

ECUACIONES NO LINEALES

F. VADILLO

RESUMEN. Estas notas explican algunos métodos numéricamente para resolver ecuaciones no lineales de la forma

$$f(x) = 0,$$

donde f es una función real de variable real. Dichas soluciones son los ceros de función f .

El orden de exposición es el siguiente: se comienza por el método más "ingenuo" que es el método de bisección para después comentar el método de la secante y las iteraciones de punto fijo describiendo el algoritmo zeroin. Finalmente se construye el método de Newton y su extensión al problema de los sistemas de ecuaciones no lineales.

ÍNDICE

1. Introducción	1
2. El método de bisección	2
3. El método de la secante	3
4. Iteraciones de punto fijo	4
5. El algoritmo zeroin	6
6. El método de Newton	7
7. El método de Newton para sistemas no lineales	8
Referencias	9

1. INTRODUCCIÓN

Se consideran ecuaciones no lineales de la forma

$$(1.1) \quad f(x) = 0,$$

donde f es una función real de variable real. Un caso sencillo es cuando la función f es un polinomio. Los polinomios de primero y segundo grado ya fueron tratados con éxito en tiempos muy remotos, los babilonios eran capaces de resolver ecuaciones cuadráticas. La historia del descubrimiento de la solución algebraica de la cúbica enfrentó a dos grandes rivales italianos: Cardano y Tartaglia hacia 1540, y Ferrari, alumno y secretario de Cardano resolvió en 1545 la ecuación de cuarto grado. Posteriormente fueron muchos los matemáticos eminentes que trataron de resolver las ecuaciones de grado superior a cuatro, aunque en vano puesto que el matemático

Received by the editors 17 de enero de 2006.

noruego Abel en 1893 probó que es imposible resolver por radicales la ecuación general de grado mayor que cuatro. En consecuencia, para calcular las raíces de polinomios de grado mayor que cuatro es imprescindible usar técnicas numéricas.

2. EL MÉTODO DE BISECCIÓN

El método de bisección es el más sencillo posible. Dada una función f real y continua en el intervalo $I_0 = [a, b]$ con valores de signos contrario en a y b , esto es $f(a)f(b) < 0$, el teorema de Bolzano asegura que f se anula en algún punto interior de I_0 .

El método de bisección calcula el punto medio del intervalo $c = \frac{a+b}{2}$ y $f(c)$. Si $f(c) = 0$, c es la raíz y el problema está resuelto. En el caso contrario, si $f(c) \neq 0$ tiene el mismo signo que $f(a)$ o $f(b)$, es decir, en uno de los dos intervalos $[a, c]$ o $[c, b]$ la función toma valores de signo contrario en los extremos y por lo tanto contiene una raíz. Llamando ahora I_1 a dicho intervalo estamos en la situación anterior pero con I_1 en lugar de I_0 , la ventaja es que la longitud del intervalo I_1 es la mitad de la I_0 . En el siguiente paso se repite el procedimiento, se evalúa f en el punto intermedio de I_1 y se queda con la mitad I_2 donde haya un cambio de signo en los valores de los extremos.

Continuando este proceso se construye una sucesión de intervalos encajados I_n de longitud $\frac{b-a}{2^n}$ que contienen una raíz de f .

Ejemplo 2.1. Para conocer en qué punto de la trayectoria se encuentra un satélite que gira en una órbita elíptica en torno a la Tierra en un instante t hay que resolver la ecuación de Kepler

$$(2.1) \quad x - e \sin x = z,$$

donde e es la excentricidad y z es un número que se calcula a partir de t .

En el caso $e = 0,5$ y $z = 0,7$, $f(0) = -0,7 < 0$ y $f(2) = 0,8 > 0$ por lo que el proceso iterativo se puede comenzar en el intervalo $[0, 2]$. Las cinco primeras bisecciones están recogidas en la siguiente tabla

n	Intervalo	Punto medio	Valor de f en el punto medio
0	[0,2]	1	-0.121
1	[1,2]	1.5	0.301
2	[1, 1.5]	1.25	0.075
3	[1,1.25]	1.125	-0.026
4	[1.125., 1.25]	1.1875	0.023
5	[1.125, 1.875]	1.1563	-0.001

El método de bisección es muy robusto porque se puede aplicar a cualquier función continua y no presenta posibilidades de error, aunque es un método lento: en cada paso dividimos el intervalo por la mitad. Además no aprovecha otras características de la función, por ejemplo no valora el tamaño en los extremos de los intervalos aunque parezca más plausible que la raíz esta más cerca al extremo donde se encuentre el menor valor absoluto.

3. EL MÉTODO DE LA SECANTE

En el método de la secante en lugar de obtener una sucesión de intervalos se calcula una sucesión de números que aproximan el cero de la función.

Comenzando de dos aproximaciones x_0, x_1 a la raíz que se busca α , primero se construye la recta secante que pasa por los puntos $(x_0, f(x_0)), (x_1, f(x_1))$ y se calcula su corte con el eje OX que se le llama x_2 , después se evalúa $f(x_2)$. El proceso se repite con x_1 y x_2 para calcular x_3 y sucesivamente x_4, x_5, \dots . La figura 1 ilustra el primer paso del método de la secante para el ejemplo 2.1.

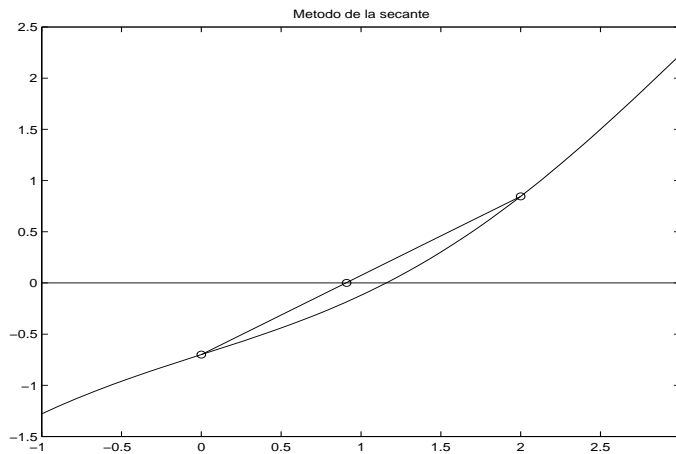


FIGURA 1. Método de la secante para el ejemplo 2.1.

La expresión que generaliza el método de la secante es la siguiente

$$(3.1) \quad x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, \dots$$

Las iteración del método de la secante (3.1) no son infalibles, puede ocurrir que algún de los x_n no se encuentre en el dominio de f o que el denominador sea casi cero, en cuyo caso la secante es horizontal y salvo que x_n fuese ya un cero de f no corta al eje de las abscisas.

La siguiente tabla recoge los resultados del método de la secante aplicado al ejemplo 2.1

n	x_n	$f(x_n)$
0	0	$-7,0 \times 10^{-1}$
1	2	$8,45 \times 10^{-1}$
2	0.9059	$-1,88 \times 10^{-1}$
3	1.1046	$-4,2 \times 10^{-2}$
4	1.1619	$3,2 \times 10^{-3}$
5	1.1579	$-4,91 \times 10^{-5}$

Comparado este método con el de bisección se ve claramente que ahora la convergencia es más rápida que. En cinco iteraciones se tienen residuos del orden de 10^{-5} .

Los posibles criterios de parada del algoritmo son tres:

$$(3.2) \quad |x_{n+1} - x_n| < Tol_1,$$

donde Tol_1 es una tolerancia elegida por el usuario, o bien

$$(3.3) \quad |f(x_n)| < Tol_2,$$

o se supera un número de iteraciones Max también elegida por el usuario

$$(3.4) \quad n > Max.$$

Una extensión obvia del método de la secante es **método de Muller**. Comenzando con tres valores $x_{-2}, x_{-1}, x_0 \in [a, b]$ se puede usar interpolación cuadrática, es decir, construiremos el polinomio de segundo grado que interpola por los puntos $(x_{-2}, f(x_{-2})), (x_{-1}, f(x_{-1}))$ y $(x_0, f(x_0))$ y hallar el punto de intersección con el eje de abscisas que sería x_1 . Repitiendo este proceso se obtiene una sucesión que en determinadas condiciones aproximan la raíz de la función.

4. ITERACIONES DE PUNTO FIJO

La iteración de punto fijo es la técnica más importante para calcular los ceros de funciones no lineales. Se comienza escribiendo la ecuación (1.1) de la forma

$$(4.1) \quad x = g(x),$$

tal que los ceros de la función f son exactamente los puntos fijos de la función g . Geométricamente resolver (4.1) significa hallar la intersección de la gráfica de g con la bisectriz de los cuadrantes primero y tercero.

Existen varias formas de reescribir (1.1) en la forma (4.1), por ejemplo se puede tomar $g(x) = x - f(x)$. La forma de construir la función g diferenciará a unos métodos iterativos de otros.

Después de elegir la llamada **función de iteración** g , el método iterativo construye la sucesión $\{x_n\}$ usando la recurrencia

$$(4.2) \quad x_{n+1} = g(x_n), \quad n = 0, 1, 2, \dots,$$

comenzando con un dato inicial x_0 . La figura 2 ilustra la geometría del método iterativo en el ejemplo 2.1 con $g(x) = 0,7 + 0,5 \sin x$ comenzando en $x_0 = 0$.

La cuestión que ahora se plantea es si es la sucesión $\{x_n\}$ converge a la raíz α de la función f , el siguiente teorema da una condición suficiente de convergencia.

Teorema 4.1. *Sea g una función con derivada continua en un intervalo de la forma $(\alpha - r_0, \alpha + r_0)$ tal que $r_0 > 0$ y $g(\alpha) = \alpha$. Entonces si*

$$(4.3) \quad |g'(\alpha)| < 1,$$

existe $r < r_0$ tal que para $x_0 \in [\alpha - r, \alpha + r]$ la sucesión $x_{n+1} = g(x_n), n = 0, 1, \dots$ está en $[\alpha - r, \alpha + r]$ y converge hacia α . Además, si los errores $e_n = x_n - \alpha \neq 0$ sus cocientes $\frac{e_{n+1}}{e_n} \rightarrow g'(\alpha)$ cuando $n \rightarrow \infty$.

La demostración de este teorema se puede consultar en la página 119 de la referencia [3].

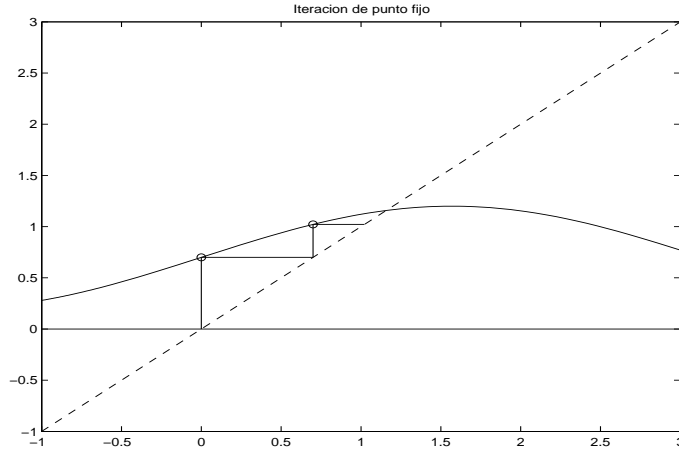


FIGURA 2. Iteración de punto fijo para el ejemplo 2.1.

Volviendo al ejemplo 2.1 con $g(x) = 0,7 + 0,5 \sin x$, el módulo de su derivada $|g'(x)| = |0,5 \cos x| < 1$ de donde se deduce la convergencia de la figura 2.

Es muy importante destacar que este resultado es un **teorema de convergencia local** porque la convergencia sólo se garantiza si el valor inicial x_0 está próximo a la raíz α que se busca.

Aunque la condición de convergencia (4.3) es sólo suficiente, es casi necesaria, de hecho se puede demostrar que en las hipótesis del teorema si

$$(4.4) \quad |g'(\alpha)| > 1,$$

las únicas sucesiones de la forma $x_{n+1} = g(x_n)$ que convergen a α son aquellas cuyos términos son α a partir de uno dado (teorema 9.4.3 de la página 129 de [3]).

Además de la convergencia de la sucesión a la raíz α es también importante la velocidad de convergencia para estimar el tiempo de computación necesario. Un método iterativo se dice que tiene un **orden de convergencia** $p \geq 1$ si:

$$(4.5) \quad |\alpha - x_{n+1}| \leq \lambda |\alpha - x_n|^p, \quad n \geq 0,$$

para algún $\lambda > 0$. Si $p = 1$ se dice que la convergencia es **lineal**, si $p = 2$ la convergencia es **cuadrática**, **cúbica** cuando $p = 3$ etc.

Ejemplo 4.2. El método de la secante tiene una convergencia superlineal ($p > 1$), exactamente el orden es la famosa razón áurea $\frac{1+\sqrt{5}}{2} \approx 1,62$ (Teorema 2.1 de la página 49 de [4]).

En la convergencia lineal con $\lambda = |g'(\alpha)| < 1$ para mejorar r decimales en la solución el número de iteraciones m necesarias sería

$$\lambda^m < 10^{-r} \Leftrightarrow \frac{r}{-\log_{10} \lambda} < m,$$

donde el factor $-\log_{10} \lambda$ se denomina **razón de convergencia del método**. De esta forma

1. Si $0 \ll \lambda < 1$ (λ próximo a uno), $\log_{10} \lambda \approx 0$ por lo que m es muy grande y la convergencia muy lenta.

2. Por el contrario si $0 < \lambda \ll 1$ (λ cercano a cero), $-\log_{10} \lambda$ es un valor grande y m pequeño, es decir, con pocas iteraciones se mejora mucho la aproximación.

5. EL ALGORITMO ZEROIN

La primera versión del algoritmo zeroin fue desarrollado por T.J.Dekker y sus colaboradores en 1969, aunque versiones posteriores lo han mejorado considerablemente. La idea de este algoritmo es la de combinar la versatilidad del método de bisección con la velocidad del método de la secante y el método IQI (inverse quadratic interpolation).

El método IQI en cada paso determina el polinomio cuadrático $P(y)$ con las condiciones:

$$(5.1) \quad x_n = P(f(x_n)), \quad x_{n-1} = P(f(x_{n-1})), \quad x_{n-2} = P(f(x_{n-2})),$$

que siempre corta al eje de abscisas en $y = 0$, así la siguiente iteración es:

$$(5.2) \quad x_{n+1} = P(0).$$

El método IQI puede tener serios problemas, por ejemplo necesita que $f(x_n)$, $f(x_{n-1})$ y $f(x_{n-2})$ sean diferentes lo que no está garantizado, pero sin embargo es muy rápido cerca de la raíz.

A grandes rasgos el algoritmo es como sigue

- Se comienza con a y b tales que $f(a)$ y $f(b)$ tienen signos opuestos.
- Se usa el método de la secante para calcular c .
- Se repiten los siguientes pasos hasta que $|b - a| < \epsilon|b|$ o $f(b) = 0$.
- Se ordenan a, b, c tales que
 - $f(a)$ y $f(b)$ tienen signos opuestos,
 - $|f(b)| \leq |f(a)|$,
 - c es el anterior a b .
- Si $c \neq a$ se da un paso con el algoritmo IQI.
- Si $c = a$ se da un paso con el método de la secante.
- Si el paso dado está en $[a, b]$ se toma.
- Si el paso no está en $[a, b]$ se usa bisección.

La implementación en MATLAB del algoritmo zeroin es el comando `fzero` que permite con gran eficacia aproximar raíces de ecuaciones no lineales. Para el ejemplo 2.1 el resultado de la instrucción `fzero('x-.5*sin(x)-.7',1)` es 1,1580 y $f(1,1580) = -1,1102 \times 10^{-16}$.

Moler en [1] da otra versión llamada `fzerotx` con dos argumentos, el primero especifica la función cuyo cero se quiere calcular y el segundo el intervalo $[a, b]$ de búsqueda, el resulta de `fzerotx('x-.5*sin(x)-.7', [1,1.2])` es otra vez 1,1580.

Finalmente el fichero `fzerogui.m` muestra gráficamente el comportamiento del algoritmo zeroin, si teclea la instrucción `fzerogui('x-.5*sin(x)-.7', [1,1.2])` se abrirá una ventana gráfica donde puede optar por correr el algoritmo eligiendo el botón `auto`, o bien puede ir eligiendo la opción del método: bisección está marcado en rojo, el punto secante en verde y el punto del algoritmo IQI en azul.

6. EL MÉTODO DE NEWTON

Si se quisiera sustituir la función f no lineal por una recta que pasa por el punto $(x_n, f(x_n))$ de ecuación

$$(6.1) \quad y - f(x_n) = m(x - x_n).$$

La intersección de la recta con el eje de abscisas tiene lugar para el valor de $x = x_n - \frac{1}{m}f(x_n)$ y el método iterativo que resulta es

$$(6.2) \quad x_{n+1} = x_n - \frac{1}{m}f(x_n).$$

El error es $e_n = x_n - \alpha$ que llevado a la ecuación anterior resulta

$$(6.3) \quad \alpha + e_{n+1} = \alpha + e_n - \frac{1}{m}f(\alpha + e_n).$$

Suponiendo regularidad de f en α se escribe

$$(6.4) \quad e_{n+1} = e_n - \frac{1}{m} \left(e_n f'(\alpha) + \frac{1}{2} e_n^2 f''(\alpha) + \dots \right),$$

o también

$$(6.5) \quad e_{n+1} = \left(1 - \frac{1}{m} f'(\alpha) \right) e_n - \frac{1}{2m} e_n^2 f''(\alpha) - \dots .$$

La elección óptima de la pendiente será entonces $m = f'(\alpha)$, sin embargo como α no es un valor conocido elegiremos $m = f'(x_n)$ de donde obtenemos el método de Newton-Raphson

$$(6.6) \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$

suponiendo siempre que la derivada f' no se anula en los puntos x_n . Este método simplemente sustituye la función por la recta tangente a la curva en el punto de abscisa x_n .

La iteración de Newton es una iteración de punto fijo para la función de iteración

$$(6.7) \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

Notese que efectivamente si $f(\alpha) = 0$ entonces $g(\alpha) = \alpha$. La convergencia depende de $|g'(\alpha)|$, calculando la derivada de g resulta que

$$(6.8) \quad g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2}.$$

Como $f(\alpha) = 0$ cuando $f'(\alpha) \neq 0$, $g'(\alpha) = 0$ y la convergencia es cuadrática. Más concretamente si f es de clase C^2 en el teorema 2.2 de la página 54 de [4] se demuestra que

$$(6.9) \quad \lim_{n \rightarrow \infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

El método de Newton tiene evidentes dificultades cuando la raíz α es múltiple porque entonces $f'(\alpha) = 0$. Si α es de multiplicidad p entonces $f(x) = (x - \alpha)^p F(x)$ donde $F(\alpha) \neq 0$ y la función de iteración de Newton tiene una derivada que en α vale $1 - 1/p$ por lo cual la convergencia del método de Newton ya no es cuadrática.

Incluso puede darse el caso perverso de que el método de Newton se eternice en una oscilación entorno a una raíz. Si se supone que

$$(6.10) \quad x_{n+1} - a = -(x_n - a) \quad \forall n,$$

entonces $f(x)$ debe satisfacer la ecuación diferencial de variables separadas

$$(6.11) \quad x - \frac{f(x)}{f'(x)} - a = -(x - a),$$

cuya solución exacta es

$$(6.12) \quad f(x) = \text{sign}(x - a)\sqrt{|x - a|}.$$

Evidentemente a es una raíz de f pero el método de Newton nunca llegará a alcanzarla porque calculado x_n

$$x_{n+1} = 2a - x_n \quad \Rightarrow \quad x_{n+2} = 2a - x_{n+1} = 2a - (2a - x_n) = x_n.$$

7. EL MÉTODO DE NEWTON PARA SISTEMAS NO LINEALES

Para construir el Método de Newton para sistemas de ecuaciones lineales, se considera el caso de dos ecuaciones de la forma

$$(7.1) \quad f_1(x_1, x_2) = 0,$$

$$(7.2) \quad f_2(x_1, x_2) = 0.$$

En la iteración k -ésima se conoce $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)})$ y tomando el desarrollo de Taylor hasta el primer orden se tiene

$$f_1(\mathbf{x}) \cong f_1(\mathbf{x}^{(k)}) + \frac{\partial f_1}{\partial x_1}(\mathbf{x}^{(k)})(x_1 - x_1^{(k)}) + f_1(\mathbf{x}^{(k)}) + \frac{\partial f_1}{\partial x_2}(\mathbf{x}^{(k)})(x_2 - x_2^{(k)}) = 0,$$

$$f_2(\mathbf{x}) \cong f_2(\mathbf{x}^{(k)}) + \frac{\partial f_2}{\partial x_1}(\mathbf{x}^{(k)})(x_1 - x_1^{(k)}) + f_2(\mathbf{x}^{(k)}) + \frac{\partial f_2}{\partial x_2}(\mathbf{x}^{(k)})(x_2 - x_2^{(k)}) = 0,$$

que de forma matricial se escribe

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}(\mathbf{x}) \cong \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}(\mathbf{x}^{(k)}) + \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \end{bmatrix}(\mathbf{x}^{(k)}) \begin{bmatrix} x_1 - x_1^{(k)} \\ x_2 - x_2^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Resolviendo ahora la ecuaciones en x_1 y x_2 la iteración que resulta es

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix} - \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \end{bmatrix}^{-1}(\mathbf{x}^{(k)}) \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}(\mathbf{x}^{(k)}),$$

que habitualmente se escribe en la forma

$$(7.3) \quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}^{(k)})^{-1}\mathbf{f}(\mathbf{x}^{(k)}),$$

con el Jacobiano $\mathbf{J}^{(k)}(m, n) = [\partial f_m/\partial x_n](\mathbf{x}^{(k)})$.

En el programa newtons.m de la referencia [8] se puede observar una implementación de este método de Newton para sistema de ecuaciones no lineales.

La extensión del método de Newton para sistemas de n ecuaciones no lineales es trivial y se puede consultar en [4], [2], [6], [5] y [7]. Lo que resulta más complicado son los resultados de convergencia. Los teoremas de convergencia son locales por lo cual es necesario comenzar las iteraciones suficientemente cerca de la solución que se busca, y la convergencia, en general, es cuadrática.

REFERENCIAS

1. C.B.Moler, *Numerical computing with MATLAB*, SIAM, 2004.
2. H.Keller E.Isaacson, *Analysis of numerical methods*, Wiley, 1966.
3. J.M.Sanz-Serna, *Diez lecciones de cálculo numérico*, Universidad de Valladolid, 1998.
4. K.E.Atkinson, *An introduction to numerical analysis*, Wiley, 1978.
5. C.F.Van Loan, *Introduction to scientific computing*, Prentice-Hall, 2000.
6. R.Bulirsch and J.Stoer, *Introduction to numerical analysis*, Springer, 1980.
7. A.Quarteroni R.Sacco, F.Saleri, *Numerical mathematics*, Springer, 2000.
8. T.S.Chung W.Y.Yang, W.Cao and J.Morris, *Applied numerical methods using MATLAB*, Wiley Interscience, 2005.

DEP. MATEMÁTICA APLICADA Y ESTADÍSTICA DE LA UNIVERSIDAD DEL PAIS VASCO
E-mail address: `fernando.vadillo@ehu.es`