prestamos atención veremos que sí: antes buscábamos una solución en un intervalo de longitud $b - a$ y ahora la buscamos en un intervalo de la mitad de longitud. El método de bisección consiste en repetir este proceso hasta que se cumple una condición (*condición de parada*) suficientemente satisfactoria.

La principal propiedad del método de bisección es la siguiente. Llamemos c_1, c_2, \dots a los puntos medios calculados por el algoritmo. Entonces:

Teorema 4.1.1 *Si $f \in C[a, b]$ y $f(a)$ y $f(b)$ tienen signo diferente, entonces el método de bisección descrito arriba genera una sucesión c_n , $n \geq 1$, que aproxima una raíz ζ de f en el sentido de que*

$$|c_n - \zeta| \leq \frac{b-a}{2^n}, \quad n \geq 1.$$

DEMOSTRACIÓN. La longitud del intervalo de búsqueda se divide por dos en cada paso, luego tras n pasos la hemos dividido por 2^n . \square Vemos una ilustración gráfica de este método en la Figura 4.1.

Ha de notarse que la cota dada al error es siempre cierta, pero puede ser demasiado conservadora (esto es, puede suceder que las aproximaciones sean mucho mejores), y también puede suceder que c_n sea mejor aproximación que c_{n+1} para algún valor de n . Es más, bisección solo calcula la aproximación de una solución pero no nos dice nada sobre la existencia de otras soluciones, ni tenemos a priori control sobre a qué solución convergerá el método, por ejemplo, no podemos saber a priori si, de haber varias soluciones, bisección convergerá a la mayor, la menor, la más cercana al punto medio... no tenemos ninguna información sobre ese punto.

Ejemplo 4.1.2 *Calculemos una aproximación de $\sqrt{2}$ usando el método de bisección para la ecuación $f(x) = x^2 - 2$ en el intervalo $[0, 3]$. Escribimos en una tabla cada intervalo de búsqueda, el punto medio del intervalo, y el valor de la función en el punto medio.*

Describimos ahora, en código Matlab, el proceso descrito. Lógicamente debemos incluir una condición de parada razonable, en este caso, dada por el usuario con la precisión ϵ .

```
function c=biseccion(f,a,b,epsilon,M)
```

```
% Si los signos de f(a) y f(b) son diferentes se utiliza el metodo de biseccion
% para encontrar, con precision epsilon, una solucion de f(x)=0.
% Tambien incluimos un contador para hacer a lo mAs M iteraciones.
```

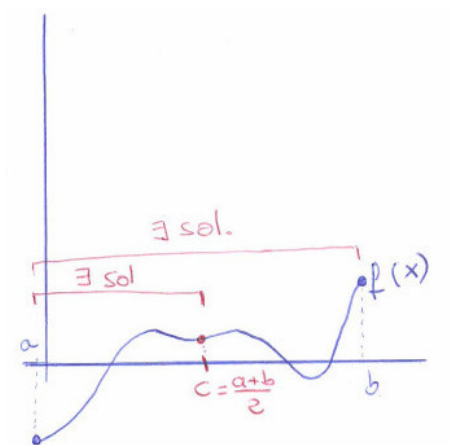


Figura 4.1: Ilustración de la primera iteración del método de bisección. Comenzamos en el intervalo de búsqueda $[a, b]$ y en una iteración reducimos el intervalo de búsqueda a $[a, c]$ que tiene la mitad de longitud.

$[a, b]$	$c = \frac{a+b}{2}$	$f(c)$
$[0, 3]$	1,5	0,25
$[0, 1,5]$	0,75	-1,4375
$[0,75, 1,5]$	1,125	-0,7344
$[1,125, 1,5]$	1,3125	-0,2773
$[1,3125, 1,5]$	1,4062	-0,0225
$[1,4062, 1,5]$	1,4531	0,1116
$[1,4062, 1,4531]$	1,4297	0,044

Cuadro 4.1: Método de bisección aplicado a $f(x) = x^2 - 2$ con $a = 0$, $b = 3$. La convergencia del punto medio al valor exacto (aprox. 1,4142) es algo lenta pero sucede de manera segura.

```

alpha=f(a);
beta=f(b);
if sign(alpha)==sign(beta)
    error('El signo de la funciOn debe cambiar en los extremos del intervalo!');
end
mierror=1;
contador=1;
while (mierror>epsilon) && (contador<=M)
    c=a+(b-a)/2;
    gamma=f(c);
    if sign(gamma)~=sign(alpha)
        b=c;
    else
        a=c;
        alpha=gamma;
    end
end

```

```

mierror=max(abs(gamma),b-a);
contador=contador+1;
end
if contador==(M+1)
error('Se ha alcanzado el numero maximo de iteraciones');
end

```

Si bien el método de bisección es el de elección en el caso de que la función sea fácil de calcular y conozcamos dos puntos en que los valores tienen distinto signo, no siempre poseemos esta información. Pongamos que en el problema $f(x) = 0$ que estamos tratando de resolver, $f(x)$ tiene la forma aproximada de la función $g(x) = x^2 - \epsilon$. Por mucho que busquemos un punto en que f sea negativa, lo tenemos muy difícil si $\epsilon \ll 1$. En el caso extremo $\epsilon = 0$, es directamente imposible.

Además, el método de bisección depende fuertemente de la propiedad de que los números reales están bien ordenados. Esto no se cumple ni en el caso de los complejos ni en el caso de que estemos tratando de resolver sistemas de ecuaciones. Por ello necesitamos otras alternativas.

4.1.2. Métodos de punto fijo

Un punto fijo de una función g es un punto tal que $g(p) = p$. Observamos que todo problema de “encontrar un punto fijo” puede escribirse como “resolver la ecuación $g(p) - p = 0$ ”, con lo que encontrar puntos fijos debería ser “más sencillo que” resolver ecuaciones. No obstante, si quiero resolver $f(x) = 0$ puedo en su lugar buscar un punto fijo de la función $g(p) = p - f(p)$, con lo que descubrimos que buscar puntos fijos y resolver ecuaciones son en realidad problemas equivalentes: un método para uno de esos problemas nos proporciona inmediatamente un método para el otro.

Resulta que los métodos de punto fijo son frecuentemente más efectivos y fáciles de estudiar, por lo que son una opción muy eficaz para resolver ecuaciones. Comenzamos con un resultado de existencia y, bajo ciertas condiciones, unicidad:

Teorema 4.1.3 *Sea $g \in C[a, b]$, y supongamos que $g(x) \in [a, b]$ para todo $x \in [a, b]$. Entonces, g tiene un punto fijo en $[a, b]$. Adicionalmente, si g es derivable en (a, b) y $|g'(x)| < 1$ para $x \in (a, b)$, entonces el punto fijo es único.*

DEMOSTRACIÓN. Para la primera parte, si $g(a) = a$ o $g(b) = b$, ya tenemos lo que queremos. En otro caso, necesariamente $g(a) > a$ y $g(b) < b$ con lo que la función $h(x) = g(x) - x$ es continua, positiva en $x = a$ y negativa en $x = b$, luego por el Corolario 1.1.8 tiene un cero, esto es, g tiene un punto fijo, en (a, b) .

Para la segunda parte, si hubiese dos puntos fijos p, q por el Teorema 1.1.9 habría un número $c \in (a, b)$ tal que

$$g'(c) = \frac{g(p) - g(q)}{p - q} = 1,$$

lo que contradice las hipótesis, luego no pueden existir dos puntos fijos diferentes. □

El método (o iteración) de punto fijo consiste en tomar un candidato inicial $p_0 \in [a, b]$ y generar la secuencia

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

esto es calcular $p_0, g(p_0), g(g(p_0)), \dots$. Si la secuencia converge a algún punto $p \in [a, b]$ y asumiendo que g es continua tenemos por el Teorema 1.1.4

$$p = \lim_{n \rightarrow \infty} p_n = \lim_{n \rightarrow \infty} g(p_{n-1}) = g(p),$$

luego p es un punto fijo de g .

El siguiente resultado (y su corolario) nos da una herramienta para garantizar la convergencia de este método en algunos casos.

Teorema 4.1.4 *Sea $g \in C[a, b]$ derivable en (a, b) , y supongamos que $g(x) \in [a, b]$ para todo $x \in [a, b]$. Si existe una constante positiva $k < 1$ tal que $|g'(x)| < k$ para $x \in (a, b)$, entonces la sucesión generada por la iteración de punto fijo converge al único punto fijo de g en $[a, b]$. Es más, $|p_n - p| \leq k^n |p_0 - p|$.*

DEMOSTRACIÓN. Por el Teorema 4.1.3 sabemos que el punto fijo existe y es único. Por otro lado, el Teorema 1.1.9 garantiza que (para algunos $\zeta_n \in (a, b)$):

$$|p_n - p| = |g(p_{n-1}) - g(p)| = |g'(\zeta_n)| |p_{n-1} - p| \leq k |p_{n-1} - p|,$$

con lo que por inducción $|p_n - p| \leq k^n |p_0 - p|$. Como $k < 1$, tenemos que $|p_n - p|$ converge a 0, lo que es equivalente a que p_n converge a p , como queríamos. □

El término de error contiene un factor desconocido ($|p - p_0|$), que obviamente podemos acotar por el máximo de los valores $b - p_0$, $p_0 - a$. También podemos acotar de forma concreta el error calculando una única iteración:

Corolario 4.1.5 *En las hipótesis del Teorema 4.1.4, tenemos también*

$$|p_n - p| \leq \frac{k^n}{1 - k} |p_1 - p_0|.$$

DEMOSTRACIÓN. Siguiendo el mismo procedimiento que en la demostración del Teorema 4.1.3, tenemos

$$|p_{n+1} - p_n| = |g(p_n) - g(p_{n-1})| \leq k |p_n - p_{n-1}| \leq \dots \leq k^n |p_1 - p_0|.$$

Por lo tanto, para $m > n \geq 1$ tenemos:

$$\begin{aligned} |p_m - p_n| &\leq |p_m - p_{m-1}| + \dots + |p_{n+1} - p_n| \\ &\leq k^{m-1} |p_1 - p_0| + \dots + k^n |p_1 - p_0| \\ &= k^n (1 + \dots + k^{m-n-1}) |p_1 - p_0| \\ &= k^n \frac{1 - k^{m-n}}{1 - k} |p_1 - p_0|, \end{aligned}$$

y tomando el límite cuando $m \rightarrow \infty$ se obtiene el resultado deseado. □

Ejemplo 4.1.6 *Resolver la ecuación $x^2 - 2 = 0$ equivale a encontrar un punto fijo de $g(x) = x + 1 - x^2/2$. Para $x \in [1, 1, 1, 9]$ se tiene que $g(x) \in [1, 1, 1, 9]$ y además $g'(x) = 1 - x$ tiene valor absoluto acotado por 1, con lo que se cumplen las hipótesis del Teorema 4.1.3 con $k = 0,9$. Consecuentemente, la iteración de punto fijo aplicada a $g(x)$ con punto inicial cualquiera en $[1, 1, 1, 9]$ ha de converger a $\sqrt{2}$ y cumplir la cota del Corolario 4.1.5. Por ejemplo, tomando $x_0 = 1,1$ generamos la sucesión:*

$$1,1 \quad 1,4950 \quad 1,3775 \quad 1,4288 \quad 1,4081 \quad 1,4167 \quad 1,4132 \quad 1,4146 \quad 1,4140 \quad 1,4143$$

que aparentemente converge en efecto a $\sqrt{2}$, como de hecho sabemos que sucede.

Dada una ecuación $f(x) = 0$, hay muchas maneras de escribir un problema de punto fijo $g(p) = p$ tal que la solución a la ecuación y la solución al problema de punto fijo sean equivalentes. De acuerdo con los resultados vistos, lo óptimo sería elegir una función g tal que g' sea lo más pequeña (en valor absoluto) posible. Una implementación en Matlab del método de punto fijo sería por ejemplo:

```
function p=puntofijo(g,p,epsilon,M)
% Dada la funcion g y el punto inicial p, realiza el metodo del punto fijo
% hasta alcanzar una precision epsilon. El programa realiza un maximo de M iteraciones
contador=1;
mierror=1;
while (mierror>epsilon) && (contador<=M)
    contador=contador+1;
    gp=g(p);
    mierror=abs(gp-p);
    p=gp;
    contador=contador+1;
end
if contador==(M+1)
    error('Se ha alcanzado el numero maximo de iteraciones');
end
```

4.1.3. Métodos de Newton y de la secante

El método de Newton (o Newton-Raphson) es otra de las grandes ideas que uno de los mayores genios de la historia nos dejó. Para explicarlo, aparte de su conocida interpretación geométrica, supongamos que $f \in C^2[a, b]$ y que $x \in (a, b)$ con $f(x) = 0$. Por el Teorema 1.3.1, sabemos que para cada $x_0 \in [a, b]$ tenemos

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(\zeta_x)}{2}(x - x_0)^2,$$

para algún $\zeta_x \in (a, b)$. Supongamos ahora que x_0 está lo suficientemente cerca de p como para poder despreciar $(p - x_0)^2$. Evaluando entonces la expresión anterior en $x = p$ tenemos:

$$0 = f(p) \approx f(x_0) + f'(x_0)(p - x_0),$$

de donde

$$p \approx x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Esto es, a partir de una aproximación x_0 de la raíz p , obtenemos una aproximación (siendo optimistas, una mejor aproximación) de la raíz p . El método de Newton consiste en repetir este procedimiento, de forma que se genera la sucesión

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

Nótese que si $f'(x_n) = 0$ para algún n , no podemos definir la siguiente iteración. En ese caso concluimos que hemos elegido mal x_0 . Lo realmente sorprendente de este método es que en numerosas ocasiones converge a una solución muy rápidamente, incluso si x_0 está muy lejos de ella al principio. Lo que podemos afirmar es el siguiente resultado.

Teorema 4.1.7 *Sea $f \in C^2[a, b]$. Si $p \in [a, b]$ es tal que $f(p) = 0$ y $f'(p) \neq 0$, entonces existe $\delta > 0$ tal que el método de Newton genera una sucesión definida para todo $n \geq 0$ y que converge a p , siempre que x_0 se elija en $(p - \delta, p + \delta)$.*

DEMOSTRACIÓN. Consideremos $g(x) = x - f(x)/f'(x)$. Como $f'(p) \neq 0$ y f' es continua, podemos suponer (tras pasar a un intervalo $[p - \delta_1, p + \delta_1]$ lo bastante pequeño) que $f' \neq 0$, con lo que g está bien definida en ese intervalo. Además,

$$g'(x) = \frac{f(x)f''(x)}{f'(x)^2},$$

y como $f \in C^2[a, b]$ tenemos que $g \in C^1[p - \delta_1, p + \delta_1]$. Dado que $f(p) = 0$, tenemos $g'(p) = 0$. Sea $k \in (0, 1)$ un número cualquiera. Podemos encontrar $\delta < \delta_1$ tal que $|g'(x)| < k$ para $x \in (p - \delta, p + \delta)$. Veamos ahora que $g([p - \delta, p + \delta]) \subseteq [p - \delta, p + \delta]$. En efecto, si $|x - p| < \delta$, por el Teorema 1.1.9 tenemos que

$$|g(x) - g(p)| = |g'(\zeta)| |x - p| \leq k\delta < \delta.$$

Todas las hipótesis del Teorema 4.1.3 se cumplen, con lo que existe un único punto fijo (una única solución de $f(x) = 0$) en el intervalo $(p - \delta, p + \delta)$, y la sucesión generada por el método de Newton converge a dicha solución. El teorema queda así demostrado.

□Vemos una ilustración gráfica de este método en la Figura 4.2.

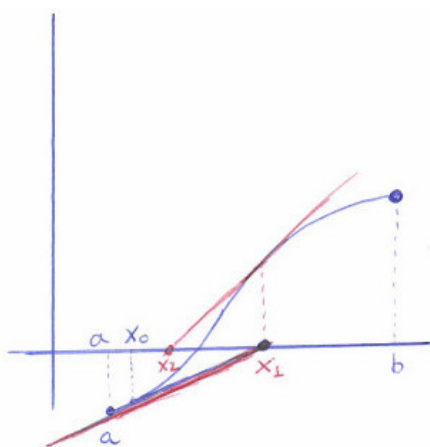


Figura 4.2: Ilustración de las dos primeras iteraciones del método de Newton.

Ejemplo 4.1.8 Utilizar las sucesivas iteraciones del método de Newton para resolver $f(x) = x^2 - 2$ con $x_0 = 2$.
Solución: La fórmula de la iteración es:

$$x \mapsto x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - 2}{2x} = \frac{x + \frac{2}{x}}{2}.$$

El cálculo de tres iteraciones produce:

$$2 \quad 1,5 \quad 1,4167 \quad 1,4142,$$

alcanzándose rápidamente la convergencia a $\sqrt{2}$. Nótese que

Ejemplo 4.1.9 Calcular las primeras tres iteraciones del método de Newton para resolver $f(x) = 0$ donde $f(x) = x^6 - x - 1$. Tomar como punto inicial de la iteración $x_0 = 2$. *Solución: la iteración de Newton tiene la forma*

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^6 - x_i - 1}{6x_i^5 - 1} = \frac{5x_i^6 + 1}{6x_i^5 - 1}.$$

Con ello tenemos:

$$x_0 = 2, \quad x_1 = 1,6806, \quad x_2 = 1,4307, \quad x_3 = 1,2549,$$

que era lo que se pedía.

Una implementación en Matlab del método de Newton sería por ejemplo:

```
function x1=newton(f,fp,c,epsilon,M)
% Calcula una solución aproximada de f(x)=0 con el metodo de Newton
% se comienza la iteracion en x0=c y se pone un limite de iteraciones
% Tambien se pide al usuario la precision epsilon.
% se pide fp la derivada de f
x1=c;
mierror=1;
contador=1;
while (mierror>epsilon) && (contador<=M)
    x0=x1;
    x1=x1-f(x1)/fp(x1);
    mierror=abs(x0-x1);
    contador=contador+1;
end
if contador==(M+1)
    error('Se ha alcanzado el numero maximo de iteraciones');
end
```

Pese a todas las virtudes del método de Newton, en ocasiones su uso resulta complicado. El motivo es que necesitamos conocer $f'(x)$ y no siempre tenemos una fórmula para esta función (basta considerar algunos de los ejemplos mencionados en la introducción del capítulo). Una manera de evitar este problema es sustituir en el cálculo el término $f'(x_n)$ por su aproximación

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}},$$

lo que transforma la iteración de Newton dando la nueva fórmula

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

Así se define el llamado *método de la secante*, que requiere para ser inicializado dos valores aproximados de la raíz x_0, x_1 (o bien x_0 y $f'(x_0)$) para poder calcular x_1 usando el método de Newton).

Vemos una ilustración gráfica de este método en la Figura 4.3.

Ejemplo 4.1.10 Las sucesivas iteraciones del método de la secante para resolver $f(x) = x^2 - 2$ con $x_0 = 2, x_1 = 1$ producen:

$$2 \quad 1 \quad 1,333 \quad 1,4286 \quad 1,4138 \quad 1,4142,$$

alcanzándose rápidamente (aunque algo más lentamente que con el método de Newton) la convergencia a $\sqrt{2}$.

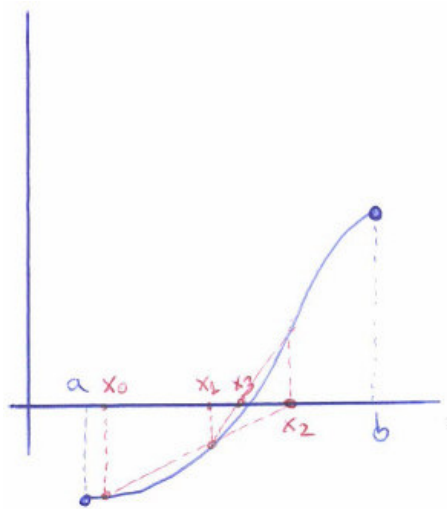


Figura 4.3: Ilustración de las dos primeras iteraciones del método de la secante.

Una posible implementación en Matlab de este método sería:

```
function x1=secante(f,c0,c1,epsilon,M)
% Calcula una solución aproximada de f(x)=0 con el método de la secante
% se comienza la iteración en x_0=c0, x_1=c1 y se pone un límite de iteraciones
% También se pide al usuario la precisión epsilon.
x0=c0;
fx0=f(x0);
x1=c1;
fx1=f(x1);
mierror=1;
contador=1;
while (mierror>epsilon) && (contador<=M)
    xaux=x1;
    x1=x1-fx1*(x1-x0)/(fx1-fx0)
    mierror=abs(x0-x1);
    contador=contador+1;
    x0=xaux;
    fx0=fx1;
    fx1=f(x1);
end
if contador==(M+1)
    error('Se ha alcanzado el número máximo de iteraciones');
end
```

Otros métodos inspirados en estos que hemos mencionado (por ejemplo el método de Regula Falsi) son de uso menos frecuente y no los estudiaremos en este curso.

Finalmente señalamos que, aunque el método de Newton y el método de la secante han sido explicados para funciones de una variable real, también son válidos para funciones de una variable compleja.

4.2. La resolución de sistemas de ecuaciones no-lineales

En muchas ocasiones tenemos que resolver sistemas de ecuaciones no-lineales. Se trata ésta de una tarea nada sencilla en según qué casos. Además, lo más probable es que el estudiante no se haya enfrentado a ella anteriormente más que en algunos casos extremadamente sencillos. Pongamos que conocemos la trayectoria de dos cuerpos en el sistema solar y que, tras simplificar enormemente el problema, concluimos que uno de ellos (la Tierra) sigue una trayectoria aproximadamente esférica que en las unidades apropiadas y fijándonos únicamente en el plano de la elíptica podemos escribir como

$$x^2 + y^2 = 1, \quad \text{esto es, una circunferencia de radio 1UA.}$$

Ahora consideremos un cometa en trayectoria parabólica cuya órbita obedece a la ecuación

$$y^2 + x^2 + 2xy - y = 1, \quad \text{que es una parábola como se puede ver al dibujarla}$$

La pregunta que nos hacemos es muy simple: ¿en qué punto el cometa atraviesa la órbita de la Tierra?

La pregunta que acabamos de plantear nos demanda simplemente resolver el sistema de ecuaciones

$$\begin{cases} x^2 + y^2 - 1 = 0 \\ y^2 + x^2 + 2xy - y - 1 = 0 \end{cases}$$

esto es resolver $f(v) = 0$ donde $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, un sistema de dos ecuaciones con dos variables.

Otro ejemplo, menos atractivo desde el punto de vista filosófico pero mucho más desde el punto de vista industrial, viene dado por el movimiento de los brazos robóticos en las grandes fábricas. En el problema de los brazos robóticos (o en el mismo problema de la intersección de las órbitas si lo consideramos en tres coordenadas espaciales) fácilmente nos requiere resolver sistemas del orden de 20, 30 o 100 ecuaciones no-lineales. ¿Qué métodos podemos usar?

La teoría en este caso es más compleja que en el caso de una variable. Obviamente son necesarios los conceptos de continuidad y derivabilidad para funciones de varias variables reales, y además las demostraciones de los resultados requieren un esfuerzo mucho mayor que en el caso anterior. Por ello indicamos simplemente algunos métodos que, combinados, son suficientes para resolver una considerable cantidad de problemas, sin centrarnos demasiado en su justificación teórica.

4.2.1. El método de Newton para varias variables

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ (o $f : \mathbb{C}^n \rightarrow \mathbb{C}^n$). El método de Newton consiste en tomar $v_0 \in \mathbb{R}^n$ (una aproximación inicial a la solución) y considerar la iteración:

$$v_{n+1} = v_n - Jf(v_n)^{-1}f(v_n),$$

siendo $Jf(v_n)$ la matriz Jacobiana de f en v_n . Al igual que en el caso de una variable, tenemos un teorema que nos garantiza la convergencia bajo ciertas condiciones. Mencionamos aquí una versión sencilla de dicho teorema, debido a Kantorovich.

Teorema 4.2.1 *Si $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ es de clase C^2 y si $v \in \mathbb{R}^n$ con $f(v) = 0$, $\det(Jf(v)) \neq 0$, entonces existe un entorno de v tal que para todo v_0 en dicho entorno la secuencia de Newton está bien definida y converge cuadráticamente a v .*

Lamentablemente es difícil encontrar un punto v_0 para el que este teorema (o sus hermanos mayores) nos garanticen la convergencia. Afortunadamente, el método de Newton funciona en muchísimos casos prácticos (más allá de las garantías del Teorema, esto es, para “aproximaciones” v_0 bastante groseras). Una implementación posible para el método de Newton para sistemas de ecuaciones sería como sigue.

```
function x1=newtonvariasvariables(f,Jf,c,epsilon,M)
% Calcula una solucion aproximada de f(x)=0 con el metodo de Newton
% se comienza la iteracion en x_0=c y se pone un limite de iteraciones
% Tambien se pide al usuario la precision epsilon.
% se pide Jf que es la jacobiana de f.
x1=c;
mierror=1;
contador=1;
while (mierror>epsilon) && (contador<=M)
    x0=x1;
    x1=x1-Jf(x1)\f(x1);
    mierror=norm(x0-x1);
    contador=contador+1;
end
if contador==(M+1)
    error('Se ha alcanzado el numero maximo de iteraciones');
end
```

Ejemplo 4.2.2 Calcular la fórmula que tiene la iteración de Newton para resolver el sistema de ecuaciones

$$\begin{cases} x^4 + y^3 = 0 \\ xy = 1 \end{cases}$$

Solución: sea

$$f \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x^4 + y^3 \\ xy - 1 \end{pmatrix}$$

La iteración de Newton tiene la fórmula

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} - Df \begin{pmatrix} x \\ y \end{pmatrix}^{-1} f \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} 4x^3 & 3y^2 \\ y & x \end{pmatrix}^{-1} \begin{pmatrix} x^4 + y^3 \\ xy - 1 \end{pmatrix}.$$

La inversa de la matriz es:

$$\begin{pmatrix} 4x^3 & 3y^2 \\ y & x \end{pmatrix}^{-1} = \frac{1}{4x^4 - 3y^3} \begin{pmatrix} x & -3y^2 \\ -y & 4x^3 \end{pmatrix},$$

con lo que la iteración tiene la expresión

$$\begin{aligned} \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \end{pmatrix} - \frac{1}{4x^4 - 3y^3} \begin{pmatrix} x & -3y^2 \\ -y & 4x^3 \end{pmatrix} \begin{pmatrix} x^4 + y^3 \\ xy - 1 \end{pmatrix} = \\ \begin{pmatrix} x \\ y \end{pmatrix} - \frac{1}{4x^4 - 3y^3} \begin{pmatrix} x^5 + xy^3 - 3xy^3 + 3y^2 \\ -x^4y - y^4 + 4x^4y - 4x^3 \end{pmatrix}. \end{aligned}$$

4.2.2. Método del gradiente o del descenso más rápido

Si bien el método de Newton funciona con mucha frecuencia, en muchos problemas sucede que la convergencia es solo posible cuando uno elige muy bien la aproximación inicial. Es por ello que a veces conviene usar un método más lento pero más seguro, al menos que proporcione alguna información sobre la función. El método del gradiente cumple a la perfección con este papel: no siempre nos devuelve un cero del sistema, pero sí que nos proporciona una secuencia de iteraciones x_0, x_1, \dots con la propiedad de que la norma de $f(x_n)$ es decreciente, y en el contexto teórico se detiene cuando esta norma alcanza un mínimo local. Obviamente, un mínimo local de $\|f(x)\|^2$ no tiene por qué ser una solución de $f(x) = 0$, pero el recíproco es cierto: toda solución de $f(x) = 0$ es un mínimo global de $\|f(x)\|^2$. Además, el método del gradiente puede ser utilizado para minimizar (o maximizar) funciones cualesquiera, independientemente de que correspondan a los ceros de un sistema o no. Este problema (la minimización de funciones complicadas) es de importancia esencial en Física, habiendo llevado a la comprensión de muchos problemas, por ejemplo, el de la estructura cristalina de algunos materiales.

Comenzamos discutiendo en general el método del gradiente. Dada $g : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable, y dado $v_0 \in \mathbb{R}^n$, podemos preguntarnos cuál es la dirección de \mathbb{R}^n tal que f decrece con mayor rapidez en dicha dirección, si es que existe una (y si es que es única). Un poco de cálculo de varias variables demuestra rápidamente que la dirección, de existir, es única, y viene dada precisamente por el opuesto del gradiente en el punto. Esto es, la dirección de más rápido descenso es precisamente $-\nabla g(v_0)$. Un método apropiado para minimizar $g(v)$ parece ser por tanto el tomar v_0 inicial (próximo al mínimo, si es posible) y mover v_0 en la dirección de $\nabla g(v_0)$ una cantidad apropiada, repitiendo el proceso después. Por lo tanto la iteración ha de ser:

$$v_{n+1} = v_n - \alpha \nabla g(v_n), \quad \text{para un valor apropiado de } \alpha > 0.$$

La manera exacta de definir α depende del autor, pero una forma muy apropiada es primero definir $h(\alpha) = g(v_n - \alpha \nabla g(v_n))$, encontrar por tanteo $\alpha_* \in [0, 1]$ tal que $h(\alpha) < h(0) = g(v_n)$, interpolar la función h por un polinomio cuadrático en $\alpha = 0, \alpha_*/2, \alpha_*$, y calcular el α que minimiza dicho polinomio, bien en $[0, \alpha_*]$, bien más allá si el valor de h disminuye más. En otras palabras, después de interpolar nos quedamos o bien con el valor en que se anula la derivada del polinomio cuadrático o bien con el α_* que hemos encontrado. Todos estos procesos han de ser controlados con la tolerancia numérica deseada por el usuario.

Un vistazo al capítulo 2 nos da una fórmula concreta para hallar α una vez que hayamos encontrado α_* : la tabla de valores que ha de cumplirse es la mostrada en la Tabla 4.2.

α	0	$\alpha_*/2$	α_*
$h(\alpha)$	$g(v_n)$	$g(v_n + (\alpha_*/2)\nabla g(v_n))$	$g(v_n + \alpha_*\nabla g(v_n))$

Cuadro 4.2: Tabla de valores que cumple el polinomio de grado 2 que interpola a $h(\alpha)$ en los 3 nodos descritos.

Por lo tanto tenemos el polinomio $p(\alpha)$ interpolante de h :

$$p(\alpha) = g(v_n) \frac{(\alpha - \alpha_*/2)(\alpha - \alpha_*)}{\alpha_*^2/2} + g(v_n + (\alpha_*/2)\nabla g(v_n)) \frac{\alpha(\alpha - \alpha_*)}{-\alpha_*^2/4} + g(v_n + \alpha_*\nabla g(v_n)) \frac{\alpha(\alpha - \alpha_*)}{\alpha_*^2/2}.$$

La derivada de este polinomio se anula cuando $p'(\alpha) = 0$, lo que viene a ser:

$$\alpha = \alpha_* \frac{g(v_n + \alpha_*\nabla g(v_n)) - 4g(v_n + (\alpha_*/2)\nabla g(v_n)) + 3g(v_n)}{4g(v_n + \alpha_*\nabla g(v_n)) - 8g(v_n + (\alpha_*/2)\nabla g(v_n)) + 4g(v_n)}.$$

El α que tomamos como óptimo es, de entre este último valor y α_* , en el que h sea menor. Con todo ello, podemos afirmar lo siguiente.

Teorema 4.2.3 *El método del gradiente tal y como ha sido descrito más arriba, tomando como entradas una función $g : \mathbb{R}^n \rightarrow \mathbb{R}$ diferenciable y un vector $v_0 \in \mathbb{R}^n$, genera una sucesión v_1, v_2, \dots con la propiedad de que $g(v_{n+1}) \leq g(v_n)$ para $n \geq 1$, deteniéndose (con la consideración de la precisión) en un mínimo local de g , o divergiendo a $-\infty$.*

Al igual que en el caso de Newton, puede darse el caso de que calcular el gradiente sea muy complicado. En ese caso, sustituiremos el valor del gradiente por el de su aproximación estudiada en el capítulo 3. Una implementación posible del método del gradiente (con esta última variación incluida) se muestra a continuación.

```
function v=gradiente(g,v0,epsilon,M)
% Encuentra un minimo local de g utilizando el metodo del gradiente
% comenzando en v0 con maximo numero de pasos M
% y usando epsilon para la precision
n=length(v0);
v=v0;
mierror=1;
contador=1;
hcero=g(v);
while (mierror>epsilon) && (contador<=M)
    gradiente=zeros(n,1);
    for j=1:n
        ej=zeros(n,1);
        ej(j)=1;
        gradiente(j)=(g(v+epsilon*ej)-g(v-epsilon*ej))/2/epsilon;
    end
    mierror=norm(gradiente);
    gradiente=gradiente/mierror;
% Con eso queda definido el gradiente (unitario) en el punto actual v
% Ahora buscamos alpha_* con la propiedad descrita arriba.
    alphastar=1;
    halphastar=g(v-gradiente);
    otrocontador=1;
    while (halphastar>hcero) && (otrocontador<=1/epsilon)
        alphastar=alphastar/2;
        halphastar=g(v-alphastar*gradiente);
        otrocontador=otrocontador+1;
    end
% Con eso queda definido alpha_*. Ahora buscamos el valor optimo de alpha:
    numerador=halphastar-4*g(v-alphastar/2*gradiente)+3*hcero;
    denominador=4*halphastar-8*g(v-alphastar/2*gradiente)+4*hcero;
    alphas=halphastar*n numerador/denominador;
    halphas=g(v+alphas*gradiente);
    if halphas<halphastar
        alphastar=alphas;
        hcero=halphas;
    else
        hcero=halphastar;
    end
% Con esto, alphastar es el valor optimo para el tamanno del paso
    contador=contador+1;
    v=v-alphastar*gradiente;
    hcero
end
```

Ahora, dado $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, podemos usar el método del gradiente para buscar mínimos locales de $g(v) = \|f(v)\|^2$, con la esperanza de que de hecho lo que encontremos sean mínimos globales (esto es, soluciones de $f(v) = 0$) y la respuesta del gradiente puede ser mejorada grandemente con el método de Newton mediante 2 ó 3 iteraciones finales.

4.3. Exercises. Solving $f(x) = 0$

Exercise 4.1

Use the bisection method to find solutions accurate to within 10^{-2} for $x^3 - 7x^2 + 14x - 6 = 0$ on each of the intervals $[0, 1]$, $[1, 3, 2]$ and $[3, 2, 4]$.

Exercise 4.2

Use the bisection method to find a solution to within 10^{-2} for $x = \tan x + 1$.

Exercise 4.3

Let $f(x) = (x+2)(x+1)^2x(x-1)^3(x-2)$. To which zero of f does the bisection method converge when applied to the following intervals?

$$[-1, 5, 2, 5], \quad [-0, 5, 2, 4], \quad [-0, 5, 3], \quad [-3, -0, 5].$$

If instead of the bisection method we consider the secant method, do we get to the same or to different zeros?

Exercise 4.4

If we want to solve $f(x) = 0$ and we know $f(-1) = -4$, $f(1) = 10$, how many iterations of the bisection method are needed to get a precision of 10^{-6} ?

Exercise 4.5

Find an approximation to $\sqrt{3}$ to within 10^{-3} using the bisection method (hint: consider $f(x) = x^2 - 3$). Do the same but this time using Newton's method.

Exercise 4.6

Write down a general formula for the Newton iteration for approximating \sqrt{c} for $c > 0$.

Exercise 4.7

The criterion for stopping Newton's method is sometimes taken as "the difference between two iterations is sufficiently small". Can it happen that for a sequence p_n of real numbers, $|p_n - p_{n+1}| \rightarrow 0$ (that is, the difference becomes arbitrarily small) but still p_n does not converge?

Exercise 4.8

Use bisection and Newton's method to find approximate solutions of the following problems in the given intervals:

$$\begin{aligned}x^3 - 2x^2 - 5 &= 0, & [1, 4] \\x &= \cos x, & [0, \pi/2] \\x^3 + 3x^2 &= 1, & [-3, -2] \\x - 0,8 &= 0,2 \sin x, & [0, \pi/2] \\x^2 - 2xe^{-x} + e^{-2x} &= 0, & [0, 1] \\e^{6x} + 3(\log 2)^2 e^{2x} - (\log 8)e^{4x} - (\log 2)^3 &= 0, & [-1, 0]\end{aligned}$$

Exercise 4.9

Find approximations to all the solutions of the given equations in the given intervals, if possible.

$$\begin{aligned}x^2 \sin(1/x) &= 0, & [0, \pi] \\2x \cos x - (x - 2)^2 &= 0, & [0, 4] \\(x - 2)^2 &= \log x, & [1, 5] \\e^x &= 3x^2, & [0, 5] \\\sin x &= e^x, & [0, 7] \\x^3 - 3x^2 &= -4, & [-3, 3] \\25x^3 - 300x^2 + 900x &= 2, & [-1, 7] \\100x^3 + 900x^2 &= -27, & [-10, 2] \\230x^4 + 18x^3 + 9x^2 - 221x - 9 &= 0, & [-5, 5].\end{aligned}$$

Compare your results to those of Matlab's `fsolve` and `fzero`.

Exercise 4.10

Produce a Matlab function `closest` which on input $(a, b) \in \mathbb{R}^2$ produces the point(s) in the graph of $y = x^2$ which is closest to (a, b) .

Exercise 4.11

The function $f(x) = \log(x^2 + 1) - e^{0,4x} \cos(\pi x)$ has an infinite number of zeros. Approximate the only negative zero, the four smallest positive zeros and the 25-th smallest positive zero of f .

Exercise 4.12

Write down a generic program `extremalzeros` which on input f and $[a, b]$ attempts to find the greatest and the smallest zero of f in $[a, b]$.

Exercise 4.13

Write down a Matlab function `Newtonseveralvariables` for Newton's method in several variables. Use your program to compute as many solutions as possible for each of the following systems:

$$\begin{cases} -x(x+1) + 2y = 18 \\ (x-1)^2 + (y-6)^2 = 25 \end{cases}, \quad \begin{cases} 5x^2 - y^2 = 0 \\ y = \frac{\sin x + \cos y}{4} \end{cases},$$

$$\begin{cases} x^2 + 2y^2 - y - 2z = 0 \\ x^2 - 8y^2 + 10z = 0 \\ x^2 - 7yz = 0 \end{cases}, \quad \begin{cases} \sin(4\pi xy) - 2y - x = 0 \\ \frac{4\pi-1}{4\pi} (e^{2x} - e) + 4ey^2 - 2ex = 0 \end{cases}$$

Exercise 4.14

Write down a program which receives a system of equations and an initial point and performs the Gradient Descent method to find a solution of the system. Apply it to the systems of Exercise (4.13). In matlab, a similar function is `fminunc`. Learn how to use it and compare the results with the ones you got.

Exercise 4.15

Find out the angle and speed of a parabolic trajectory from the center of the bottom of a $5m$ deep and $2m$ wide pit, if the impact with the floor must happen at $1m$ distance from the pit, if an impact angle of $-\pi/5$ is wanted. **Answer:** `speed= 10,7944m/s`, `angle= 1,4476 rad`.

Exercise 4.16

A bullet has been shot with unknown speed and shooting angle, but we have checked that the impact against a wall situated 100 away from the shooting point took place at a height of $34,5$ meters, and with an angle of $\pi/10$. The weapon was found to be one whose cannon is $20cm$ long. Assuming no air resistance or wind, and knowing from the kind of bullet that the initial speed was $800m/s$, give an approximate value for the height of the shooter. You can assume that the shooter was standing up and the arm whose hand pulled the trigger was straight. You can also assume that the dimensions of the shooter's body are those of the famous Da Vinci's Man of Vitrubio. **Answer:** `1,877m`.

Exercise 4.17

A robot consisting of 3 arms of lengths $3, 2, 1m$ joined by rotors and based on the origin is asked to point with the extreme of its smallest (and final) arm to the point $(0, 4)$, while keeping the x component of its center of mass as close as zero as possible. Which positions should the arms have? Can you get it to be actually at 0 ?

