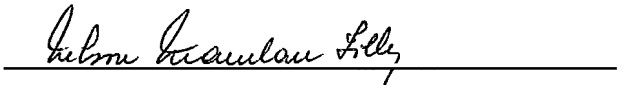
TÓPICOS EM RESOLUÇÃO NUMÉRICA DE SISTEMAS NÃO LINEARES

José Mario Martinez Perez

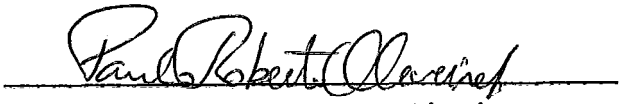
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS (D.Sc.)

Aprovada por:

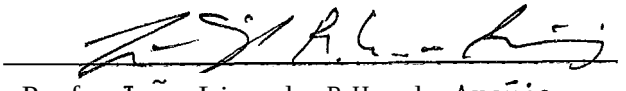
Prof. Hugo Daniel Scolnik
(presidente)



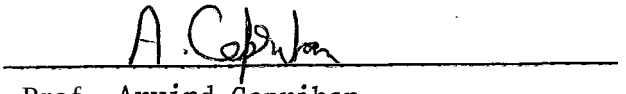
Prof. Nelson Maculan Filho



Prof. Paulo Roberto de Oliveira



Prof. João Lizardo R.H. de Araújo



Prof. Arvind Caprihan

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 1978

MARTÍNEZ PÉREZ, JOSÉ MARIO

Tópicos en Resolução Numérica de Sistemas Não Lineares [Río de Janeiro] 1978.

V, 73p. 29,7 cm (COPPE-UFRJ, D.Sc, Engenharia de Sistemas, 1978)

Tese - Univ. Fed. Río de Janeiro. Fac. Engenharia

1.Resolução Numérica de Sistemas Não Lineares I.COPPE/UFRJ II.Título(série).

Agradecimientos

A Hugo Scolnik, que me inició en el área de la optimización, y me infundió de la audacia necesaria para abordar problemas.
A mis compañeros de trabajo, y a menudo interlocutores, Carlos, Ana, Irene, Alfonso, Betty y Jorge PI.
A David Gay, del National Bureau of Economic Research, por su cuidadosa lectura de diversas partes de esta tesis, muchas correcciones y útiles sugestiones.
A Fernando Dasombrío y el Grupo Florencio Parravicini, con quienes conjugamos Análisis Numérico e ideales diversos.
A mis profesores de la COPPE, al Centro Brasileiro de Pesquisas Físicas y a la Sociedade Brasileira de Instruçao, en la persona del Profesor Cándido Mendes, por la asistencia prestada en diversos aspectos que hicieron a la elaboración de esta tesis.
A mis alumnos de la Universidad de Buenos Aires.

RESUMEN

Esta tesis contiene tres aportes al área de Resolución numérica de sistemas no lineales sin derivadas. En primer lugar, trata de un reciente método da tipo Quasi-Newton, el método de Broyden con "updates" proyectados (o de Broyden-Gay-Schnabel). Se prueba que, bajo cierta condición más débil que la de independencia lineal uniforme , dicho método tiene R-orden al menos la ra:~ positiva de $t^{2n} - t^{2n-I} - 1 = 0$. En segundo lugar, se generalizan los métodos de Brent y Brown para resolver sistemas. Los métodos generalizados explotnn los casos en que una parte sustancial del trabajo necesario para evaluar una función es común a la evaluación de otras. Se dan resultados de convergencia local y algunos experimentos numéricos. Finalmente, sc presentan algunas implementaciones estables del método secante secuencial. Se evalúan las utilidades y costos relativoa de los métodos presentados en base al análisis de resultados teóricos, posibilidades de extensión, costos computacionales y aprovechamiento de la información en casos particulares. Se hacèn comparaciones con el método de Broyden con "updates" proyectados. Se demuestran teoremas de convergencia.

RESUMO

Esta tese contém três contribuicões ã área de Resolução numérica de sistemas não lineares sem derivadas. Em primeiro lugar, trata-se de um método recente do tipo Quasi-Newton, o método de Broyden com "updates" projetados (Broyden-Gay-Schnabel). Prova-se que sob certa condição mais fraca que a da independencia linear uniformo, o R-ordem do mencionado método é ao menos a raíz positiva de $t^{2n} - t^{2n-1} - 1 = 0$. Em segundo lugar, os métodos dc Brent e Brown para resolver sistemas são generalizados, Os novos métodos exploram os casos em que uma parte substancial do trabalho necessário para avaliar mia função é comum a avaliação de outras, Apresentam-se resultados de convergencia local e algumas experiencias numéricas. Finalmente, apresentam-se algumas implementações estáveis do método secante sequencial. Avaliam-se as utilidades e custos dos métodos apresentados com base na análise de resultados teóricos, possibilidades de extensão, custos computacionais, e aproveitamento do informação em casos particulares. Fazem-se comparações com o método de Broyden com "updates" projetados, Demonstram-se teoremas dc convergencia.

ABSTRACT

In this thesis, three contributions to the area of Numerical Resolution of Nonlinear Systems without derivatives are presented. First, it deals with *a* recent Quasi-Newton method: Broyden's method with projected updates (Broyden-Gay-Schnabel). It is proved that, under certain condition, which is weaker than the uniformly linear independence, the R-order of that method is at least the positive root of $t^{2n} - t^{2n-1} - 1 = 0$. Next, Brent and Brown's methods for solving systems are generalized. The generalized methods exploit the cases in which a substantial part of the work wasted to evaluate c function ia common to the evaluation of other functions. Local convergence results are given and some numerical experiences. Finally, some atable implementations of the sequential secant method are presented. The advantages and relative costs of these methods are evaluated, based on the theoretical results, extension possibilities, computational costs and exploitation of the information in particular cases. Comparisons with Broyden's method with projected updatoa are made and convergenco theorems are proved.

I N D I C E

I.- INTRODUCCION

El tema de esta tesis es: algoritmos para la resolución numérica de sistemas algebraicos no lineales que no hacen uso de derivadas analíticas.

Nos ocuparemos de tres tipos de métodos, los tres susceptibles de implementaciones prácticas eficientes: los Quasi-Newton, los secantes y los derivados de la aproximación de Brown. Los tres tipos se basan en la aproximación lineal del sistema, lo cual es, seguramente, lo que los hace implementables y prácticos en la mayoría de los problemas corrientes. Métodos basados en aproximaciones de más alto orden han sido considerados en el pasado reciente y remoto, pero no parecen haber dado origen a rutinas con espectro razonablemente amplio de aplicación. Desde el punto de vista de su complejidad, tanto los métodos Quasi-Newton como los secantes que serán considerados aquí utilizan una evaluación completa de todo el sistema por iteración y una cantidad de operaciones dada por un polinomio de grado 2 en n; en tanto que los métodos de tipo Brown evalúan más veees el sistema (aunque no la misma cantidad de veces todas sus componentes) y usan una cantidad de operaciones por iteración del orden de la dimensión al cubo. El mayor trabajo se compensa con un orden más alto de convergencia. De todos modos, cuando el proceso es "controlado", es decir, cuando cada cierto número de pasos se verifica si el método está realizando algún progreso, y se modifica la predicción dada por el mismo en caso contrario (usualmente con técnicas de "damping"), el trabajo por iteración puede variar en forma casi impredecible. Sin embargo, puede postularse que, independientemente de la eficiencia de los controles, calidad de las búsquedas unidimensionales, etc., la versión "no controlada" de un algoritmo eficiente debe tener un comportamiento razonable.

Los métodos Quasi-Newton han sido los más estudiados. La gran variedad de los mismos, riqueza de resultados, y abundancia de experiencia al respecto se dehen a un hecho simple: todos ellos se derivan de una única ecuación matricial, con infinitas soluciones, siendo que cada solución da origen, por definición, a un método. No obstante esta variedad intrínseca, la supervivencia del método de Croyden clásico, también llamado "método de Broyden bueno", es

llamativa. Pocas cosas en los últimos anos, parecieron justificar el empeño en la búsqueda de fórmulas más eficientes; lo que no sucedió , por cierto en el área vecina de minimización de funciones, en donde el requisito adicional de mantener la simetria y, a veces, la definición positiva de las matrices, llevó a numerosas, polémicas, e ingeniosas fórmulas rivales. Es notable que la innovación en el área tenqa como origen la meditación acerca de una de laç propiedades del método de"Broyden bueno", la de producir la matriz que menos varía, en el sentido de la norma de Frobenius, respecto de la aproximación anterior del jacobiano. Spedicato y Greenstadt han trabajado en los Últimos tiempos en fórmulas "variacionalmente derivadas" que, esencialmente, explotan la idea de mínima variación, utilizando normas que obedecen a criterios adicionales. El método considerado en I-a primer parte de esta tesis, de Gay y Schnabel, tambien fue pensado con el mismo tipo de ideas, aunque un **enfoque diferente** lo coloca, si se quiere, "del lado secante".

Prestigiosos y elegantes, no es casual que los métodos Quasi-Newton hayan dado origen a rutinas harto difundidas y eficientes. Para resolver sistemas no lineales; NS02A, una subrutina de Harwell, que implementa la versión de Powell híbrida del método de Broyden **bueno, es una rutina rica en controles y salvaguardas Útiles que** debe ser referencia obligada en la elaboración de programas orientados al usuario. **Más aún, los grados de libertad permitidos por** la ecuación matricial fundamental, admiten la satisfacción de propiedades deseables cuando el caso es la minimización de funciones, propiedades que métodos más restrictivos no pueden cumplir.

Los métodos secantes (nos referiremos aquí con ese nombre al llamado método secante de n + 1 puntos en el libro de Ortega y Rheinboldt, método secante secuencial de Gragg y Stewart, método secante de Wolfe, etc.) han sido mucho menos estudiados en los últimos años. Tanto es así, que el libro de Ortega y Rheinboldt, de 1970, da casi toda la información publicada acerca de los mismos, lo que , por cierto no sucede en tratándose de los métodos Quasi-Newton. Lo tradicional ha sido atribuír al método dos pecados formidables: la eventual degeneración de los incrementos en la variable independiente, y la inestabilidad de la fórmula de generación de la matriz del sistema. Válidas las críticas, es injusto sin embargo que se haya dedicado tan poco esfuerzo a aliviar tales culpas. Sobre todo porque el método tiene una clara y distinta justificación geométrica y una simplicidad de formulación que lo hacen digno de entrar en la arena de

los algoritmos competitivos para el problema de sistemas no lineales. Aparentemente Cragg y Stewart lo entendieron así, y publicaron en 1976, una implementación interesante del método. Lamentablemente, según el notable estudio de Bus (A comparative study of programs for solving nonlinear equations, TR Dept. of Numer. Anal., NW 25/76, Stichting Math. Centr., Amsterdam), y también según la experiencia del autor de esta tesis, el programa publicado por ellos en 1975 contiene un error que lo hace incomparable con otros algoritmos.

El método secante puede ser visto como una de las formas de resolver la ecuación Quasi-Newton, por eso en apariencia poco se puede decir del mismo. Esa coercividad le impide satisfacer en principio las folklóricas virtudes de los algoritmos dedicados a la minimización de funciones y, finalmente, las salvaguardas a la degeneración lo hacen "poco elegante". Sin embargo, en principio, nada de ello le impide resolver sistemas no lineales.

El método de Brown nació como un método con propiedades de convergencia similares a las del método de Newton (discreto), que realiza menos trabajo por iteración que aquél. La evaluación diferenciada de distintas componentes de la función lo colocan, en cierto sentido, fuera de competencia con los anteriores. Por esa circunstancia, relativa a la estructura del método, existen funciones para las cuales debe funcionar, en principio, mejor que los otros, y a la inversa. Originado en 1969, el método fue modificado en 1975 por Gay, que logró reducir drásticamente el trabajo computacional por iteración de su antecesor. Un método similar, en términos de estructura computacional y propiedades de convergencia (aunque no en estabilidad y número de operaciones por iteración, fue introducido por Brent, en 1973 La programación de instrumentos de control para estos métodos es un tanto engorrosa, y, aparentemente, no puede hacerse sin destruír, en cierto modo, algunas de sus características esenciales. Por lo tanto su apariencia es la de no ser "globalmente convergentes", y ello es lo que hizo que se los considerara poco en el primer quinquenio de los años 70. Sorprendió entonces una evaluación comparativa de Michel Cosnard (Cornell, 1975), que, en cierto número de funciones test mostró que una implementación suya del método de Brent y otra de el de Brown-Gay, sin "damping", funcionaban mejor que NS02A, en términos de tiempo de computación, y a veces hasta de robustez. Dichas experiencias motivaron a J.J. Moré y al propio Cosnard a elaborar una

rutina eficiente implementando el método de Brent para IMSL, cuyos fundamentos incluyen una variedad de tests de convergencia y estabilidad, pero no 'damping".

La evaluación de las teorias de convergencia existentes, tanto en sistemas no lineales como en minimización de funciones, ha de hacerse, lamentablemente,en términos no rigurosos. Si aceptamos que una teoría debe explicar hechos, la polémica aparece no sólo en la pregunta de si determinada teoria explica los hechos, sino de la definición de "qué *es* un hecho" en esta área del conocimients. A menudo se encuentra en la literatura la afirmación acerca de "la mayoria de las aplicaciones" o "la mayoría de los casos" (y çin duda el lector las encontrará también en esta tesis). Cuál es el significado de esos clichés es tema de teoría del conocimiento y escapa a los objetivos de esta introducción, pese a lo cual es recomendable mirar con crítica desconfianza las sentencias sobre tan dudosos sujetos. El objetivo de este párrafo es otro: existe abundante teoria de convergencia "global" cuyo denominador común es dar resultados del tipo: "Todo punto de acumulación de la sucesión es una solución (i.e. las teorias de Zangwill, Polak, Elkin, y otros). Es más, resultados de convergencia de ese tipo son fáciles de obtener. A menudo modificaciones insignificantes de algoritmos "malos" se convierten así en "globalmente convergentes". En mi opinión, como tales, esos resultados no explican nada. Sin embargo, las hipótesis que permiten obtenerlos son a menudo salvaguardas prácticas que es razonable introducir en subrutinas que implementen los algoritmos. Tales, por ejemplo, los casos de las hipótesis de tino "gradient related" en la teoría de Elkin. Pero lo paradójico es que los resultados , estrictamente locales, de orden de convergencia, si parecen mucho más explicativos del comportamiento global de algoritmos. Esto se debe, no *a* que "el entorno al cual se refiere el teorema sea en realidad más grande de lo que parece", sino a que las propiedades de que hacen uso los teoremas de orden son propiedades importantes aún lejos de la solución, si bien sólo sabemos cuantificar su importancia (orden) en un entorno de la misma. La relación, por lo tanto, es más bien de analogía, y la elaboración de una teoria global rigurosa y explicativa es un campo aparentemente virgen.

Los métodos Quasi-Newton tienen orden de convergencia superli-

neal, aún con la introducción de técnicas de "damping" sofisticadas como las de NS02A. El resultado de Gay (1377) según el cual el método de Broyden bueno , sin relajación, tiene convergencia 2n-O- cuadrática, sugiere que aquel resultado está lejos de no poderse superar, al menos en lo que respecta al R-orden. Salvada la degeneración de los incrementos, el orden del método secante eç la raíz positiva de $t^{n+1} - t^n - 1 = 0$: y los resultados de orden para los métodos de tiro Brown son los mismos que para el método de Newton.

En cuanto a métodos de "damping", como ya hemos dicho, lo más elaborado es la relajación curvilínea de Powell, implementada en NS02A. Los métodos de tipo Brown precisan de técnicas ad hoc al respecto, algunas de las cuales fueron estudiadas por Gay.

Los métodos de Quasi-Newton, el método de Brown-Gay y algunas versiones del método secante se pueden implementar de manera de aprovechar determinadas Situaciones de esparcimiento del jacobiano. Los métodos de tipo Brown pueden aprovechar la disponibilidad de algunas derivadas analíticas de las funciones.

De manera general, todo método para resolver sistemas no lineales es un método de minimización de funciones, en donde, simplemente, cambia la función objetivo. El problema es el uso y aprovechamiento de la información. La simetría y positividad del hessiano de una función convexa posibilita en el segundo caso, el uso de la barata y estable factorización de Chòlesky en tanto que en sistemas no 71- neales buscamos la estabilidad a través de factorizaciones ortogonales. Sin embargo, tres hechos deben ser destacados: a) Las factorizaciones ortogonales son más estables que la de Cholesky, b) La rutina MINFA, de M.J.D. Powell, que utilizaba la fórmula simétrica de Broyden-Powell para minimización de funciones, no generaba direcciones de descenso y sin embargo fue desde 1970 hasta 1975 considerada la mejor rutina para minimizaciòn sin restricciones disponible; c) Ideas próximas a I?- secante han si?.trabajadas con éxito en el método de Davidon de 1975.

Los capítulos que componen el cuerpo de esta tesis son los tres siguientes, y pueden ser leídos independientemente. El segundo, "Sobre el orden de convergencia del método de Broyden-Gay-Schnabel" se refiere a uno de los más recientes métodos Quasi-Newton. El resultado fundamental es que su R-orden de convergencia es por lo menos la raíz positiva de $t^{2n} - t^{2n-1} - 1 = 0$. El tercero, "Métodos de Brent y Brown generalizados", extiende los métodos de tipo Brown a una estructuraciónpor bloques de manera de aprovechar lo mejor posible e1 trabajo

utilizado en la evaluación de componentes del sistema. En el cuarto se dan y discuten algunos algoritmos basados en el método secante secuencial.

II.- SOBRE EL ORDEN DE CONVERGENCIA DEL METODO DE BROYDEN-GAY- SCHNABEL

1.- Preliminares

El método de Broyden goza desde hace varios años de la preferencia de muchos autores para resolver sistemas de ecuaciones no lineales sin derivadas. (Ver [?]). Diversas modificaciones del mismo resultaron metodos robustos, rápidos y eficientes (ver [?] y [9]).

Desde tiempo atrás (ver [3]) se conocía la converqencia superlineal del método pero solo en 1977, Gay probó que tiene terminación finita en 2n pasos para sistemas lineales, converqencia 2n-O-cuadrática y R-orden de converqencia igual a $2^{1/2n}$. (Ver [4]).

También en 1977 Gay y Schnabel ([5]) propusieron una modificación del updating tradicional del método, probaron terminación en n + 1 pasos para sistemas lineales y convergencia superlineal siguiendo las líneas de L33 .

El nuevo updating utiliza la proyección del último incremento sobre el subespacio ortogonal al generado por algunos de los anteriores. Por lo tanto está muy relacionado con el método secante secuencial (ver [1], [6] , [8] , y [10]) y puede ser considerado como una modificación de éste.

Gay conjetura en [4] que el método tiene convergencia (n + 1) - cuadrática y R-orden de convergencia igual a $2^{1/(n+1)}$. En las secciones que siguen probamos que el R-orden es al menos la raíz positiva de $t^{2n} - t^{2n-1} - 1 = 0$, número mayor que $2^{1/(n+1)}$, para lo cual explotamos la relación con el método secante secuencial y utilizamos una hipótesis más debil que la generalmente usada para probar la convergencia de éste.

2.- Resultados básicos

Indicaremos con $\|.\|$ siempre la norma 2.

Supongamos

$$F: A \subset R^n \longrightarrow R^n , \qquad (2.1.a)$$

A abierto y convexo? $F \in C^1(A)$, (2.1.b)

$$F = (f_1,\ldots,f_n)^t , \qquad (2.1.c)$$

$$x^* \in A \text{ tal que } F(x^*) = 0, \qquad (2.2)$$

$$J(x) = (\partial f_i/\partial x_j)(x) \qquad (2.3)$$

$$J(x^*) \text{ no singular} \tag{2.4}$$

y para todo $x, y \in A$,

$$\|J(x) - J(y)\| \leq M \|x - y\| \quad \text{con } M > 0. \tag{2.5}$$

Supongamos que una sucesión en R^n es generada por las fórmulas:

$$x^o \in A, \; B_o \in R^{nxn} \tag{2.6.a}$$

$$x^{k+1} = x^k - B_k^{-1} F(x^k), \tag{2.6.b}$$

con

$$B_{k+1} = B_k + \Delta B_k \tag{2.7}$$

y

$$\Delta B_k = (\Delta F_k - B_k \Delta x_k) z_k^t / z_k^t \Delta x_k, \tag{2.8}$$

donde

$$\Delta x_k = x^{k+1} - x^k, \tag{2.9}$$

$$\Delta F_k = F(x^{k+1}) - F(x^k), \tag{2.10}$$

y

$$|z_k^t \Delta x_k| / \|z_k\| \|\Delta x_k\| \geq \delta > 0 \tag{2.11}$$

para todo $k = 0,1,2,...$

Para evitar hipótesis engorrosas supondremos además que

$F(x^k) \neq 0$ para todo $k = 0,1,2,...$ y B_k invertible , (2.12)

de manera que las fórmulas anteriores están siempre hien definidas. (2.6) - (2.11) describen un "método de Broyden con updating variable". El caso $z_k = \Delta x_k$ da el método de Broyden original.

Lema 2.1

Bajo las hipótesis (2.1)(2.3) y (2.5), se tiene que para todo $x, y \in A$

$$\|F(x) - F(y) - J(x)(y - x)\| \leq (M/2) \|y - x\|^2 \tag{2.13}$$

Demostración.

Ver [8] .

Lema 2.2.

Bajo las hipótesis (2.1), (2.3), (2.5), (2.6) - (2.12), si x^k y x^{k+1} pertenecen a A, entonces,

$$\| B_{k+1} - J(x^{k+1}) \| \leq (1 + 1/\delta)\| B_k - J(x^k)\| + M(1 + 1/(2\delta))\|\Delta x_k\| \tag{2.14}$$

Demostración.

$$\| B_{k+1} - J(x^{k+1})\| \leq \|B_{k+1} - B_k\| + \|B_k - J(x^k)\| + \|J(x^k) - J(x^{k+1})\| \tag{2.15}$$

Además, por (2.13),

$$\| \Delta F_k - B_k\Delta x_k\| \leq \| \Delta F_k - J(x^k)\Delta x_k\| + \| J(x^k)\Delta x_k - B_k\Delta x_k\| \leq$$
$$\leq (M/2)\| \Delta x_k\|^2 + \| J(x^k) - B_k\| \|\Delta x_k\|.$$

Por lo tanto, por (2.7) -(2.11),

$$\|B_{k+1} - B_k\| = \| \Delta B_k\| \leq \| \Delta F_k - B_k\Delta x_k\| \|z_k\| / | z_k^t \Delta x_k| \leq$$
$$\leq (M\|\Delta x_k\|^2/2 + \| J(x^k) - B_k\| \|\Delta x_k\|) / (\delta \|\Delta x_k\|).$$

Luego,

$$\| B_{k+1} - B_k\| \leq (M/2\delta)\|\Delta x_k\| + \| J(x^k) - B_k\|/\delta . \tag{2.16}$$

Pero por (2.5),

$$\|J(x^{k+1}) - J(x^k)\| \leq M \|\Delta x_k\|, \tag{2.17}$$

luego la tesis se obtiene remplazando (2.16) y (2.17) en (2.15).

Los siguientes lemas muestran la relación entre el clásico criterio de indegendencia lineal uniforme y los tests usados para detectar singularidad en los prscesos de ortogonalización.

Lema 2.3.

Sea $A = (v_1,\ldots,v_n) \in R^{nxn}$, $S_i = [v_1,\ldots,v_i]$ el subespacio generado por $v_1,\ldots,v_i$, $i = 1,\ldots,n$; a_i el ángulo entre S_i y v_{i+1}, $i = 1,\ldots,n-1$. Entonces,

$$|\det A| = |\text{sen } \alpha_1| \ldots |\text{sen } \alpha_{n-1}| \, \|v_1\| \ldots \|v_n\| . \quad (2.18)$$

Demostración.

Si A es singular el resultado es trivial. De lo contrario sean Q y R talos que Q ortogonal ($QQ^t = Q^tQ = I$), R triangular superior, y $A = QR$. Indiquemos $R = (r_1,\ldots,r_n) = (r_{ij})$, $i = 1,\ldots,n$, $j = 1,\ldots,n$, $r_i \in R^n$ para todo $i = I,\ldots,n$. Luego $|\det A| = |\det R|$ y $\|v_j\|^2 = \|r_j\|^2 = r_{1j}^2+\ldots+r_{jj}^2$, $j = 1,\ldots,n$. Además,

$$S_i = [Qr_1,\ldots,Qr_i] \text{ y}$$

$$v_{i+1} = Qr_{i+1},$$

luego, si β_i es el ángulo entre $[r_1,\ldots,r_i]$ y r_{i+1}, por la ortogonalidad de Q , resulta $|\alpha_i| = |\beta_i|$. Pero , por ser R triangular superior,

$$[r_1,\ldots,r_i] = e \ ,\ldots,e \quad \text{(los i primeros vectores}$$

de la base canónica de R^n; por lo tanto,

$$|\text{sen } \alpha_i| = |\text{sen } \beta_1| = |r_{i+1,i+1}|/\|r_{i+1}\|$$

para todo $i = 1,\ldots,n-1$; luego

$$|\text{sen } \alpha_1| \ldots |\text{sen } \alpha_{n-1}| = |r_{22}\ldots r_{nn}|/\|r_2\|\ldots\|r_n\| =$$

$$*|r_{11}\ldots r_{nn}|/\|r_1\|\ldots\|r_n\| = |\det R|/\|r_1\|\ldots\|r_n\| =$$

$$= |\det A| / \|v_1\|\ldots\|v_n\|, \text{ lo que prueba}$$

la tesis.

Lema 2.4.

Sea $\{A_\lambda\}_{\lambda \in L}$ una familia de matrices no singulares de n x n, $A_\lambda = (v_1(\lambda),\ldots,v_n(\lambda))$, y denotemos por $\alpha_i(\lambda)$ al ángulo entre v A, . . v A y $v_{i+1}(\lambda)$, $i = 1,\ldots,n-1$, $A \in L$; entonces,

a) Si $|\det A_\lambda| / (\|v_1(\lambda)\| \ldots \|v_n(\lambda)\|) \geq \varepsilon > 0$ para todo $\lambda \in L$, entonces $|\operatorname{sen} a_i(\lambda)| \geq \varepsilon$ para todo $i = 1,\ldots,n-1$, $\lambda \in L$.

b) Si $|\operatorname{sen} \alpha_i(\lambda)| \geq \delta > 0$ para todo $i = 1,\ldots,n-1$, $h \in L$, entonces $|\det A_\lambda| / (\|v_1(\lambda)\| \ldots \|v_n(\lambda)\|) \geq \delta^{n-1}$ para todo $\lambda \in L$.

Demostración.

Trivial a partir del lema 2.3.

El siguiente es una generalización del Teorema 11.3.3 de [8], cuya prueba omitiremos por ser análoga a la de aquél.

Lema 2.5.

Supongamos 2.1, (2.3) y (2.5), y además, $x_1,\ldots,x_n \in A$, $p_1,\ldots,p_n \in R^n$, $x \in A$, $x_i' = x_i + p_i \in A$ para todo $i = 1,\ldots,n$, $q_i = F(x_i') - F(x_i)$, con

$$|\det (p_1,\ldots,p_n)| / (\|p_1\| \ldots \|p_n\|) \geq \varepsilon > 0,$$

$$B = (q_1,\ldots,q_n)(p_1,\ldots,p_n)^{-1}.$$

Entonces, existe $K > 0$, que sólo depende de n y ε, tal que

$$\|J(x) - B\| \leq K \max\{\|p_j\|/2 + \|x_j - x\|\}_{j=1}^{n}.$$

3.- El updating de Gay - Schnabel.

A continuación describimos la elección de Gay - Schnabel del vector z_k en (2.8).

Supongamos 2.1), (2.3), (2.6) - (2.10), y (2.12), y además, sea R_k, $k = 1,2,\ldots$ una sucesión de enteros positivos y $\overset{h}{z}_k$, $k = 0,1,2,\ldots$ una sucesión de vectores de R^n generados como sigue:

$$\text{a) } \hat{z}_o = 0. \qquad (2.18.a)$$

b) Si $|\hat{z}_k^t \Delta x_k| > 6 \|\hat{z}_k\| \|\Delta x_k\|$, entonces

$z_k = \hat{z}_k$ y $\ell_{k+1} = \ell_k + 1$.

Si $|\hat{z}_k^t \Delta x_k| \leq 6 \|\hat{z}_k\| \|\Delta x_k\|$, entonces

$z_k = \Delta x_k$ y $\ell_{k+1} = 1$. (2.18.h)

c) Para $k = 1,2,\ldots$, $\hat{z}_k$ es la proyección ortogonal de Δx_k sobre $[\Delta x_{k-1},\ldots,\Delta x_{k-\ell_k}]^{\perp}$. (2.18.c)

El test $|\hat{z}_k^t A x_k| \geq 6 \|\hat{z}_k\| \|\Delta x_k\|$ sirve para detectar si el nuevo incremento es dependiente de los ℓ_k anteriores, con la tolerancia 6. El cocientc $|\hat{z}_k^t \Delta x_k| / \|\hat{z}_k\| \|\Delta x_k\|$ es el módulo del seno del ángulo entre Δx_k y $[\Delta x_{k-1},\ldots,\Delta x_{k-\ell_k}]$. Naturalmente, cuando ℓ_k llega a n, ese seno es 0, Δx_k es declarado dependiente y $z_k = \Delta x_k$, retomándose la fórmula clásica de Broyden. Esto es lo que diferencia al método de Gay-Schnabel del método secante secuencial.

Proposición 3.1.

Con la definición 2.18 y las hipótesis que la posibilitan, se tiene $B_{k+1} \Delta x_{k-j} = \Delta F_{k-j}$ para todo $j = 0,1,\ldots,\ell_{k+1}-1$.

Demostración.

Ver [5].

Proposición 3.2.

Supongamos (2.18) con sus hipótesis previas. Si $x^k,\ldots,x^{k+n} \in A$ y $|\hat{z}_j^t \Delta x_j| \geq 6 \|\hat{z}_j\| \|\Delta x_j\|$ para todo $j = k+1,\ldots,k+n-1$, (2.19) entonces, existe $K > 0$ independiente de k , tal que

$$\|B_{k+n} - J(x^{k+n})\| \leq K (\|\Delta x_k\| + \ldots + \|\Delta x_{k+n-1}\|).$$

Demostración.

Por la proposición 3.1,

$$B_{k+n}\Delta x_k = \Delta F_k$$

$$\vdots$$

$$B_{k+n}\Delta x_{k+n-1} - \Delta F_{k+n-1} .$$

Por el lema (2.1) y (2.19). $(\Delta x_k,\ldots,\Delta x_{k+n-1})$ es no singular,

$$B_{k+n} = (\Delta F_k,\ldots,\Delta F_{k+n-1})(\Delta x_k,\ldots,\Delta x_{k+n-1})^{-1},$$

y

$$|\det(\Delta x_k,\ldots,Ax_{k+n-1})| A_k \|\ldots\|\Delta x_{k+n-1}\| \geq \delta^{n-1},$$

por lo tanto , la tesis se sigue del lema 2.5.

Proposición 3.3.

Supongamos (2.1),(2.3),(2.5)-(2.12) y (2.18) r (2.19) y además $x^{k+n+1},\ldots,x^{k+n+s} \in A$. Entonces , existe $K_s > 0$, independiente de k tal que

$$\|B_{k+n+s} - J(x^{k+n+s})\| \leq K_s(\|\Delta x_k\|+\ldots+\|\Delta x_{k+n+s-1}\|).$$

Demostración.

Para s = 0 es la proposición 3.2. Supongamos la tesis cierta para s - 1. Entonces por el lema 2.2,

$$\|B_{k+n+s} - J(x^{k+n+s})\| \leq (1 + 1/\delta)\|B_{k+n+s-1} - J(x^{k+n+s-1})\| +$$

$$+ M(1 + 1/(2\delta))\|\Delta x_{k+n+s-1}\| \leq$$

$$\leq (1 + 1/\delta) K_{s-1}(\|\Delta x_k\|+\ldots+\|\Delta x_{k+n+s-2}\|) + M(1 + 1/(2\delta))\|\Delta x_{k+n+s-1}\| \leq$$

$$\leq K_s(\|\Delta x_k\|+\ldots+\|\Delta x_{k+n+s-1}\|),$$

con $K_s = \max\{(1 + 1/\delta)K_{s-1}, M(1 + 1/(2\delta))\}$.

Proposición 3.4.

Supongamos las hipótesis de la proposición 3.3 y supongamos que (2.19) se cumple para todo k de la forma in + p con i = 1,2,..., y p un entero positivo fijo (o sea, siempre es posible completar

n incrementos independientes consecutivas). Supongamos además que $x^k \in A$ para todo $k = 0,1,2,\ldots$. Entonces, existe $K > 0$ (independiente de k), tal que

$$\| Bk - J(x^k) \| \leq K \, (\|\Delta x_{k-1}\| + \ldots + \|\Delta x_{k-2n+1}\|)$$

para todo $k = 2n-1, 2n, 2n+1, \ldots$

Demostración.

Sea $K = \max \left\{ K_s \right\}_{s=1}^{n-1}$.

En el conjunto de índices $k-2n+1,\ldots,k-n$ necesariamente hay uno (llamémosle j) de la forma $in + p$. En el peor de los casos es $k-2n+1$. Aplicando la proposición 3.3 con j remplazando a k, resulta

$$\| B_{j+n+s} - J(x^{j+n+s}) \| \leq K_s (\|\Delta x_j\| + \ldots + \|\Delta x_{j+n+s-1}\|).$$

Ahora, si $j+n+s = k$ (luego $s = k-j-n$),

$$\| B_k - J(x^k) \| \leq K_{k-j-n} (\|\Delta x_j\| + \ldots + \|\Delta x_{k-1}\|).$$

Pero $j \geq k-2n+1$, luego $k-j-n \leq n-1$, por lo tanto

$\| Bk - J(x^k) \| \leq K \, (\|\Delta x_{k-2n+1}\| + \ldots + \|\Delta x_{k-1}\|)$, como queríamos demostrar.

Teorema 3.1.

Supongamos (2.1) - (2.12) y las hipótesis de la proposición **3.0**. Denotemos $E_k = x^k - x^*$. Entonces,

a) Existe $K > 0$, independiente de k tal que

$$\| Bk - J(x^k) \| \leq K \, (\|E_k\| + \ldots + \|E_{k-2n+1}\|).$$

b) Si $\lim x^k = x^*$ (ver en [5] las condiciones que garantizan esta hipótesis), entonces, existe $R > 0$, independiente de k, tal que

$$\| E_{k+1} \| \leq K \| E_k \| (\|E_k\| + \ldots + \|E_{k-2n+1}\|) \text{ para todo } k \geq 2n - 1.$$

c) Bajo las hipótesis de b), x^k converge a x^* con R-orden mayor o igual que la raiz positiva de $t^{2n} - t^{2n-1} - 1 = 0$.

Demostración.

La parte a) se sigue inmediatamente de la proposición 3.4.

La deducción de b) es clasica: Si $x^{k+1} = x^k - B_k^{-1} F(x^k)$, entonces

$$\|F(x^{k+1}) - F(x^k) + J(x^k)B_k^{-1}F(x^k)\| \leq \|B_k^{-1}F(x^k)\|^2,$$ por el lema (2.1).

Pero $\lim E_k = 0$ implica que $\lim B_k = J(x^*)$ por la parte a), luego $\lim B_k^{-1} = J(x^*)^{-1}$. Por lo tanto , existe $K_1 > 0$ tal que

$$\|F(x^{k+1}) - F(x^k) + J(x^k)B_k^{-1}F(x^k)\| \leq K_1\|F(x^k)\|^2.$$

Ahora, por el lema 2.1, existe $K_2 > 0$ tal que

$$\|F(x^k)\| \leq K_2 \|E_k\| \text{ para todo } k = 0,1,2,\ldots,$$

luego por la parte a) del teorema:

$$\|F(x^{k+1})\| \leq K_3(\|E_k\|+\ldots+\|E_{k-2n+1}\|)\|E_k\|,$$

con $K_3 > 0$.

Ahora, $J(x^*)$ no singular, junto con el lema 2.1, implican que existe $K_4 > 0$ que cumple

$$\|F(x^{k+1})\| \geq K_4\|E_{k+1}\| \text{ para todo } k = 0,1,2,\ldots,$$

por lo tanto b) se sigue con $K = K_3/K_4$.

Finalmente, c) resulta de b) por el teorema 9.2.9 de [8].

4.-Consideraciones finales.

Si el resultado obtenido en este articulo no puede ser mejorado, el método de Broyden-Gay-Schnabel , tendría un R-orden de convergencia inferior al del método secante secuencial (ver [1], [6] , [8] ; [10] (cuyo R-orden es la raíz positiva de $t^{n+1} - t^n - 1 = 0$). No obstante otros hechos lo hacen plenamente justificable en relación a aquél.

En particular, la modificación propuesta para B_k es la de norma de Frobenius más chica entre todas las que cumplen $B_{k+1}\Delta x_k = \Delta F_k, \ldots$ $B_{k+1}\Delta x_{k-\ell_k} = \Delta F_{k-\ell_k}$. Esto hace que en implementaciones en donde B_k es periódicamente repuesto como una aproximación al jacobiano en x^k (o aún cuando B_o por algún motivo es una buena aproximación de $J(x^o)$ y x^o está próximo de x^*), B_{k+1}, B_{k+2}, etc. resulten mejores aproximaciones a los verdaderos jacobianos que si la modificación se hiciera por las reglas del método secante secuencial. De un modo general podriamos decir que este método es una adecuada combinación de las virtudes del método de Broyden tradicional y del método secante secuencial. Con respecto al Q-orden de convergencia, que como es sabido es más fuerte que el R-orden, el mejor resultado obtenido hasta el momento es el de superlinealidad que figura en [5].

Referencias.

1) J.G.P. Barnes (1965), An algorithm for solving nonlinear equations based on the secant method, Computer Journal, 8, pp. 66-72.

2) C.G. Rroyden (1965), A class of methods for solving nonlinear simultaneous equations, Math. Comput., 19, pp. 577-593.

3) J.E. Dennis y J.J. Moré (1974), R characterization of superlinear convergence and its application to quasi-Newton Methods, Math. Comput 28, pp. 549-560.

4) D.M. Gay (1977), Some convergence properties of Broyden's method, Working paper No. 175, NBER.

5) D. M. Gay y R.B. Schnabel (1977), Solving çystems of nonlinear equations by Broyden's method with projected updates, Working paper No. 169, NBER.

6) W.B. Gragg y G.W. Stewart (1976), A stahle variant of the secant method for solving nonlinear equations, SIAM J. of Numer . Anal., 13, pp.

7) J.J. Moré y J. Tragenstein (1976), On the global convergence of Eroyden's method, Math. Comput., 30, pp. 523-540.

8) J.M. Ortega y W.C. Rheinboldt (1970), Iterative solution of nonlinear equations ir, several variables, Academic Press, New York.

9) M.J.D. Powell (1970), A hybrid method for nonlinear equations, en Rabinovitz,P. (editor), Numerical methods for nonlinear algehraic equations, Gordon & Breach, London.

10) P. Wolfe (1959), The secant method for solving nonlinear equations, Comm. ACM, 12, pp. 12-13.

III.- METODOS DE BRENT Y BROWN GENERALIZADOS

1.- Preliminares

Sea $F : S \subset R^n \longrightarrow R^n$ de clase C^1 en el abierto S y sea $x^* \in S$ tal. que $F(x^*) = 0$.

Nos conciernen métodos para resolver $F(x) = 0$ que no hacen uso de derivadas analíticas.

Como es bien sabido, cuando las derivadas analíticas son remplazadas por diferencias finitas (i.e. $\partial f_i / \partial x_j \sim \overline{\partial f_i / \partial x_j} = (f_i(x_1,\ldots,x_j+h,\ldots,x_n) - f_i(x_1,\ldots,x_n))/h$), entonces el método de Newton discreto (MND):

$$x^{k+1} = x^k - \left[\bar{J}(x^k)\right]^{-1} F(x^k) ,$$

con $\bar{J} = (\overline{\partial f_i / \partial x_j})$, tiene orden 2, si $h_k = O(\| F(x^k) \|)$ (Ver [10]).

Ahora, una iteración del MND involucra n+1 evaluaciones de cada función f_i, por lo tanto $n^2 + n$ evaluaciones individuales de función.

Brown ([2]) propuso un método que, preservando convergencia cuadrática, hace solamente $(n^2 + 3n) / 2$ evaluaciones de función por iteración.

Sin embargo, en su implementación original, el método de Brown hacía $O(n^4)$ multiplicaciones por iteración además de las evaluaciones funcionales.

Gay ([4] , [5]), siguiendo una idea de Xrent ([1]), superó esa dificultad modificando el método de tal manera que solamente se hacen $O(n^3)$ multiplicaciones por iteración.

Por otro lado, Brent ([1]), propuso un método que aún usa $(n^2 + 3n)/2$ evaluaciones individuales de función por iteración, pero que es numéricamente más estable (aunque más caro) que el método de Brown (Ver [3]).

En el mismo artículo, Brent desarrolló la idea de Shamanskii ([13]) que, cuando aplicada al MND, consiste en dejar $[J(x^k)]^{-1}$ fijo durante algunos pasos, suprimiendo la computación e inversión de J . El número óptimo de "pasos de refinamiento" es calculado de manera de maximizar la eficiencia de Ostrowski del método ([11]).

La idea de Shamanskii fue aplicada por Brent a su propio método y puede también ser aplicada al método de Brown.

El esquema común a los métodos de Brent y Brown (convendremos en llamar "de tipo Brown" a los métodos con tal esquema) es el siguiente: Dado x^k , $y_1^{li} = x^k$; entonces y_{j+1}^{Ir} se obtiene de y_j^k evaluando la función f_j en n-j+2 puntos.

Si la idea de Shamanski no es utilizada, $x^{k+1} = y_{n+1}^k$; de lo contrario y_{n+1}^k es el primer punto para el refinamiento iterativo. Entonces, excluyendo el refinamiento iterativo, cada iteración involucra:

n+1 evaluaciones de f_1
n evaluaciones de f_2
⋮
2 evaluaciones de f_n .

En la implementación del método, las funciones deben ser ordenadas en creciente dificultad de evaluación (funciones lineales primero, etc.).

Ahora bien, en general, los n+j-2 puntos en que f_j es evaluada no son un subconjunto de los puntos en que f_{j-1} es evaluada.

Por lo tanto, los métodos de tipo Brown no explotan los casos en los cuales una paste sustancial del trabajo necesario para evaluar una función es común a la evaluación de otras.

Por otra parte el MND sí explota esa situación, pero la ventaja de poner las funciones más caras al final se pierde.

El propósito de este artículo es desarrollar una clase de algoritmos que combinen ambas ventajas. Brevemente, el conjunto de funciones $f_1,\ldots,f_n$ se divide en N bloques, de manera que las funciones del bloque j son evaluadas en los mismos $p \bullet q$ puntos , donde p es el numero de puntos donde se evalua el bloque j-1 y q es el número de funciones de ese bloque (el primer bloque se evalúa, al igual que en los métodos de Brown y Brent, en n+1 puntos). Esta clase de métodos incluye el MND (un hloque con n funciones) y los métodos de tipo Brown (n bloques con una función cada uno),

La idea de Shamanskii es también incorporada, siguiendo las idcas del artículo de Brent [1] .

La sección 2 es una descripción de ios algoritmos, se prueba que proveen soluciones exactas para sistemas lineales y se muestra que los métodos de Brent y Erown pertenecen a la clase definida.

La sección 3 es una prueba de la convergencia local y el orden de los métodos.

En la sección 4 se explica cómo se usa la idea de Shamanskii y se da un ejemplo de la aplicación del método de Brent generalizado,

En la sección 5 se asientan algunas conclusiones; se puntualizan futuros desarrollos y se discute la aplicación a minimización de funciones escalares.

2.- Descripción de los algoritmos.

Sea $\mathcal{F} = \mathcal{F}(K_1 , K_2)$ el conjunto de matrices de orden menor o igual que n tal que para todo $A \in \mathcal{F}$, A^{-1} existe, $\|A\| \leq K_1$ y $\|A^{-1}\| \leq K_2$. ($\|.\|$ denotará la norma 2 durante toda esta sección). Supongamos K_1 y K_2 tales que $\mathcal{F}(K_1,K_2) \neq \emptyset$.

Sea N un entero positivo menor o igual que n y ν_i , i=1,...,N una secuencia finita de enteros positivos tal que

$$\sum_{i=1}^{N} \nu_i = n \quad .$$

Sea $L \geq 1$ un entero.

Describamos ahora una iteración del algoritmo definido por K_1, K_2, ν_i y L.

Supongamos dados $x^i \in R^n$; $B^{'i}$ una matriz de nxn en $\mathcal{F}$, $h^i \neq 0$. Entonces los paços del algoritmo para obtener x^{i+1} son los siguientes: (omitimos el superíndice i para simplificar la notación)

1) $A_1 = E$; $y_{1,0} = x$.

2) Ejecutar los paços 3 hasta 5 desde $j = 1$ hasta N.

3) Definir

$$p = \sum_{k=1}^{j-1} \nu_k \quad (\text{si } j = 1,\ p = 0),\ q = \nu_j \ ;$$

$$A_j = \left[c_1^j, \ldots, c_p^j, c_{p+1}^j, \ldots, c_n^j\right] .$$

Computar

$$(a_j)_{rs} = \left(f_{p+r}(y_{j,0} + h\, c_s^j) - f_{p+r}(y_{j,0}) \right) / h$$

$$1 \leq r \leq q$$

$$\text{si } s \geq p+1 \qquad (2.1)$$

$$= 0 \text{ si } s < p+1 .$$

4) Encontrar $\hat{P}_j$ una matriz de (n-p)x(n-p) tal que si

$$P_j = \begin{bmatrix} I_{pxp} & 0 \\ 0 & \hat{P}_j \end{bmatrix} \ ;$$

entonces $P_j \in \mathcal{F}$ y $a_j P_j$ tiene la forma:

$$a_j P_j = \left[0 : S_j : 0\right] \qquad (2.2)$$

(S_j^{qxq} es una matriz triangular inferior),

5) Computar $A_{j+1} = A_j P_j$ y

$$y_{j+1,0} = y_{j,0} - \left[c_{p+1}^{j+1}, \ldots, c_{p+q}^{i+1}\right] S_j^{-1} \begin{bmatrix} f_{p+1}(y_{j,0}) \\ \vdots \\ f_{p+q}(y_{j,0}) \end{bmatrix}$$

6) **Si** $L = I$, ir a 8), de lo contrario ejecutar el paso 7 desde $m = 1$ hasta L-1.

7) $y_{1,m} = y_{N+1,m-1}$

Para $j = 1, \ldots, N$ computar:

$$q = \nu_j \; ; \; p = \sum_{k=1}^{j-1} \nu_k \quad (\text{o } p = 0 \text{ si } j = 1)$$

$$y_{j+1,m} = y_{j,m} - \left[c_{p+1}^{N+1}, \ldots, c_{p+q}^{N+1}\right] S_j^{-1} \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix}$$

(2.3)

8) $x^{i+1} = y_{N+1,L-1}$.

Teorema 2.1

Si $F(x) = 0$ es un sistema lineal no singular arbitrario y x^i, B^i, h^i son arbitrarios, entonces x^{i+1} es la solución del sistema.

Demostración:

Probemos por inducción que:

a) $y_{j+1,0}$ es una solución de $f_k(x) = 0$; $k = 1, \ldots, \sum_{\ell=1}^{j} \nu_\ell$

b) Las Últimas $n - \sum_{\ell=1}^{j} \nu_\ell$ columnas de A_{j+1} forman una base del núcleo del sistema homogéneo asociado a $f_k(x) = 0$, $k = 1, \ldots, \sum_{\ell=1}^{j} \nu_\ell$.

Escribamos $F(x) = E\,x + b$; $E = \begin{bmatrix} E_1 \\ \vdots \\ E_N \end{bmatrix}$ siendo cada E_i una matriz con ν_i filas y n columnas; $b = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix}$.

Para $j = 1$, por 2.2.1, $a_j = E_1\,A_1$,

Pero el sistema $E_1\,x = b_1$ es equivalente a

$$E_1\,A_1\,A_1^{-1}\,x = b_1$$

y a:

$$E_1\,A_1\,P_1\,(A_1\,P_1)^{-1}\,x = b_1 \ ,$$

una solución del cual, naturalmente, es computada por el paao 5 del algoritmo.

Más aún, por (2.2) las últimas $n - \nu_1$ colurnnas de $E_1\,A_1\,P_1$ son ceros, y por lo tanto , las últimas $n - \nu_1$ colurnnas de $A_2 = A_1\,P_1$ son una base del subespacio ortogonal a las filas de E_1, como queríamos probar.

Supongamos ahora la tesis para j-1. Entonces $y_{j,0}$ es una solución de $E_1\,x = b_1$, ... , $E_{j-1}\,x = b_{j-1}$ y las Ultimas $n-\nu_1-\cdots-\nu_{j-1}$ cplumnas de A_j son una base del subespacio ortogonal a las filas de $E_1,\ldots,E_{j-1}$.

Por (2.3), $y_{j+1,0} - y_{j,0} + z_j$ donde z_j es una combinación de esas columnas de A_j ; por lo tanto $y_{j+1,0}$ es también una solución de $E_1\,x = b_1,\ldots,\ E_{j-1}\,x = b_{j-1}$.

Ahora, el sistema:

$E_j\,x = b_j$ es equivalente a

$E_j\,A_j\,A_j^{-}\,x = b_j$ y también a

$E_j A_j A_j^{-1} z_j = b_j - E_j y_{j,0}$; sistema para el cual, trivialmente, los pasos 4 y 5 del algoritmo hallan una solución.

Por lo tanto $F(y_{N+1,0}) = 0$; por lo tanto $y_{j,m} = y_{N+1,0}$ para todo $j = 1,\ldots,N+1$; $m=1,\ldots,L-1$ y el teorema está probado.

Ejemplos

1) Método de Rrent generalizado: Si $K_1 = K_2 = 1$ y $\overset{h}{P}_j$ es elegido como un producto de matrices ortogonales (Transformaciones de Householder ([7]) o haciendo uso del procedimiento de ortogonalización do Powell ([12]) obtenemos la generalización del método presentado por Brent en ([1]).

2) Método de Brown generalizado: La matriz a_j es triangulada en la siguiente forma:

1.- Para $j=1,\ldots q$ ejecutar los pasos 2 y 3,

2.- Permutar las columnas de a_j de manera que

$|aj,p+j| = \max \{|a_{j,p+j}| ,\ldots, |a_{j,n}|\}$ (pivotaje).

3.- Eliminar los elementos $p+j+1, \ldots, n$ en la fila j sustrayendo a la columna respectiva un múltiplo adecuado de la "nueva " columna $p+j$.

Después de este proceso la matriz tiene la forma (2.2).

Ahora, por construcción, la matriz P_j es un producto alternado de matrices de permutación y matrices triangulares superiores con unos en la diagonal y elementos con módulo menor o igual que uno en las otras entradas. Para ser precisos:

$$\hat{P}_j = \Pi_1 . E_1 . \Pi_2 . E_2 \ldots \Pi_q . E_q ,$$

donde Π_k es o bien una permutación elemental o la identidad $(n-p)x(n-p)$, y E_k tiene la forma:

$$\begin{bmatrix} I_{(k-1)x(k-1)} & \vdots & 0 & : & 0 \\ 0 & . & 1 & : & \alpha_{k,k+1} \cdots \alpha_{k,n-p} \\ 0 & . & 0 & : & I_{(n-p-k)x(n-p-k)} \end{bmatrix}$$

con $|\alpha_{ks}| \leq 1$; $s = k+1,\ldots,n-p$.

Un sencillo cálculo muestra que $\| E_k \| \leq \sqrt{1+n-p}$. Más aún E_k^{-1} es la misma matriz con los signos de los α_{ks} cambiados, de manera que $\| E_k^{-1} \| \leq \sqrt{1+n-p}$.

Por supuesto, $\| TT_k \| = \| TT_k^{-1} \| = 1$, de manera que

$$\| P_j \| \leq (n+1)^{n/2} \geq P_j^{-1} .$$

En consecuencia, poniendo $K_I = K_2 = (n+1)^{n/2}$ obtenemos que este método pertenece a la clase definida en la primera sección.

3.- Un teorema de convergencia local.

En lo que sigue, generalizamos el teorema de convergencia dado por Brent en [1] .

Hipótesis generales.

Sea $S \subset R^n$ un abierto convexo que contiene a x^* , $F: S \longrightarrow R^n$ una función de clase C^1 en S, y $F(x^*) = 0$, donde $F = (f_1,\ldots,f_n)^t$,

$$J(x) = (\partial f_i/\partial x_j)(r), \text{ y } J = J(x^*) \text{ (notación).}$$

Supongamos que $J(x)$ satisface una condición de Lipschitz en S, esto es: para todo $x, y \in S$

$$\| J(x) - J(y) \| \leq M \| x-y \| . \qquad (3.1)$$

Esta condición (ver [10]) implica que para todo $x, y \in S$

$$\| F(y) - F(x) - J(x)\,(y-x) \| \leq (M/2) \| y-x \|^2 . \qquad (3.2)$$

En todo lo que sigue supondremos que se genera una sucesión por un algoritmo de la clase definida en §2. Por lo tanto supondremos dados L, la sucesión finita ν_i, K_1 y K_2 .

Recalquemos que el superíndice "i" será usado para el número de iteración y será omitido para simplificar la notación, cuando esto no conduzca a confusión.

Lema 3.1

Dado $K_3 > 0$, existen $\varepsilon_1 > 0$, $K_4 > 0$, $K_5 > 0$ tales que si $\|y^i_{1,0} - x^*\| \leq \varepsilon_1, \ldots, \|y^i_{j,0} - x^*\| \leq \varepsilon_1$, $0 < |h^i| < K_3\, \varepsilon_1$, entonces S^i_j es invertible,

$$\|(S^i_k)^{-1}\| \leq K_4 \|J^{-1}\| \quad \text{para todo } k = 1,\ldots,j,$$

y

$$\|y^i_{j+1,0} - x^*\| \leq K_5 \|y^i_{j,0} - x^*\| .$$

Demostración:

Escribamos:

$$y_k = y_{k,0} = y^i_{k,0}$$

$$A_k = A^i_{k,0} = [c^k_1,\ldots,c^k_n] = [c^{i,k}_1,\ldots,c^{i,k}_n]$$

$$p_1 = 0;\ p_k = \sum_{\ell<k} \nu_\ell \quad \text{si } k > 1\ ;\ q_k = \nu_k .$$

Llamemos además:

$$(m_{rs}) = J\, A_{j+1} .$$

Definimos una nueva matriz $\Omega = (\omega_{rs})$ como sigue:

1) Para $k = 1,\ldots j$, si $p_k < r \leq p_{k+1}$ Y $p_k < s \leq p_{k+1}$, entonces ω_{rs} es la entrada $(r-p_k\ ,\ s-p_k)$ de S_k .

2) Si $s > r$ y $r \leq p_{j+1}$, entonces $\omega_{rs} = 0$.

3) En los demás casos $\omega_{rs} = m_{rs}$.

Escribimos , por simplicidad $p = p_k$, $q = q_k$.

Definimos $\tilde{a}_k$ una matriz de $qx(n-p)$, como sigue:

$$(\bar{a}_k)_{rs} = (f_{p+r}(y_k+hc^k_{p+s}) - f_{p+r}(y_k))/h$$

$r=1,\ldots,q$; $s=1,\ldots,n-p$

Si ε_1 es tan pequeño que tanto y_k como $y_k+hc^k_{p+s}$ están en S, entonces, por (3.2):

$$\left\| \begin{bmatrix} f_{p+1}(y_k+hc^k_{p+s}) \\ \vdots \\ f_{p+q}(y_k+hc^k_{p+s}) \end{bmatrix} - \begin{bmatrix} f_{p+1}(y_k) \\ \vdots \\ f_{p+q}(y_k) \end{bmatrix} - \begin{bmatrix} \nabla f^t_{p+1}(y_k) \\ \vdots \\ \nabla f^t_{p+q}(y_k) \end{bmatrix} c^k_{p+s}\, h \right\| \leq$$

$$\leq (M/2)h^2 \left\|c^k_{p+s}\right\|^2 \leq K_6 h^2 \text{ , con } K_6 = (M/2)\, K_1^{2N} .$$

Entonces, para $0 < |h| \leq K_3\, \varepsilon_1$,

$$\left\| \begin{bmatrix} (f_{p+1}(y_k+hc^k_{p+s}) - f_{p+1}(y_k))/h \\ \vdots \\ (f_{p+q}(y_k+hc^k_{p+s}) - f_{p+q}(y_k))/h \end{bmatrix} - \begin{bmatrix} f^t_{p+1}(y_k) \\ \vdots \\ f^t_{p+q}(y_k) \end{bmatrix} c^k_{p+s} \right\| \leq K_7 \quad 1$$

(3.3)

con $K_7 = K_6\, K_3$.

Ahora, por (3.1), $\|y_k - x^*\| \leq \varepsilon_1$ implica

$$\left\| \begin{bmatrix} \nabla f^t_{p+1}(y_k) \\ \vdots \\ \nabla f^t_{p+q}(y_k) \end{bmatrix} - \begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} \right\| \leq M \left\|y_k - x^*\right\| \leq M\, \varepsilon_1 \quad .$$

(3.4)

Por lo tanto, combinando (2.3.3) con (2.3.4):

$$\left\| \begin{bmatrix} (f_{p+1}(y_k+hc^k_{p+s})-f_{p+1}(y_k))/h \\ \vdots \\ (f_{p+q}(y_k+hc^k_{p+s})-f_{p+q}(y_k))/h \end{bmatrix} - \begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} c^k_{p+s} \right\| \leq K_8 \varepsilon_1 \qquad (3.5)$$

con $K_8 = K_7 + MK_1^N$.

Definimos

$$\bar{b}_k = \begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} \left[c^k_{p+1}, \ldots, c^k_n \right] .$$

Por (3.5) y la definición de $\bar{a}_k$:

$$\| \bar{a}_k - \bar{b}_k \| \leq K_9 \varepsilon_1 ,$$

donde $K_9 = \sqrt{n}\, K_8$.

Ahora, por definición:

$$\left[S_k : 0 \right] = \bar{a}_k \bar{H}_k$$

con $\bar{H}_k = \tilde{P}_k \ldots \tilde{P}_j$; $\tilde{P}_\ell \in \mathcal{P}$; $\ell = k, \ldots, j$; $\tilde{P}_\ell = \begin{bmatrix} I(p_\ell - p) & 0 \\ 0 & \hat{P}_\ell \end{bmatrix}$

Entonces:

$$\| \left[S_k : 0 \right] - \bar{b}_k \bar{H}_k \| = \| (\bar{a}_k - \bar{b}_k) \bar{H}_k \| \leq$$

$$\leq \| \bar{a}_k - \bar{b}_k \| \, \| \bar{H}_k \| \leq K_{10} \varepsilon_1 \qquad (3.6)$$

con $K_{10} = K_9 K_1^n$.

Ahora, por construcción:

$$\bar{b}_k \bar{H}_k = \begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} \left[c^{j+1}_{p+1}, \ldots, c^{j+1}_n \right] . \qquad (3.7)$$

Luego, por (3.6) y 7 (3.7) para k=1,...,j y por la definición de Ω :

$$\| \Omega - J A_{j+1} \| \leq K_{11} \varepsilon_1 \qquad (3.8)$$

con $K_{11} = \sqrt{N}\ K_{10}$.

En consecuencia:

$$\| \Omega A_{j+1}^{-1} - J \| \leq K_{12}\ \varepsilon_1$$

con $K_{L2} = K_{11}\ K_2^{N+1}$.

hhora, J es invertible, por lo tanto, si ε_1 es suficientemente chico, $\Omega\ A_{j+1}^{-1}$ es también invertible y por lo tanto Ω es invertible.

Sea ahora H una matriz ortogonal de la forma :

$$H = \begin{bmatrix} I_{p_j+1} & 0 \\ 0 & H' \end{bmatrix}$$

tal que ΩH es triangular inferior. Es claro entonces que Ω invertible implica que ΩH es invertible y por lo tanto S_k es invertible para todo $k = 1,..,j$. Además $(\Omega H)^{-1}$ tiene.en sus j primeros bloque diagonales a S_k^{-1}. Luego:

$$\| S_k^{-1} \| \leq \| (\Omega H)^{-1} \| = \| \Omega^{-1} \| . \qquad (3.9)$$

Ahora, por (3.8), si ε_1 es suficientemente chico, entonces:

$$\| \quad - (J\ A_{j+1})^{-1} \| \leq \| J^{-1} \| \text{ , por ejemplo.}$$

Luego:

$$\| \Omega^{-1} \| \leq \| (J\ A_{j+1})^{-1} \| + \| J^{-1} \|$$

y por lo tanto:

$$\| \Omega^{-1} \| \leq K_{12} \| J^{-1} \| \qquad (3.10)$$

con $K_4 = 1 + K_2^{N+1}$.

Por (3.9) y (3.10):

$$\| S_k^{-1} \| \leq K_4 \| J^{-1} \|$$

Ahora, por (3.2), pnniendo $p = p_j$, $q = q_j$:

$$\left\| \begin{bmatrix} f_{p+1}(y_{j,0}) \\ \vdots \\ f_{p+q}(y_{j,0}) \end{bmatrix} \right\| \leq \|J\| \, \|y_{j,0} - x^*\| + (M/2) \, \|y_{j,0} - x^*\|^2 \leq$$

$$\leq K_{13} \, \|y_{j,0} - x^*\|$$

con $K_{13} = \|J\| + M/2$ (suponiendo, digamos, que $\varepsilon_1 < 1$).

Entonces, por definición de $y_{j+1,0}$:

$$\|y_{j+1,0} - x^*\| \leq K_5 \, \|y_{j,0} - x^*\| \qquad (3.11)$$

donde $K_5 = 1 + K_1^{N+1} \, K_4 \, \|J^{-1}\| \, K_{13}$; con lo cual el lema está probado.

Lema 3.2.

Dado $K_3 > 0$, existen $\varepsilon_2 > 0$, $K_{14} > 0$, $K_{15} > 0$ tales que si $\|x^i - x^*\| \leq \varepsilon_2$ y $0 < |h^i| < K_3 \, \varepsilon_2$, entonces

1) $\|y^i_{j,m} - x^*\| \leq K_{14} \, \|y^i_{1,m} - x^*\|$ (3.12)

para todo $j = 2,\ldots,N+1$ y $m = 0,1,\ldots,L-1$, y

2) S^i_j es invertible y $\|(S^i_j)^{-1}\| \leq K_{15} \, \|J^{-1}\|$ para todo $j=1,\ldots N$. (3.13)

Demostración:

Sean ε_1, K_4 and K_5 los números positivos que son mencionados en el lema 2.3.1 . Definimos $\varepsilon_2 = \varepsilon_1 / K_5^{NL}$. Supongamos que:

$$\|x^i - x^*\| \leq \varepsilon_2 \; ; \quad 0 < |h^i| \leq K_3 \, \varepsilon_2 \, . \qquad (3.14)$$

Probamos por inducción que :

$$\|y^i_{j,m} - x^*\| \leq \varepsilon_1 / K_5^{NL-(mN+j-1)}, \qquad (3.15)$$

$j=1,\ldots,N+1$; $m=0,1,\ldots,L-1$.

Para $j=1$ y $m=0$ (3.15) se sigue de (3.14).

Fijemos m' y k y supongamos que (3.15) vale para todo j con $m < m'$ y para todo $j \leq k$ con $m = m'$. En particular:

$$\|y^i_{j,0} - x^*\| \leq \varepsilon_1 / K_5^{NL-j+1} \leq \varepsilon_1 \quad \text{para todo } j \leq k,$$

y $0 < |h^i| \leq K_3 \, \varepsilon_2 \leq K_3 \, \varepsilon_1$.

Entonces, por lema 2.3.1 , S^i_j es invertible para todo $j \leq k$ y

$$\|(S_j^i)^{-1}\| \leq K_4 \|J^{-1}\| .$$

Por lo tanto, como en la deducción de (2.3.11),

$$\|y^i_{k+1,m} - x^*\| \leq K_5 \|y^i_{k,m} - x^*\| \leq$$

$$\leq \varepsilon_1 K_5 / K_5^{NL-(mN+k-1)} = \varepsilon_1 / K_5^{NL-(mN+k)} \qquad (3.16)$$

Como $y^i_{N+1,m} = y^i_{1,m+1}$ para todo m, el paso inductivo se sigue directamente cuando m' es incrementado en lugar de k.

Entonces:

$$\|y^i_{j,m} - x^*\| \leq \varepsilon_1 \quad \text{para todo } m = 0,1,\ldots,L-1$$

y $j = 1,\ldots,N+1$,

y por lo tanto (3.12) se prueba facilmente con $K_{14} = K_5^N$ y (3.13) también es válido con $K_{15} = K_4$.

Lema 3.3.

Dado $K_3 > 0$, existe $\varepsilon_3 > 0$, $K_{16} > 0$ tales que si $\|x^i - x^*\| \leq$ $\leq E \leq \varepsilon_3$ y $0 < |h^i| \leq K_3 \varepsilon$, entonces

$$\|y^i_{1,m} - x^*\| \leq K_{16} \varepsilon^{m+1}$$

para todo $m = 0,1,\ldots,L$.

Demostración:

Si $\varepsilon_3 \leq \varepsilon_2 \leq \varepsilon_1$ y $K_{17} = K_{14}^{NL}$, entonces (3.12) lleva a :

$$\|y^i_{j,m} - x^*\| \leq K_{17} \varepsilon \qquad (3.17)$$

para $j = 1,\ldots,N+1$; 0 $,\ldots,L-1$. (Recordar que $y^i_{N+1,L-1} = y^i_{1,L} = x^{i+1}$). Por lo tanto:

$$\|y^i_{j,m} - y^i_{j,0}\| \leq K_{18} \varepsilon$$

para $j = 1,\ldots,N+1$; $m=0,1,\ldots,L-1$; con $K_{18} = 2 K_{17}$.

Ahora, (2.3.1) implica:

$$\|J(y^i_{j,m}) - J(y^i_{j,0})\| \leq M \|y^i_{j,m} - y^i_{j,0}\|$$

(si ε_3 es tan pequeño que tanto $y^i_{j,m}$ y $y^i_{j,0}$ están en S.).

Por lo tanto,

$$\|J(y^i_{j,m}) - J(y^i_{j,0})\| \leq K_{19} \varepsilon \qquad (3.18)$$

para $j = 1,\ldots,N+1$; $m = 0,1,\ldots,L-1$; con $K_{19} = M\, K_{18}$.

También,

$$\|J(y^i_{j,0}) - J(x^*)\| \leq K_{20}\,\varepsilon \qquad (3.19)$$

para $j = 1,\ldots,N+1$; con $K_{20} = M\, K_{17}$.

Definiendo $\Omega = \Omega_{ij}$ como en el lema 2.3.1 y repitiendo el argumento de éste con ε remplazando a ε_1 :

$$\|J(x^*)\, A^i_{j+1} - \Omega_{ij}\| \leq K_{21}\,\varepsilon \quad .$$

con $K_{21} = K_{11}\, K_{97}$. Por lo tanto;

$$\|J(y^i_{j,0})\, A^i_{j+1} - \Omega_{ij}\| \leq K_{22}\,\varepsilon$$

para $j = 1,\ldots,N+1$; con $K_{22} = K_{20}\, K_1^{N+1} + K_{21}$.

Procedemos ahora por inducción en m para probar la tesis. Para $m = 0$ es trivial. Supongamos que tornando ε_3 suficientemente pequeno existe $R_{23,m} > 0$ tal que:

$$\|y^i_{1,m} - x^*\| \leq K_{23,m}\,\varepsilon^{m+1} \quad .$$

Entonces, por (3.12):

$$\|y^i_{j,m} - x^*\| \leq K_{24,m}\,\varepsilon^{m+1} \qquad (3.20)$$

con $K_{24,m} = K_{23,m}\,'14$ para todo $j = 2,\ldots,N+1$.

Ahora, por (2.3.2):

$$\|F(y^i_{j,m})\| \leq \|J\|\,\|y^i_{j,m} - x^*\| + (M/2)\,\|y^i_{j,m} - x^*\|^2 .$$

Entonces,

$$\|F(y^i_{j,m})\| \leq K_{25,m}\,\|y^i_{j,m} - x^*\| \leq K_{26,m}\,\varepsilon^{m+1} \quad , \qquad (3.21)$$

suponiendo, digamos, que $\varepsilon_3 < 1$, donde

$$K_{25,m} = \|J\| + M/2 \qquad \text{y}$$

$$K_{26,m} = K_{25,m}\, K_{24,m} \quad .$$

Fijemos ahora j, $1 < j \leq N$, y pongamos $p = \bigvee_{k<j}$, $q = v_j$. Por (3.2):

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+1,m}) \\ \vdots \\ f_{p+q}(y_{j+1,m}) \end{bmatrix} - \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix} - \begin{bmatrix} \nabla f^t_{p+1}(y_{j,m}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j,m}) \end{bmatrix} (y_{j+1,m} - y_{j,m}) \right\| \leq$$

$$\leq (M/2) \left\| y_{j+1,m} - y_{j,m} \right\|^2 \leq K_{27,m}\, \varepsilon^{2m+2}$$

con $K_{27,m} = 2\, M\, K^2_{24,m}$.

Por lo tanto (desde que $\varepsilon_3 < 1$),

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+1,m}) \\ \vdots \\ f_{p+q}(y_{j+1,m}) \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix} + \begin{bmatrix} \nabla f^t_{p+1}(y_{j,m}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j,m}) \end{bmatrix} (y_{j+1,m} - y_{j,m}) \right\| +$$

$$+ K_{27,m}\, \varepsilon^{2m+2} \leq$$

$$\left\| \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix} + \begin{bmatrix} \nabla f^t_{p+1}(y_{j,0}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j,0}) \end{bmatrix} (y_{j+1,m} - y_{j,m}) \right\| +$$

$$+ \left\| J(y_{j,0}) - J(y_{j,m}) \right\| \left\| y_{j+1,m} - y_{j,m} \right\| + K_{27,m}\, \varepsilon^{2m+2} \leq$$

$$\left\| \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix} + \begin{bmatrix} \nabla f^t_{p+1}(y_{j,0}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j,0}) \end{bmatrix} A_{j+1}\, A^{-1}_{j+1}\, (y_{j+1,m} - y_{j,m}) \right\| +$$

$$+ K_{28,m}\, \varepsilon^{m+2}$$

con $K_{28,m} = 2\, K_{19}\, K_{24,m} + K_{27,m}$.

Definiendo ahora

$$D_1 = \begin{bmatrix} \nabla f^t_{p+1}(y_{j,0}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j,0}) \end{bmatrix} A_j$$

$$D_2 = \begin{bmatrix} (f_{p+1}(y_{j,0}+hc_1^j)-f_{p+1}(y_{j,0}))/h & \dots & (f_{p+1}(y_{j,0}+hc_n^j)-f_{p+1}(y_{j,0}))/h \\ \vdots & & \vdots \\ (f_{p+q}(y_{j,0}+hc_1^j)-f_{p+1}(y_{j,0}))/h & \dots & (f_{p+q}(y_{j,0}+hc_n^j)-f_{p+q}(y_{j,0}))/h \end{bmatrix}$$

resulta, como en la prueba del lema 2.3.1,

$$\|D_1 - D_2\| \leq K_{29}\, h \quad \text{con } K_{29} = \sqrt{n}\ K_6 .$$

Por lo tanto,

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+1,m}) \\ \vdots \\ f_{p+q}(y_{j+1,m}) \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} f_{p+1}(y_{j,m}) \\ \vdots \\ f_{p+q}(y_{j,m}) \end{bmatrix} + D_2\, P_j\, A_{j+1}^{-1}(y_{j+1,m}-y_{j,m}) \right\| +$$

$$+ K_{30,m}\, \varepsilon^{m+2} \qquad (3.22)$$

con $K_{30,m} = 2\, K_{29}\, K_3\, K_1\, K_2^{N+1}\, K_{24,m} + K_{28,m}$.

Pero el primer término del segundo miembro de (2.3.22) es 0 por la definición del algoritmo, luego:

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+1,m}) \\ \vdots \\ f_{p+q}(y_{j+1,m}) \end{bmatrix} \right\| \leq K_{30,m}\, \varepsilon^{m+2} .$$

Dejemos fijos ahora j, p, q y supongamos que hemos probado que:

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+r,m}) \\ \vdots \\ f_{p+q}(y_{j+r,m}) \end{bmatrix} \right\| \leq K_{31,m,r}\, \varepsilon^{m+2} \qquad (3.23)$$

para algún $r \geq 1$ $(K_{31,m,1} = K_{30,m})$. Ahora , por (2.3.2),

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+r+1,m}) \\ \vdots \\ f_{p+q}(y_{j+r+1,m}) \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} f_{p+1}(y_{j+r,m}) \\ \vdots \\ f_{p+q}(y_{j+r,m}) \end{bmatrix} \right\| + \left\| \begin{bmatrix} \nabla f^t_{p+1}(y_{j+r,m}) \\ \vdots \\ \nabla f^t_{p+q}(y_{j+r,m}) \end{bmatrix} (y_{j+r+1} - y_{j+r}) \right\| +$$

$$+ (M/2) \left\| y_{j+r+1,m} - y_{j+r,m} \right\|^2 .$$

Entonces, por (3.1), (3.18), (3.19), (3.20) y (3.22),

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+r+1,m}) \\ \vdots \\ f_{p+q}(y_{j+r+1,m}) \end{bmatrix} \right\| \leq$$

$$\left\| \begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} \left[c^{j+r+1}_{t+1}, \ldots, c^{j+r+1}_{t+v} \right] S^{-1}_{j+r} \begin{bmatrix} f_{t+1}(y_{j+r,m}) \\ \vdots \\ f_{t+v}(y_{j+r,m}) \end{bmatrix} \right\| + K_{32,m,r} \varepsilon^{m+2} \qquad (3.24)$$

con $K_{32,m,r} = K_{31,m,r} + 2\left((K_{20}+K_{19})K_{24,m} + MK^2_{24,m}\right)$,

$t = \sum_{\ell \leq j+r} \nu_\ell$, $v = \nu_{j+r+1}$.

Ahora $\begin{bmatrix} \nabla f^t_{p+1}(x^*) \\ \vdots \\ \nabla f^t_{p+q}(x^*) \end{bmatrix} \left[c^{j+r+1}_{t+1}, \ldots, c^{j+r+1}_{t+v} \right]$ es una matriz formada por las filas p+1 hasta p+q de $J A_{j+r+1}$ desde la columna t+1 hasta la t+v con t+1 > p+q . Por lo tanto, por las mismas razones que conducen a (3.8), su norma es menor o igual que $K_{21} E$. Luego, por (3.13), (3.21) y (3.24),

$$\left\| \begin{bmatrix} f_{p+1}(y_{j+r+1,m}) \\ \vdots \\ f_{p+q}(y_{j+r+1,m}) \end{bmatrix} \right\| \leq K_{33,m,r} \varepsilon^{m+2}$$

donde $K_{33,m,r} = K_{32,m,r} + K_{21}K_{15} \left\| J^{-1} \right\| K_{26,m}$.

Por lo tanto,

$$\left\| \begin{bmatrix} f_{p+1}(y_{N+1,m}) \\ \vdots \\ f_{p+q}(y_{N+1,m}) \end{bmatrix} \right\| \leq K_{33,m,N-j}\ \varepsilon^{m+2} .$$

Repitiendo esta fórmula para $j = 2,\ldots,N+1$ ($p=p_j$, $q = \nu_j$) , obtenemos:

$$\| F(y^i_{N+1,m}) \| \leq K_{34,m}\ \varepsilon^{m+2}$$

con $K_{34,m} = \sqrt{N}\ \max\ \{K_{33,m,j}\ ;\ j = 1,\ldots,N\}$.

Ahora, por (3.2),

$$\| y^i_{N+1,m} - x^* \| \leq \| J^{-1} \| (\| F(y^i_{N+1,m}) \| + (M/2) \| y^i_{N+1,m} - x^* \|^2). \tag{3.25}$$

Luego, usando (2.3.20),

$$\| y^i_{1,m+1} - x^* \| \leq K_{23,m+1}\ \varepsilon^{m+2}$$

con $K_{23,m+1} = \| J^{-1} \| (K_{34,m} + K^2_{24,m}\ M/2)$.

Esta desigualdad completa la prueba inductiva.

Ahora, poniendo $K_{16} = \max \{K_{22,m};\ m=0,1,\ldots,L\}$ se obtiene la tesis.

Teorema 3.1

Dado $K_3 > 0$, existe $K_{35} > 0$, $\varepsilon_4 > 0$ tal que si para algún i_o , $\| x^{io} - x^* \| \leq \varepsilon_4$, y para todo $i \geq i_o$, $0 < |h^i| \leq K_3 \| x^i - x^* \|$, entonces $\lim x^i = x$ y para todo $i \geq i_o$,

$$\| x^{i+1} - x^* \| \leq K_{35} \| x^i - x^* \|^{L+1} \tag{3.26}$$

Demostración:

Sea $K_{35} = K_{16}$, $\varepsilon_4 \leq \min \{\varepsilon_3, K_{35}^{-1/L}\}$ (luego $K_{35}\ \varepsilon_4^L < 1$). Si $\| x^i - x^* \| \leq \varepsilon_4$, entonces por el lema 2.3.3,

$$\| x^{i+1} - x^* \| \leq K_{35} \| x^i - x^* \|^{L+1} .$$

Pero

$$\|x^{i+1} - x^*\| / \|x^i - x^*\| \leq K_{35}\ \varepsilon_4^L < 1\ ,$$

(3.27)

por lo tanto .podemos probar por inducción que $\|x^1 - x^*\| \leq \varepsilon_4$ para todo $i \geq i_o$ y por lo tanto (2.3.26) es válido para todo $i \geq i_o$ y (3.27) implica que $\lim x^i = x^*$.

Comentarios

1) (3.2) y (3.25) implican que $\|x^i - x^*\|$ es del mismo orden que $\|F(x^i)\|$; por lo tanto la hipótesis $|h^i| \leq O(\|x^i - x^*\|)$ puede ser remplazada por $|h^i$ o $\|F(x^i)\|)$. En términos prácticos h^i debe ser elegido por un compromiso entre las desigualdades anteriores y la necesidad computacional de evitar la cancelación tanto como sea posible en el cálculo de las a_j .

2) El anterior teorema y lemas y la definición de los sucesivos ε's y R's sugieren que el radio de convergencia es mayor cuanto más chicos sean L, N, K_1, K_2, M, $\|J\|$, y $\|J^{-1}\|$. Las últimas tres son constantes que dependen solamente de F. En la práctica, es cierto que cuando M, $\|J\|$, $\|J^{-1}\|$ son grandes el cero es más difícil de encontrar; la primera provee una medida de la no linealidad y las dos últimas de la estabilidad del sistema. La experiencia numérica (ver [3] también confirma que K_1 y K_2 deben ser pequenos, por lo menos para el caso L = 1, y $\nu_k = 1$, k = 1,...,n (métodos no generalizados). Por fin, no sabemos si el método con L = 1, N = 1 (MND sin refinamientos) tiene una región de convergencia más amplia que otras variantes.

4.- Maximización de la eficiencia.

La eficiencia asintótica de un método (ver [1]) se define como

$$E = \log_2 O \ / \ W$$

donde O es el orden del método y W es el monto de trabajo requerido por iteración. Este es el logaritmo del índice de eficiencia de Ostrovski ([11]). E puede ser intuitivamente pensado como la inversa del monto de trabajo necesario para doblar el número de dígitos correctos en la solución aproximada ya alcanzada.

De acuerdo con el teorema 3.1, diferentes valores de L dan lugar a diferentes órdenes y, por lo tanto, a diferentes eficiencias de los métodos. Es natural, entonces, preguntarse por el valor de L

que hacc alcanzar **a** la eficiencia su máximo posible valor.

Si. se postula que cada componente tiene un costo de evaluación igual a 1, y que no se puede ahorrar trabajo evaluando más de una conponente al rnismo tiempo, entonces la mejor disposición de los bloques es la definida por $\nu_i = 1$; i=1,...,n, y el análisis de cuál L es el óptimo para diferentes valores de n puede encontrarse en [1].
Si ése no es el caso, un cálculo separado del L óptimo debe hacerçe para cada función particular F (este cálculo, por supuesto, puede ser ejecutado facilmente por el programa de computación).

El siguiente ejemplo ilustra una situación donde una disposición no standard d.e los bloques es mejor que las disposiciones standard. Tales casos ocurren cuando se estiman parámetros de ciertos sistemas dinámicos discretos.

Supongamos que tenemos el sistema en diferencias finitas:

$$x_1(t+1) = \varphi_1(a_1,\ldots,a_9,x_1(t),x_2(t),x_3(t))$$
$$x_2(t+1) = \varphi_2(a_1,\ldots,a_9,x_1(t),x_2(t),x_3(t))$$
$$x_3(t+1) = \varphi_3(a_1,\ldots,a_9,x_1(t),x_2(t),x_3(t))$$
$$t = 0, 1,2,\ldots t_1$$
$$x_4(t+1) = \varphi_1(a_1,\ldots,a_9,x_4(t),x_5(t),x_6(t))$$
$$x_5(t+1) = \varphi_2(a_1,\ldots,a_9,x_4(t),x_5(t),x_6(t))$$
$$x_6(t+1) = \varphi_3(a_1,\ldots,a_9,x_4(t),x_5(t),x_6(t))$$
$$t = 0,1,2,\ldots,t_2$$
$$x_7(t+1) = \varphi_1(a_1,\ldots,a_9,x_7(t),x_8(t),x_9(t))$$
$$x_8(t+1) = \varphi_2(a_1,\ldots,a_9,x_7(t),x_8(t),x_9(t))$$
$$x_9(t+1) = \varphi_3(a_1,\ldots,a_9,x_7(t),x_8(t),x_9(t))$$
$$t = 0,1,2,\ldots,t_3$$

donde el estado inicial $(x_1(0)\ldots,x_9(0))$ y el estado final $(x_1(t_1), x_2(t_L), x_3(t_1), x4(t_2), x_5(t_2), x_6(t_2), x_7(t_3), x_8(t_3), x_9(t_3))$ son conocidos y los coeficientes $(a_1,\ldots,a_9)$ deben ser obtenidos. Este problema I leva a un sistema de ecuaciones no lineales $f_1 = 0,\ldots, f_9 = 0$ donde, claramente , el costo de evaluar f_1, f_2, y f_3 es el mismo que el de evaluar f_1 sola, etc.. Más aún, si t_L e $t_2 < t_3$ el mejor método en la clase definida en §2 es el determinado por $\nu_1 = \nu_2 = \nu_3 = 3$. Podemos pos-

tular que el costo de la evaluación del primer bloque es t_1, del segundo es t_2 y del tercero es t_3. Si t1 = 3, t_2 = 20, t_3 = 50, se obtiene que el L óptimo es 4, lo cual da un método de orden 5, con una eficiencia de 0.394×10^{-2} .

Este ejemplo fue corrido con las siguientes especificaciones:

1) $\varphi_1(a_1,\ldots, a_9, y_1, y_2, y_3) = a_1 \text{ sen } y_1 + a_2 \text{ sen } y_2 + a_3 \text{ sen } y_3$

$\varphi_2(a_1,\ldots, a_9, y_1, y_2, y_3) = a_4 \text{ sen } y_1 + a_5 \text{ sen } y_2 + a_6 \text{ sen } y_3$

$\varphi_3(a_1,\ldots, a_9, y_1, y_2, y_3) = a_7 \text{ sen } y_1 + a_8 \text{ sen } y_2 + a_9 \text{ sen } y_3$

2) $x_1^*(0),\ldots, x_9(0)$ y $a_1, \ldots, a_9$ (los "coeficientes verdaderos") generados aleatoriamente entre 0 y 1.

3) El estado final es obtenido corriendo el sistema dinámico con los coeficientes verdaderos.

4) $b_1,\ldots,b_9$ los puntos iniciales para los algoritmos son generados en la forma $b_j = a_j (1 + w_j)$ donde w_j es un número aleatorio entre -0.02 y 0.02.

Se corrieron veinte tests con el método de Brent generalizado con el L óptimo y L = 1; y los mismos tests con el método de Brent no generalizado y el MND. Como se esperaba, el método de Brent generalizado con $\nu_i = 3$; i = 1,2,3 se comportó mejor que los otros.

He aqui un sumario de los resultados:

Método 1. Rloques (3,3,3). L = opt = 4
Convergencia: 16 experimentos
Divergencia: 4 experimentos
Trabajo total promedio para alcanzar una precisión de 10^{-7} : 865.25

Método 2. Bloques (3,3,3). L = 1
Convergencia : 13 experimentos
Paradas por singularidad: 5 experimentos
Excedido el número de iteraciones permitido (30): 2 experimentos
Trabajo promedio (casos de convergencia): 2269.8

Método 3. Bloques (9) (Newton). L = opt = 7
Convergencia : 14
Divergencia: 5
Singularidad: 1
Trabajo promedio: 1825.

Método 4. Bloques (9), L = I
Convergencia : 11
Singularidad: 6
Exceso de iteraciones: 3
Trabajo promedio: 4652.1

Método 5. Bloques (1,1,1,1,1,1,1,1,1) (No generalizado)
L = opt = 3
Convergencia: 13
Divergencia : 1
Exceso de iteraciones: 2
Singularidad : 4
Trabajo promedio: 3306.2

Método 6. Bloques (1,1,1,1,1,1,1,1,1) (No generalizado)
L= 1
Convergencia: 12
Exceço de iteraciones: 1
Singularidâd: 7
Trabajo promedio: 6621.5

5.- Consideraciones finales

En este articulo hemos introducido una generalización de los métodos de Erown y Brent, que debe resultar más económica en los casos en que es considerahlemente más barato evaluar varias componentes simultaneamente que separadamente. Hemos mostrado que esta generalizaciór tiene propiedades de convergencia similares a las de sus predecesores.

De todos modos, nada se dice sobre canvergencia global y , de hecho, es claro que solo la introducción de parámetros de relajación puede mejorar los métodos en tal sentido. Los resultados locales son sin embargo muy Útiles porque explican el comportamiento de los algoritmos en las reqiones donde la relajación no es necesaria. Una eficiencia asintótica alta es especialmente saludable cuando uno tiene que resolver una secuencia de problemas levemente diferentes, tales que la solución de uno es una sstimación inicial adecuada de la solución del çiguiente. Valiosas sugestiones sobre la introducción de parámetros de relajación pueden encontrarse en [5] .

El costo de una iteración, y, consecuentemente el L óptimo puede cambiar si se toma en cuenta también el trabajo intrínseco de los cálculos ejecutados por los alyoritmos. Esta consideración puede hacer . preferibles los métodos de Brown **a** los métodos de Brent. De todos modos, en los problemas más interesantes, el tiempo de computación está

dominado por el tiempo de Tas evaluaciones funcionales.

No es necesario usar el mismo h^i a lo largo de toda una iteración Todo lo que se necesita es que los cocientes incrementales (2.1) se aproximen a $\langle \nabla f_{p+r} , c_s^j \rangle$ evitando la cancelación tanto como sea posible. Para satisfacer los teoremãs de convergencia sólo es preciso que valga la desigualdad $(h^i| \leq K_3 \|x^i - x^*\|$ por cada uno de los h^i usados durante una iteración. Más aún, si ∇f_{p+r} puede calcularse, calcular $\langle \nabla f_{p+r} , c_{p+s}^j \rangle$ puede ser más barato que computar los cocientes incrementales correspondicntes. Es facil verificar que el teorema de convergencia sigue valiendo si algunos de dichos cocientes son remplazados por los productos escalares.

Otro problema que puede ser considerado separadamente que concierne a la implementación práctica de los algoritmos es el de los criterios de parada. Entre otras (ver [9] para una discusión detallada de criterios para el método de Erent no generalizado) debe haber una condición de parada que se refiera a la norma de $F(x)$. El hecho es que $F(x)$ no necesita ser calculada en los wasos corrientes del algoritmo (diferentes componentes son, en general, evaluadas en diferentes puntos). Gay ([6]) propone un criterio de parada que resuelve esta dificultad elegantemente para el método de Brent generalizado, Su implementación requiere cierta modificación del alçoritmo: los pasos 1 y 2 deben ser remplazados por:

1) Sea $j_o = \min \{ j \ / \ \max \{ |f_k(x)| \ / \ p_j < k \leq p_{j+1} \} > EPS \}$ (si j_o no existe , parar ($\|F(x)\|_\infty \leq EPS$). Sea $A_{j_o}^i = A_{N+1}^{i-1}$, $y_{j_o,0} = x$.

2) Para $j = j_o,\ldots,N$, ejecutar los pasos 3 hasta S.

Si las primeras componentes de F son lineales, entonces este esquema explota el hecho procesándolas sólo en la primera iteración principal, y si son casi lineales seguramente se ahorra trabajo en algunas de las iteraciones principaleç. En este último caso, sin embargo, nada riguroso puede decirse sobre esta forma modificada del algoritmo.

Con respecto a la aplicación de los algoritmos presentados en este artículo a problemas de minimización la referencia obligada es [4] , trabajo en el cual Gay aplicó a dicho tipo de problemas el método de Brown. El punto crucial de la cuestión es la elección de un metodo de relajación, pues es ahí que se puede poner en evidencia

el tipo especial de problema de que se trata. Seguramente nuestros metodos tienen propiedades similares al método de Newton, sin algunos de sus problemas intrínsecos, pero conservando la incertidumbre de aquel en cuanto a la generación de direcciones de descenso fuera de un entorno de la solución.

Referencias

1.- R.P. Brent (1973), Some efficient algorithms for solving systems of nonlinear equations, Siam J. Numer. Anal,, 10, pp 327-344 .

2.- K.M. Brown (1969), A quadratically convergent Newton - like method based on Gaussian elimination, Siam J. Numer. Anal., 6, pp 560-569 .

3.- M. Y. Cosnard (1975), A comparison of four methods for solving systems of nonlinear equations, TR 75-248, Dept. Computer Science, Cornell University.

4.- D.M. Gay (1975), Brown's method and some generalizations with applications to minimization problems, TR 75-225, Dept. Computer Science, Cornell University.

5.- D.M. Gay (1976), Implementing Erown's method, Tech Report CNA-109 Center of Numerical Analysis, The University of Texas at Austin.

6.- D.M. Gay (1977), Comunicación privada.

7.- A.S. Householder (1964), The theory of matrices in numerical analysis, Blaisdell, New York.

8.- J.J. Moré y M.Y. Cosnard (1976), Numerical comparisnn of three nonlinear solvers, Tech Mem No. 286, Argonne National Laboratory.

9.- J.J. Moré y M.Y. Cosnard (1977), On the numerical solution of nonlinear equations, manuscrito, por aparecer en ACM Transactions on Mathematical Software.

10.- J.M. Ortega y W.C. Rheinboldt (1970), Iterative solution of non-Linear equations in several variables, Academic Press, New York.

11.- A.M. Ostrowski (1960), Solution of equations and systems of equations, Academic Press, New York.

12.- M.J.D. Powell (1968), On the calculation of orthogonal vectors, Computer Journal, 11, pp 302-304.

13.- V.E. Shamanskii (1967); A modification of Newton's method, Ukrain Mat Zh., 19, pp 133-138.

CAP. IV : ALGORITMOS ESTABLES BASADOS EN EL METODO SECANTE SECUENCIAL

1.- Preliminares.

El objetivo de este articulo es presentar algunas implementacioncs estables, usando factorizaciones, del método secante secuencial para resolver sistemas algebraicos no lineales sin usar derivadas analíticas. Nos proponemos además evaluar la utilidad y los costos relativos de los métodos presentados en base al análisis de resultados teóricos, posibilidades de extensión, costo-computacionales y aprovechamiento de la información de casos particulares.

Si $F(x) = 0$: $F = (f_1,\ldots,f_n)^t$ es un sistema no lineal de n ecuaciones con n incógnitas, el método secante secuencial (llamado método secante de $n + 1$ puntos en [15]) es la generalización más natural a n variables del método secante para resolver una ecuación con una incógnita. Brevemente, el método consiste en lo siguiente: Si x^{k_1}, x^{k_2},.. ,$x^{k_{n+1}}$ son los puntos corresnondientes a $n + 1$ iteraciones, eventualmente consecutivas, con $F_{k_j} = F(x^{k_j})$, $j = 1,\ldots,n + 1$; entonces x^k será el "Único" cero de la "única" función afín L_k tal que $F_{k_j} = L_k(x^{k_j})$, $j = 1,\ldots,n + 1$. Las comillas están puestas arriba por razones obvias: para que exista una única función afín en las condiciones descritas, con un único cero, es necesario que tanto $\{x^{k_j}\}_{j=1}^{n+1}$ como $\{F_{k_j}\}_{j=1}^{n+1}$ sean afinrnente independientes .

Wolfe ([19]) consideró en 1959 una implementación de dicho método en donde , una vez obtenido x^k, este punto pasaba a remplazar a x^{k_j}, donde

$$\|F_{k_j}\| = \max\{\|F_{k_i}\|\}_{i=1}^{n+1}$$

Asimismo, Wolfe dio uno forma para calcular recursivamente la solución de $L_k(x) = 0$ sin necesidad de resolver un sistema lineal en todas las iteraciones. Llamemos, para simplificar la notación, $x^j = x^{k_j}$, $j = 1,\ldots,n + 1$; $F_j = F_{k_j}$, $j = 1,\ldots,n + 1$. Wolfe observa que x^k se ohtiene mediante la solución del sistema de ecuaciones (en w):

$$\sum_{j=1}^{n+1} w_j f_i(x^j) = 0 \quad , \text{ para } i = 1,\ldots,n$$

$$\sum_{j=1}^{n+1} w_j = 1$$

y la multiplicación:

$$x^k = \sum_{j=1}^{n+1} w_j x^j .$$

El punto crucial es, pues, que w se obtiene de la forma

$$w = A^{-1}.\ (0,\ldots,0,1)^t,\ w \in R^{n+1},$$

donde

$$A = \begin{pmatrix} f_1(x^1) & \ldots & f_1(x^{n+1}) \\ \vdots & & \vdots \\ f_n(x^1) & \ldots & f_n(x^{n+1}) \\ 1 & \ldots & 1 \end{pmatrix}.$$

Suponiendo que la inversa de A está computada y que x^{k+1} debe ser ahora calculado, resulta que una columna de A, digamos, la j-ésima, debe ser remplazada por

$$p = (f_1(x^k),\ldots,f_n(x^k),1)^t.$$

Llamando A* a la nueva matriz, resulta:

$$A^{-1}A^* = \begin{pmatrix} 1 & 0 & \ldots & 0 & q_1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & q_2 & 0 & \ldots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \ldots & 0 & q_n & 0 & \ldots & 1 \end{pmatrix}$$

con $q = A^{-1}p$. Luego:

$P_k A^{-1}A^* = I_{n+1}$, donde P_k es una transformación elemental adecuada; además:

$$(A^*)^{-1} = P_k A^{-1}.$$

Ahora bien, dicho esquema es inestable debido a que $\|P_k\| >> 1$ y, en consecuencia, despues de cierto número de iteraciones el error de redondeo acumulado en A^{-1} puede hacerse intolerable, obligando a un "recomienzo", tal como ocurre en el método simplex de Programación Lineal. De todos modos, la de Wolfe fue la primer implementación del método que utilizaba solo $O(n^2)$ operaciones por iteración, en vez de las $O(n^3)$ que son necesarias para resolver un sistema lineal.

Barnes ([1]) dio en 1965 otra implementación del método, hasada en las siguientes fórmulas:

$$x^{k+1} = x^k - A_k^{-1}F(x^k)$$

$$A_{k+1} = A_k + u_k v_k^t$$

con

$$\begin{cases} v_k^t p_j = 0,\ j = k-1,\ldots,k-n+1, \\ u_k v_k^t p_k = F(x^{k+1}), \\ p_j = x^{j+1} - x^j. \end{cases}$$

Por lo tanto en la implementación de Barnes, se asume que el punto remplazado en el updating de la transformación afín, es el más antiguo, en tanto que en la de Wolfe, se remplazaba aquel en el cual el valor de la norma de la función era mayor . En un entorno de la solución, o en implementaciones con relajación, ambos criterios coinciden.

La inversión de la matriz A_k, realizada por variantes de la formula de Sherman-Morrison (ver [15] , 7.2.8) adolece de los mismos defectos en cuanto a acumulación de error que el "updating" de Wolfe.

Bajo hipótesis adecuadas, Ortega y Rheinboldt ([15]) prueban que el método converge localmente a un 0 de F con R-orden igual a la raíz positiva de $t^{n+1}-t^n-1 = 0$. La hipótesis más restrictiva es la imposición de que las secuencias de n incrementos consecutivos cumplan la condición de "independencia lineal uniforme" (ver [15] y [13]).

Durante varios años se prestó poca atención a los métodos derivados de la idea "secante secuencial". Esto se debió seguramente al gran éxito logrado por los métodos de tipo Quasi-Newton (ver [5] y [7] para una excelente revisión). Algunos de los raros intentos en ese sentido, deben incluir el trabajo de Schwetlik ([18]). Sin embargo, este trabajo sólo resuelve un sistema lineal completo cada cierto número de iteraciones, en vez de trabajar con modificaciones de la inversa en la línea de Wolfe.

Sólo en 1976, Cragg y Stewart dieron un método que reanima las investigaciones en esta línea. La idea del mismo se basa en la descomposición ortogonal de las matrices de los puntos x^k y de los valores de F en dichos puntos, de modo que cada iteración hace un $O(n^2)$ de operaciones, como la implementación de Wolfe, pero además el método es estable debido al uso de transformaciones ortogonales unitarias en los "updatings". Al mismo tiempo (ver [13]), la constatación de que en la implementación estable de los métodos de tipo Quasi-Newton también es conveniente usar modificación de factorizaciones por razones de estabilidad, hace desaparecer la idea de que estos métodos son esencialmente más económicos que aquéllos. Un nuevo método de tipo Quasi-Newton, el de Gay y Schnabel ([?]) es en cierto sentido, una combinación de las ideas

clásicas de Broyden con la geometría del método secante secuencial. Por esc motivo, lo incluímos en nuestro análisis.

En la sección 2 , describimos los algoritmos básicos de álgebra lineal que utilizaremos en las secciones subsiguientes. En la 3, describimos los algoritmos MSS0, MSS1, MSS2 y MSS3, los últimos tres de los cuales son "implementaciones estables del método secante secuencial", y además describimos GS1, una implementación del método de Gay-Schnabel. Las relaciones con el método de Gragg y Stewart son destacadas. También llamamos la atención sobre la manera de corregir en cada caso la eventual pérdida de independencia lineal de las matrices de incrementos, que ha sido durante mucho tiempo un notivo de descorazonamiento para el uso de este tipo de métodos. En la sección 4, los algoritmos definidos son justificados, sus propiedades demostradas, incluyendo las de orden de convergencia (que se mantienen en virtud de las salvaguardas a la independencia lineal uniforme) y su estabilidad es fundamentada. En la sección 5 se comparan los algoritmos en base a varios criterios a priori. En la sección 6 se describe una implementación globalmente converaente de los algoritmos, igualmente aplicable a otros métodos. En la sección 7 damos algunos experimentos numéricos. Finalmente, en la sección 3, puntualizamos líneas de investigación futura.

Convendremos, en la descripción de algoritmos, en utilizar el signo " = " en el sentido de "asignación" del Fortran.

2.- Modificación de factorizaciones

En esta sección describimos algunos algoritmos que serán usados en las implementaciones del método secante secuencial y métodos relacionados. Las fuentes de los mismos son los trabajos [4], [2], [3], [9], y [6].

a.- Modificación de la factorización QR por introducción de una columna a derecha.

Sean A, C, R matrices de n x n no singulares tales que

$$QA = R, \qquad (2.1.a)$$

R triangular superior, $A = (v_1,\ldots,v_n)$, $R = (r_1,\ldots r_n) = (r_{ij})$, $Q = (q_{ij})$. (2.1.b)

Si A es modificada de acuerdo a:

$$A' = (v_2,\ldots,v_n,v),\ v \in R^n,$$

se quiere obtener una factorización similar a (2.1.a) para A', y detectar el caso en que A' es singular, o casi singular. Describimos dos algoritmos a este propósito:

Algoritmo UPLU1 (Q,R,v,n,TOL)

1.- $F = \|R\|_F$.

2.- Calcular $w = Qv$ y construir $R = (r_2,\ldots,r_n,w)$.

3.- Para $j = 1,\ldots,n-1$, ejecutar los pasos 4 y 5.

4.- Comparar $|r_{jj}|$ con $|r_{j+1,j}|$ y permutar, eventualmente, las filas de R y Q de modo que $|r_{jj}| = \max\{|r_{jj}|,|r_{j+1,j}|\}$.

5.- Poner $r_{j+1,j} = 0$, y calcular

$$r_{j+1,k} = r_{j+1,k} - (r_{j+1,j}/r_{jj})r_{jk} \text{ para } k = j+1,\ldots,n;$$

$$q_{j+1,k} = q_{j+1,k} - (r_{j+1,j}/r_{jj})q_{jk} \text{ para } k = 1,\ldots,n.$$

6.- Si $|r_{nn}|/F < TOL$, declarar R casi singular.

Algoritmo UPQR1 (Q,R,v,n,TOL)

1.- Calcular $w = Qv$ y construir $R = (r_2,\ldots,r_n,w)$

2.- Para $j = 1,\ldots,n-1$, ejecutar los pasos 3 y 4.

3.- Calcular la rotación plana

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

(ver [10]) que manda $(r_{jj},r_{j+1,j})^t \longrightarrow (\rho, 0)^t$.

4.- Poner $r_{j+1,j} = 0$, $r_{jj} = \rho$, y calcular

$$r_{jk} = ar_{jk} + br_{j+1,k},$$

$$r_{j+1,k} = -br_{jk} + ar_{j+1,k}$$

para $k = j+1,\ldots,n$;

$$q_{jk} = aq_{jk} + bq_{j+1,k},$$

$$q_{j+1,k} = -bq_{jk} + aq_{j+1,k}$$

para $k = 1,\ldots,n$.

5.- Si $\|r_n\| = 0$ ó si $|r_{nn}|/\|r_n\| < TOL$, declarar R casi singular.

Observaciones.

a) Si la matriz Q es ortogonal ($QQ^t = Q^tQ = I$), el algoritmo UPQR1 conserva esa propiedad debido al uso de rotaciones, que son transformaciones ortogonales, en ese caso $|r_{nn}|$ es el módulo de la proyección de la última columna de A sobre el complemento ortogo-

nal de las columnas anteriores; luego, el test de singularidad pregunta por el seno del ángulo respectivo.

b) Numéricaménte, aún cuando A sea mal condicionada, o cuando el mal condicionamiento o inclusive la singularidad sea introducida por v, el algoritmo UPQR1 conserva la ortoqonalidad de Q.

c) El costo de UPLU1 es de $2.5n^2 + O(n)$ sumas y productos, y el de UPQR1 de $7n^2 + O(n)$ sumas y productos más el cálculo de n - 1 rotaciones.

b.- Modificación de la factorización QR por suma de una matriz uv^t.

Sean A, R y Q como en (2.1), u, v ε R^n; $A' = A + uv^t$. El algoritmo UPQR2 da una factorización similar a la de (2.1) para A', y además detecta su eventual singularidad. La eventual ortogonalidad de O se conserva.

Algoritmo UPQR2 (Q,R,u,v,n,TOL)

1.- Para j = 1,...,n-1, ejecutar los pasos 2 y 3.

2.- Calcular la rotación

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

que aplica $(u_{n-j}, u_{n-j+1})^t$ en $(\rho_j, 0)$ y escribir

$$u = (u_1,\ldots,u_{n-j+1},\rho_j,0,\ldots,0)^t.$$

3.- Calcular

$$r_{n-j,k} = ar_{n-j,k} + br_{n-j+1,k}\ ,$$

$$r_{n-j+1,k} = -br_{n-j,k} + ar_{n-j+1,k}\ ,$$

para k = n-j,...,n; y

$$q_{n-j,k} = aq_{n-j,k} + bq_{n-j+1,k}\ ,$$

$$q_{n-j+1,k} = -bq_{n-j,k} + aq_{n-j+1,k}\ ,$$

para k = 1,..., n.

4.- Sumar u_1v^t a la primer fila de R.

5.- Aplicarle los paços 3 y 4 de UPQR1 a Q y R.

6.- Si, para algún j entre 1 y n, es

$$|r_{jj}| < TOL\sqrt{\sum_{i=1}^{j} r_{ij}^2}$$

entonces declarar R casi singular.

Observnciones.

a) La eventual singularidad de $A + uv^t$ se detecta por la nnarición de ceros en la diagonal de R. La ortogonalidad de Q se conserva, aún numéricaménte, debido a que sólo se le aplican rotacio-

nes , independientemente del condicionamiento de A.

b) El costo de este algoritmo es de $13n^2 + O(n)$ sumas y productos más el cálculo de 2n - 2 rotaciones planas.

c.- Ortogonalización de un vector respecto de un conjunto de vecto-

Sean $v_1,\ldots,v_p$, v, vectores no nulos de R^n tales que $v_i^t v_j = 0$ sii $i \neq j$; $i, j = 1,\ldots,p$. Para calcular v_{p+1}, la proyección de v sobre el complemento ortogonal del subespacio generado por $v_1,\ldots, v_p$; puede utilizarse la fórmula:

$$v_{p+1} = \prod_{j=1}^{p} (I - v_j v_j^t / v_j^t v_j)\, v ,$$

reduciendo el problema a un paso del método de Gram-Schmidt modificado ([4]). Este cálculo insume $(2n + 1)p + 1$ nroductos y aproximadamente el mismo número de sumas, asumiendo que $v_j^t v_j$; $j = 1,\ldots,p$, ya han sido calculados previamente y calculando a posteriori $v_{p+1}^t v_{p+1}$. Razones de estabilidad pueden hacer recomendable una normalización previa de v y a posteriori de v_{p+1}. Llamando $A = (v_1,\ldots,v_p)$, puede pensarse que este proceso agrega una columna a A en base al siguiente algoritmo, que, por otra parte, reduce A a la única columna v- si v es dependiente de las columnas anteriores:

Algoritmo HINS1(A,v,p,n,TOL)

1.- Si p = n, saltar a 8.

2.- $\tilde{F} = \|v\|$.

3.- Para j = 1,..., p, ejecutar el paso 4.

4.- $v = v - v_j^t v v_j / v_j^t v_j$.

5.- $v_{p+1} = v$.

6.- Si $\|v_{p+1}\|/\tilde{F} <$ TOL declarar v dencndiente de $v_1,\ldots,v_p$ y saltar a 8.

7.- p = p+1 ; A = $(v_1,\ldots,v_p)$; Parar.

8.- p = 1; A = (v), Parar.

Observaciones.

Contrariamente a lo que ocurre con los métodos basados en transformaciones ortogonales, la ortogonalidad de las columnas de A se ve afectada por el condicionamiento de la matriz. Si $v_{p+1} = 0$, v es dependiente de $v_1,\ldots, v_p$. El cociente $\|v_{p+1}\|/\|v\|$ da el seno del ángulo entre v y el subespacio generado por $\{v_1,\ldots,v_p\}$.

3.- Algoritmos que implementan el método secante secuencial.

Describiremos una serie de implementaciones del método secante secuencial y un algoritmo estrechamente relacionado con éste, que implementa el método de Broyden-Gay-Schnabel ([8]). A lo largo de toda la sección, F será una función de R^n en R^n. Brevemente, llamando ΔF a los incrementos en F y Δx a los incrementos en x, el algoritmo MSS0 trabaja con una factorización de tipo LU de la matriz ΔF y con los Δx sin factorizar; MSS1 , en cambio, trabaja con factorización de tipo QR de la matriz de ΔF; MSS2 con factorizaciones QR de ambas matrices y MSS3 con factorización QR de $(\Delta F)(\Delta x)^{-1}$. Las eventuales singularidades son siempre detectadas con una tolerancia TOL > 0; que podemos considerar del orden de la precisión de la aritmética utilizada en los cálculos.

Algoritmo MSS0

Sea $x^o \in R^n$ arbitrario; $A_o = (a_1^o,\ldots,a_n^o)$, Q_o, $R_o \in R^{nxn}$ no singulares , R_o triangular superior; $v_o = (v_1^o,\ldots,v_n^o)^t \in R^n$ un vector cuyas coordenadas son las normas de las columnas de A_o; $d_o = |\det A_o|$ con $d_o \geq TOL^{n-1}\ v_1^o \ldots v_n^o$. Los pasos del algoritmo son los siguientes:

1.- $k = 0$; $F_o = F(x^o)$; si $F_o = 0$, parar.

2.- $c_k = R_k^{-1} Q_k F_k$; $c_k = (c_1^k,\ldots,c_n^k)^t$.

3.- $\delta x_k = -A_k c_k$; $x^{k+1} = x^k + \delta x_k$; $F_{k+1} = F(x^{k+1})$.

4.- Si $F_{k+1} = 0$, parar.

5.- $d'_{k+1} = d_k |c_1^k|$,

$v'_k = (v_2^k,\ldots,v_n^k,\|\delta x_k\|)^t$,

$PNOR_k = v_2^k \ldots v_n^k \|\delta x_k\|$.

6.- Si $d'_{k+1} \geq TOL^{n-1}.\ PNOR_k$, saltar a 9.

7.- (Este paso y el siguiente corrigen la singularidad)

$\Delta x_k = a_1^k \gamma_k$, con $0 < \gamma_k \leq \|\delta x_k\| / v_1^k$

8.- $\Delta F_k = F_{k+1} = F(x^{k+1} - \Delta x_k)$,

$d_{k+1} = d_k \gamma_k$, $v_{k+1} = (v_1^{k+1},\ldots,v_n^{k+1})^t = (v_2^k,\ldots,v_n^k,\|\Delta x_k\|)^t$.

Saltar a 10.

9.- $\Delta x_k = \delta x_k$,

$\Delta F_k = F_{k+1} - F_k$,

$d_{k+1} = d'_{k+1}$,

$v_{k+1} = (v_1^{k+1},\ldots,v_n^{k+1})^t = v'_k$.

10.- Ejecutar UPLU1(Q_k,R_k,ΔF_k,n,TOL), para obtener Q_{k+1} y R_{k+1}. Si R_{k+1} es casi singular, parar.

11.- $A_{k+1} = (a_1^{k+1},\ldots,a_n^{k+1}) = (a_2^k,\ldots,a_n^k,\Delta x_k)$,

$k = k + 1$; saltar a 2.

Algoritmo MSS1

Lo mismo que MSS0 salvo que la matriz original Q_0 es ortogonal y que en el paso 10 se ejecuta UPQR1 en lugar de UPLU1.

Algoritmo MSS2

Sea $x^o \in R^n$ arbitrario, Q_o, P_o, R_o, S_o matrices de n x n tales que Q_o, P_o ortogonales; R_o, S_o triangulares superiores no singulares; $R_o = (r_{ij}^o)$; $S_o = (s_{ij}^o)$;

$$|r_{jj}^o| \Big/ \sqrt{\sum_{i=1}^{j} r_{ij}^2} \geq TOL\ ,$$

$$|s_{jj}^o| \Big/ \sqrt{\sum_{i=1}^{j} s_{ij}^2} \geq TOL\ ,\ \text{para todo } j = 1,\ldots,n.$$

Los pasos del algoritmo son los siguientes:

1.- $k = 0$, $F_o = F(x^o)$; si $F_o = 0$, parar.

2.- $\delta x_k = -P_k^t S_k R_k^{-1} Q_k F_k$,

$x^{k+1} = x^k + \delta x_k$, $F_{k+1} = F(x^{k+1})$.

3.- Si $F_{k+1} = 0$, parar.

4.- Ejecutar UPQR1(P_k,S_k,Δx_k,n,TOL) para obtener P_{k+1} y S_{k+1}. Si S_{k+1} no fue declarada casi singular, saltar a [illegible]

5.- (Este paso corrige [illegible] singularidad)

$\Delta x_k = \tilde{p}_{k+1}\gamma_k$ donde $\tilde{p}_{k+1}^t$ es la última fila de P_{k+1}, $0 < \gamma_k \leq \|\delta x_k\|$ y $\Delta F_k = F_{k+1} - F(x^{k+1} - [illegible]_k)$. Remplazar la última columna de $[illegible]_{k+1}$ por $(0,\ldots,0,\gamma_k)^t$. Saltar a 7.

6.- $\Delta x_k = \delta x_k$,

$\Delta F_k = F_{k+1} - F_k$;

7.- Ejecutar UPQR1(Q_k,R_k,ΔF_k,n,TOL) para obtener Q_{k+1} y R_{k+1}. Si R_{k+1} fue declarada casi singular, parar.

V.- $k = k + 1$; saltar a 2.

Observación 3.1

Este algoritmo es similar al de Gragg-Stewart ([11]). Brevemente, las diferencias son las siguientes: a) MSS2 trabaja con incremen-

tos secuenciales ($x^k - x^{k-1}$, $x^{k-1} - x^{k-3}$,...) en tanto que el algoritmo de Gragg-Stewart lo hace con incrementos centrales ($x^k - x^{k-1}$, $x^k - x^{k-2}$,...); h) MSS2 factoriza directamente las matrices de incrementos en tanto que Gragg-Stewart trabaja con las factorizaciones de (x^k, x^{k-1},...) y de (F_k, F_{k-1},...); c) El algoritmo de Gragg-Stewart utiliza un método más costoso y que no es pasible de justificación teórica para corregir singularidades.

Algoritmo MSS3

Sea $x_0 \in R^n$ arbitrario, Q_0, P_0, R_0, S_0 como en MSS2. Los pasos del algoritmo son los siguientes:

1.- $k = 0$, $F_0 = F(x^0)$; Si $F_0 = 0$, parar.

2.- $\delta x_k = -R_k^{-1} Q_k^t F_k$, $x^{k+1} = x^k + \delta x_k$, $F_{k+1} = F(x^{k+1})$.

3.- Si $F_{k+1} = 0$, parar.

4.- Ejecutar UPQR1 ($P_k, S_k, \delta x_k, n, TOL$) para obtener P_{k+1} y S_{k+1}. Sea $\tilde{p}_{k+1}^t$ la última fila de P_{k+1}. Si $\tilde{p}_{k+1}$ no fue declarada casi singular, saltar a 6.

5.- (Corrección de singularidad)

$\Delta x_k = \tilde{p}_{k+1} \gamma_k$ con $0 < \gamma_k \leq \|\delta x_k\|$.

$\Delta F_k = F_{k+1} - F(x^{k+1} - \Delta x_k)$.

Remplazar la última columna de S_{k+1} por $(0,\ldots,0,\gamma_k)^t$.

$u_k = \Delta F_k - Q_k^t R_k \Delta x_k$.

Saltar a 7.

6.- $\Delta x_k = \delta x_k$,

$\Delta F_k = F_{k+1} - F_k$,

$u_k = F_{k+1}$.

7.- $v_k = \tilde{p}_{k+1} / \tilde{p}_{k+1}^t \Delta x_k$.

Ejecutar UPQR2 ($Q_k, R_k, u_k, v_k, n, TOL$) para obtener Q_{k+1} y R_{k+1}.

Si R_{k+1} fue declarada casi singular, parar.

8.- $k = k + 1$: saltar a 3.

Algoritmo GS1

Este es una implementación del método de Broyden-Gay-Schnabel, que incluímos a efectos comparativos con los anteriores.

Sea $x^0 \in R^n$ arbitrario, p_0 un entero $1 \leq p_0 < n$, P_0 una matriz de $n \times p_0$ cuyas columnas son perpendiculares: Q_0, R_0 como en MSS2. Los pasos del algoritmo son los siguientes:

1.- $k = 0$, $F_o = F(x^o)$, si $F_o = 0$, parar.

2.- $\Delta x_k = -R_k^{-1} Q_k F_k$, $x^{k+1} = x^k + \Delta x_k$, $F_{k+1} = F(x^{k+1})$.

3.- Si $F_{k+1} = 0$, parar.

4.- Ejecutar HINS1(P_k, Δx_k, p_k, n, TOL) para obtener P_{k+1}, p_{k+1}. Sea $\tilde{p}_{k+1}$ la columna p_{k+1} de P_{k+1}, $v_k = \tilde{p}_{k+1} / \tilde{p}_{k+1}^t \Delta x_k$.

5.- Ejecutar UPQR2(Q_k, R_k, F_{k+1}, v_k, n, TOL) para obtener Q_{k+1} y R_{k+1}; si R_{k+1} fue declarada casi singular, parar.

6.- $k = k + 1$: saltar a 2.

Observación 3.2.

En los cinco algoritmos se ha dejado abierta la elección inicial de las matrices que dan el cambio de estado. Las elecciones de éstas como matrices diagonales es natural. También es natural elegirlas de manera que la primer iteración sea un paso del Método de Newton discretizado, lo cual involucrará n evaluaciones adicionales de F y factorizaciones completas, que usan $O(n^3)$ operaciones , en esta "iteración previa". En los casos que los algoritmos paran por singularidad (en la variable dependiente) puede ser aconsejable reiniciar con una iteración del Método de Newton discretizado, para asegurarse que dicha parada se produce por una singularidad efectiva del jacobiano y no por una desafortunada disposición de los incrementos independientes. Sin embargo, la carestía de este paso, es un fuerte argumento contra él.

4.- Justificación de los algoritmos definidos.

En esta sección probaremos que los algoritmos MSS0, MSS1, MSS2 y MSS3 son efectivamente implementaciones del método secante secuencial, que las "correcciones de singularidad" - en todos los casos permiten que se satisfaga el test de independencia lineal uniforme (ILU) (ver [13] y [14]) y que, de hecho, los métodos con dichas correciones convergen bajo las hipótesis adecuadas, con R-orden igual a t, la única raiz positiva de $t^{n+1} - t^n - 1 = 0$.

Definición 4.1

Sea $\{A_\lambda\}_{\lambda \in \Omega}$ una familia de matrices no singulares de n x n; $A_\lambda = (v_1(\lambda), \ldots, v_n(\lambda))$. Diremos que esa familia satisface la propiedad de independencia lineal uniforme (ILU) para el número positivo δ, si y sólo si

$$|\det A_\lambda| \geq \delta \|v_1(\lambda)\| \ldots \|v_n(\lambda)\| \text{ para todo } \lambda \in \Omega.$$

Teorema 4.1

Sea $\{A_\lambda\}_{\lambda\in\Omega}$ como en la Definición 4.1: $S_j(\lambda) = [v_1(\lambda),\ldots,v_j(\lambda)]$ el subespacio generado por las j primeras columnas de A_λ; $\alpha_j(\lambda)$ el ángulo entre $S_j(\lambda)$ y $v_{j+1}(\lambda)$; $j = 1,\ldots,n-1$. Entonces:

a) Si $\{A_\lambda\}_{\lambda\in\Omega}$ satisface la ILU para $\delta > 0$, entonces

$$|\operatorname{sen} \alpha_j(\lambda)| \geq \delta \text{ para todo } j = 1,\ldots,n-1,\ \lambda \in \Omega.$$

b) Si $|\operatorname{sen} \alpha_j(A)| \geq \delta$ para todo $j = 1,\ldots,n-1$, $h \in R$, entonces: (4.1)

$\{A_\lambda\}_{\lambda\in\Omega}$ satisface la ILU para δ^{n-1}.

Demostración.

Se sigue de la caracterización:

$$|\det Ah| = \prod_{j=1}^{n-1} |\operatorname{sen} \alpha_j(\lambda)| \|v_j(\lambda)\| \|v_n(\lambda)\|$$

El teorema 4.1 provee la relación natural entre el clásico criterio de ILU y el test usado en los procesos de ortogonalización (cociente entre las longitudes del vector proyectado y sin proyectar). Se infiere que, para ser coherentes, cuando la tolerancia usada para el seno del ángulo es, digamos, TOL, la tolerancia correspondiente para el test del determinante debe ser del orden de TOL^{n-1}

El siguiente teorema es una generalización del 11.3.3 de [15], apta para justificar los algoritmos definidos en la sección 3.

Teorema 4.2

Sea $F: S \subset R^n \to R^n$, S abierto convexo, $F \in C^1(S)$, $F = (f_1,\ldots,f_n)^t$; $J(.)$ el jacobiano de F. (4.2)

Además, para todo x, y $\in$ S,

$$\|J(x) - J(y)\| \leq M\|x - y\| \quad (M > 0).$$

(Esto implica que

$$\|F(y) - F(x) - J(x)(y - x)\| \leq (M/2)\|Y - x\|^2,$$

ver [15].) (4.2.b)

Supongamos

$$\psi_1,\ldots,\psi_n \in S;\ \tilde{p}_1,\ldots,\tilde{p}_n \in R^n,\ \psi \in S,$$
$$\kappa_i = \psi_i + \tilde{p}_i \in S \text{ para todo } i = 1,\ldots,n;$$
$$q_i = F(\kappa_i) - F(\psi_i);\ i = 1,\ldots,n,$$

y además

$$|\det(\tilde{p}_1,\ldots,\tilde{p}_n)| / (\|\tilde{p}_1\| \ldots \|\tilde{p}_n\|) \geq \delta > 0,$$

$$B = (q_1,\ldots,q_n)(\tilde{p}_1,\ldots,\tilde{p}_n)^{-1}.$$

Entonces, existe K > 0 (que sólo depende de n y ε) tal que

$$\|J(\psi) - B\| \leq K \max\{\|\tilde{p}_j\|/2 + \|\psi_j - \psi\|\}_{j=1}^{n}.$$

Demostración.

Es análoga a la de 11.3.3 de [14].

Teorema 4.3

Supongamos que $\{x^k\}_{k=0}^{N}$, $N \leq \infty$, es engendrada por el algoritmo MSS0 o por MSS1; entonces:

a) Si $N < \infty$, entonces $F(x^{PJ}) = 0$ ó R_N es "casi singular".

b) Para todo $k = 0, 1, 2, \ldots$ $d_k = |\det A_k|$.

c) La familia $\{A_k\}_{k=0}^{N}$ cumple la ILU para $\delta = TOL^{n-1}$

d) Para todo $k \geq n$, $A_k = (\Delta x_{k-n},\ldots,\Delta x_{k-1})$.

e) Para $k \geq n$, $x^{k+1} = x^k - (\Delta x_{k-n},\ldots,\Delta x_{k-1})(\Delta F_{k-n},\ldots,\Delta F_{k-1})^{-1}F(x^k)$

f) Si F es una función afín no singular, entonces $N \leq n+1$ y $F(x^N) = 0$.

Demostración.

a) Trivial. "Casi singular" debe entenderse, respectivamente, con los criterios de UPLU1 o de UPOR1 según se trate de MSS0 o de MSS1.

b) Por inducción. Para $k = 0$, es la definición de d_0. Supongamos que $d_k = |\det A_k|$ y que $d'_{k+1} \geq TOL^{n-1}PNOR_k$. Entonces, por el paso 11,

$$\begin{aligned} A_{k+1} &= (a_2^k,\ldots,a_n^k,\Delta x_k) = \\ &= (a_2^k,\ldots,a_n^k,\delta x_k) = \\ &- (a_2^k,\ldots,a_n^k,-A_k c_k) = \\ &- (a_2^k,\ldots,a_n^k,-\textstyle\sum_i c_i^k a_i^k). \end{aligned}$$

Luego, $|\det A_{k+1}| = |c_1^k \det A_k| = |c_1^k| d_k = d'_{k+1} = d_{k+1}$,

por los pasos 3, 5 y 10 de MSS0 y MSS1.

Si $d'_{k+1} < TOL^{n-1}PNOR_k$, entonces, por los pasos 7, 8 y 11,

$$A_{k+1} = (a_2^k,\ldots,a_n^k,a_1^k\gamma_k), \text{ luego}$$

$$|\det A_{k+1}| = \gamma_k |\det A_k| = \gamma_k d_k = d_{k+1}$$

c) Por los pasos 5 y 9 y la definición de v_0, el vector v_k contiene las normas de las columnas de A_k. La familia $\{A_0\}$, compuesta por un solo miembro, cumple la ILU para TOL^{n-1} por la aserción

inicial del algoritmo. Supongamos que, para cierto k,

$$|\det A_k| \geq TOL^{n-1} v_1^k \ldots v_n^k;$$

entonces, si en el paso 6 $d'_{k+1} \geq TOL^{n-1} PNOR_k$, se sigue, por el paso 9 y el 11, que

$|\det A_{k+1}| \geq TOL^{n-1} v_1^{k+1} \ldots v_n^{k+1}$. En cambio, si $d'_{k+1} < TOL^{n-1} PNOR_k$, por los pasos 7 y 8 es:

$$\det A_{k+1} = \gamma_k \det A_k, \text{ y}$$

$$v_{k+1} = (v_2^k, \ldots, v_n^k, \gamma_k v_1^k), \text{ luego}$$

$$|\det A_{k+1}| / (v_1^{k+1} \ldots v_n^{k+1}) = |\det A_k| / (v_1^k \ldots v_n^k) \geq TOL^{n-1}.$$

d) y *e)* El algoritmo UPLU1 (respectivamente UPQR1) modifica la factorización de una matriz a la cual se le ha introducido una columna a la derecha y corrido un lugar para la izquierda las columnas 2-hasta la n. Como este algoritmo se ejecuta en todas las iteraciones y la columna introducida *se* llama siempre AP_k, *se* deduce, que para $k \geq n$,

$$Q_k(\Delta F_{k-n}, \ldots, \Delta F_{k-1}) = R_k.$$

Por similares motivos, para $k \geq n$, es

$$A_I = (\Delta r_{k-n}, \ldots, \Delta x_{k-1}),$$

por lo tanto,

$$\delta x_k = -(\Delta x_{k-n}, \ldots, \Delta x_{k-1}) R_k^{-1} Q_k F(x^k) =$$

$$= -(\Delta x_{k-n}, \ldots, \Delta x_{k-1})(\Delta F_{k-n}, \ldots, \Delta F_{k-1})^{-1} F(x^k),$$

de donde la parte e del teorema 4.3 se deduce , pues $x^{k+1} = x^k + \delta x_k$.

f) Trivial a partir de e).

Teorema 4.4

Supongamos que $\{x^k\}_{k=0}^N$, $N \leq \infty$ es engendrada por el algoritmo MSS2; cntonces:

a) Si $N < \infty$, entonces $F(x^N) = 0$ ó R_N es casi singular por el criterio del algoritmo UPQR1.

b) **Si** $A_k = P_k^t S_k$; $k = 0,1,2,\ldots,N$, entonces la familia $\{A_k\}_{k=0}^N$ cumple la ILU para $\delta = TOL^{n-1}$.

c) Para todo $k \geq n$, $A_k = (\Delta x_{k-n}, \ldots, \Delta x_{k-1})$.

d) Si, en el paso 4 de MSS2, S_{k+1} es declarada casi singular, entonces Δx_k será ortogonal a $[\Delta x_{k-n+1}, \ldots, \Delta x_{k-1}]$.

e) Para $k \geq n$,

$$x^{k+1} = x^k - (\Delta x_{k-n},\ldots,\Delta x_{k-1})(\Delta F_{k-n},\ldots,\Delta F_{k-1})^{-1} F(x^k)$$

f) Si F es afín no singular, $N \leq n + 1$ y $F(x^N) = 0$.

Demostración

a) Igual que el Teorema 4.3.

b) Probaremos (4.1) con k remplazando λ y $R = \{0, 1,\ldots,N\}$. Como P_k es ortogonal, conserva ángulos y longitudes, por lo tanto, es lo mismo probar (4.1) para A_k como para S_k. S_o verifica (4.1) por la aserción inicial de MSS2. Supongamos que S_k verifica (4.1). En el algoritmo UPQR1 (paso 4) , las columnas 2 hasta n de S_k son multiplicadas por matrices ortogonales (rotaciones). Luego los ángulos entre esas columnas y los subespacios generados por las mismas no se alteran. Por lo tanto sólo es necesario testear la última columna de S_{k+1} (como efectivamente lo hace UPQR1) para ver si (4.1) se cumple en esta matriz. Luego, si S_{k+1} no fue declarada casi singular por UPQR1, efectivamente continúa cumpliéndose (4.1). Si fue declarada singular, la modificación introducida por el paso 5 garantiza que $|\text{sen } \alpha_{n-1}(k)| = 1$, por lo tanto (4.1) se cumple para A_{k+1}.

c) Probemos que para todo $k \geq 0$, A_{k+1} es una matriz formada por las columnas 2 hasta la n de A_k, seguidas de Δx_k (De aquí se sigue fácilmente c) por inducción).

Ahora, $A_k = P_k^t S_k$, luego $P_k A_k = S_k$; entonces en el paso 4, por definición del algoritmo UPQR1, P_{k+1} y S_{k+1} se encuentran de manera que $P_{k+1} A_{k+1} = S_{k+1}$ donde A_{k+1} es una matriz formada de las columnas 2 hasta la n de A_k seguidas de δx_k. Luego si S_{k+1} no es declarada casi singular, por el paso 6 , es $\Delta x_k = \delta x_k$, como queríamos probar.

Si, en cambio, S_{k+1} es declarada casi singular en el paso 4, su última columna es remplazada por $(0,\ldots,0,\gamma_k)^t$ (paso 5); pero $\gamma_k \tilde{p}_k$ es justamente el nuevo Δx_k en este caso, donde $\tilde{p}_k$ es la última columna de P_{k+1}^t; y por lo tanto Δx_k es también la última columna de A_k.

d) Por ser S_{k+1} triangular superior, $\Delta x_{k-n+1},\ldots,\Delta x_{k-1}$ es el subespacio generado por las n - 1 primeras columnas de P_{k+1}^t, luego, como P_{k+1} es ortogonal, Δx_k es ortogonal a $\Delta x_{k-n+1},\ldots,\Delta x_{k-1}$ cuando es tomado como la última columna de P_{k+1}^t.

e) $Q_k(\Delta F_{k-n},\ldots,\Delta F_{k-1}) = R_k$ por los mismos motivos expuestos en el Teorema 4.3 (parte d). Entonces, el resultado se sigue del

paso 2 del algoritmo y de las partes b) y c) del presente teorema.

f) Igual que en el Teorema 4.3.

Teorema 4.5

Si $\{x^k\}_{k=0}^{N}$ $N \leq \infty$ es engendrada por el algoritmo MSS3, entonces:

a) Si $N < \infty$, es $F(x^N) = 0$ ó P_N es casi singular por el criterio de UPQR2.

b) Si $A_k = P_k^t S_k$, $k = 0,1,\ldots,N$, entonces la familia $\{A_k\}_{k=0}^{N}$ cumple la ILU para $\delta = TOL^{n-1}$.

c) Para todo $k \geq n$, $A_k = (\Delta x_{k-n},\ldots,\Delta x_{k-1})$.

d) Si, en el paso 4 de MSS3, S_{k+1} es declarada casi singular, entonces, Δx_k será ortogonal a $[\Delta x_{k-n+1},\ldots,\Delta x_{k-1}]$.

e) Para $k \geq n$, $x^{k+1} = x^k - (\Delta x_{k-n},\ldots,\Delta x_{k-1})(\Delta F_{k-n},\ldots,\Delta F_{k-1})^{-1} F(x^k)$

f) Si F es afín no singular, $N \leq n + 1$ y $F(x^N) = 0$.

Demostración

Sólo hay que probar e), pues las demás partes se prueban igual que en el Teorema 4.4.

Definimos $B_{k+1} = Q_{k+1}^t R_{k+1}$, $k = 0, 1, 2,\ldots$ Por construcción,

$B_{k+1} = B_k + F_{k+1} p_{k+1}^t / p_{k+1}^t x_k$, donde p_{k+1}^t es la fila n de P_{k+1}. Entonces,

$$B_{k+1} x_k = B_k x_k + F_{k+1}.$$

Ahora, si en el paso 6, S_{k+1} no fue declarada casi singular,

$$\Delta x_k = \delta x_k = -B_k^{-1} F_k, \text{ luego}$$

$$F_{k+1} = \Delta F_k - B_k \Delta x_k, \text{ por lo tanto}$$

$$B_{k+1} \Delta x_k = B_k \Delta x_k + \Delta F_k - B_k \Delta x_k, \text{ lo que sigue}$$

siendo válido si S_{k+1} fue declarada casi singular, debido a la definición de u_k en el paso 5 de MSS3. Entonces,

$$B_{k+1} \Delta x_k = \Delta F_k.$$

Pero $\tilde{p}_{k+1}$ es ortogonal a las primeras n-1 columnas de P_{k+1}^t y, por lo tanto a las primeras n-1 columnas de A_{k+1}. Luego, la tesis se skgue fácilménte de la parte c) de este teorema.

Teorema 4.6

Si $\{x^k\}_{k=0}^{N}$, $N \leq \infty$ es engendrada por el algoritmo GS1, entonces,

a) Si $N < \infty$, es $F(x^N) = 0$ ó R_N casi singular según UPOR2.

b) Si F es afín no singular, $N \leq n + 1$ y $F(x^N) = 0$.

Demostración

a) es trivial. Ver [R] para la prueba de b).

Teorema 4.7

Supongamos $F: S \subset R^n \to R^n$ cumpliendo (4.2) y además $x^* \in S$ con $F(x^*) = 0$, $J(x^*)$ no singular. (4.3)

Sea $\{x^k\}_{k=0}^N$ engendrada por MSS0, MSS1, MSS2 ó MSS3. Entonces existe $\delta > 0$ tal que si $\|x^m - x^*\| \leq \delta$, con $m \geq n$, entonces o bien $x^N = x^*$ (si $N < \infty$), ó $\lim x^k = x^*$ cuando $k \to \infty$. Además, en este caso, el R-orden de convergencia de la sucesión es la raíz positiva de $t^{n+1} - t^n - 1 = 0$.

Demostración

Sea $k \geq n$. Por el Teorema 4.3 (c, d y e), 4.4 (b, c y e) y 4.5 (h, c y e), las hipótesis del teorema 4.2 se cumplen, poniendo:

$$\psi = x^k$$

$$\psi_j = x^{k-j+1} - x^{k-j},\ j = 1,\ldots,n$$

$$\tilde{p} = \Delta x_{k-j},\ j = 1,\ldots,n$$

$$\delta = TOL^{n-1}.$$

En consecuencia, llamando

$$B_k = (\Delta F_{k-n},\ldots,\Delta F_{k-1})(\Delta x_{k-n},\ldots,\Delta x_{k-1})^{-1},$$

resulta

$\|B_k - J(x^k)\| \leq K_1 \max\{\|x_{k-j} - x^*\|\}_{j=1}^n$, siguiéndose la tesis de la sección 11.2 y del teorema 9.2.9 de [15].

Teorema 4.8

Supongamos (4.2), (4.3) y $\{x^k\}_{k=0}^N$ engendrada por GS1. Llamemos $B_k = Q_k^t R_k$. Entonces, existen $\varepsilon, \delta > 0$ tales que si $\|x^0 - x^*\| \leq \varepsilon$ y $\|B_0 - J(x^*)\| \leq \delta$, entonces, $F(x^N) = 0$ ó x^k converge Q-superlinealmente a x^* y el conjunto $\{\|B_k\|, \|B_k^{-1}\|\}_{k=0}^N$ está acotado.

Demostración

Ver [8], teorema 4.5.

Teorema 4.9

En las condiciones del Teorema 4.8, si la sucesión p_k de enteros en GS1 tiene la forma, para algún $m \geq 0$;

$p_{m+nj+i} = i$ para todo $j = 0,1,2\ldots$; $i = 1,\ldots,n$

(o sea, el algoritmo HINS1 siempre logra completar n incrementos independientes), entonces x^k converge a x^* con un R-orden mayor o igual que la raíz positiva de $t^{2n} - t^{2n-1} - 1 = 0$.

Demostración

Ver [12].

Razones de estabilidad

Brevemente, las alternativas introducidas para la implementación del método secante secuencial, son las siguientes:

a) En MSS0: Factorización de $(\Delta F_{k-n},\ldots,\Delta F_{k-1})$ usando transformaciones elementales.

b) En MSS1: Lo mismo usando transformaciones ortogonales.

c) En MSS2: Lo mismo que en MSS1 más la factorización **de** $(\Delta x_{k-n},\ldots,\Delta x_{k-1})$ **usando transformaciones ortogonales.**

d) En MSS3: Factorización **de** $(\Delta F_{k-n},\ldots,\Delta F_{k-1})(\Delta x_{k-n},\ldots,\Delta x_{k-1})^{-1}$ usando transformaciones ortogonales (la factorización de $A_{k-n},\ldots,\Delta x_{k-1}$) **no** se usa en **los** cálculos, **sólo** en la detección de singularidades).

La implementación clásica del método secante secuencial se basaba en la conservación y updating de

$\tilde{H}_k = ((\Delta F_{k-n},\ldots,\Delta F_{k-1})(\Delta x_{k-n},\ldots,\Delta x_{k-1})^{-1})^{-1}$ usando variantes de la fórmula de Sherman-Morrison (ver [14], 7.2.8). Esta forma de generación es inestable, debido a que $\tilde{H}_k$ se ve sucesivamente sometido a una serie de multiplicaciones a derecha por matrices no siempre bien condicionadas que suelen hacer tender su norma peligrosamente a infinito. En cambio, MSS3 y GS1 basan su esquema en la factorización de la aproximación B_k del jacobiano, usando premultiplicación por matrices ortogonales, que conservan el número de condición de las multiplicadas y por lo tanto evitan la tendencia a overflow proveniente del mal condicionamiento.

MSS0, MSS1 y MSS2 factorizan por separado las matrices de los ΔF respecto del tratamiento de los Δx. Si suponemos, como en [11], que:

$$Q_k((\Delta F_{k-n},\ldots,\Delta F_{k-1}) + H_k) = R_k,$$

entonces, se obtienen cotas para $H_k = (h_{k-n},\ldots,h_{k-1})$ de la forma

$$\|h_j\| \leq \|Q_j\|\|Q_j^{-1}\|\|\Delta F_j\| n^{3/2}\varepsilon ,$$

donde ε depende de la precisión de la máquina. De aquí se deriva la

mayor estabilidad de MSS1 y MSS2 respecto de MSS0 pues en los primeros Q_k es ortogonal, luego $\|Q_k\| = 1$, en tanto en MSS0 la norma de Q_k crece con k y puede llegar a hacerse intolerablemente grande. Por esa razón MSS0 no será considerada una implementación estable del método secante secuencial.

En definitiva, en MSS1 y MSS2 , es

$$\| h_{k-1} \| \leq 2n^{3/2} \| \Delta F_{k-1} \| \varepsilon$$

en tanto para las restantes columnas de H, el mismo tipo de cota puede ser obtenida porque h_{k-1} sufre sólo de un proceso de aplicación de rotaciones a medida que el proceso avanza, que no aumentarán decisivamente su norma. Finalmente , al cabo de n pasos, la columna es definitivamente descartada, de modo que su error habrá dejado en absoluto de influenciar.

El nisno análisis cabe para P_k, S_k en MSS2 y MSS3.

Trabajo computacional

En una iteración sin corrección de singularidad cada algoritmo usa el número de operaciones que sigue (sumas y multiplicaciones), además de una evaluación de función:

MSS0: $5n^2 + O(n)$ operaciones.

MSS1: $9.5n^2 + O(n)$ operaciones y n - 1 rotaciones.

MSS2: $17n^2 + O(n)$ operaciones y 2n - 2 rotaciones.

MSS3: $21.5n^2 + O(n)$ operaciones y 3n - 3 rotaciones.

GS1: (promedio) $15.5n^2 + O(n)$ operaciones y 2n - 3 rotaciones.

Evaluación comparada

En base a distintos criterios, compararemos los algoritmos presentados.

a) Corrección de singularidades: MSS2 y MSS3 corrigen la singularidad mejor que MSS0 y MSS1 pues, en caso de rechazo de un incremento, permiten que el corre-ido sea ortogonal a los anteriores en tanto MSS0 y MSS1 sólo garantizan que la nueva matriz de incrementos tenga un condicionamiento similar a la de la última na rechazada. GS1 no corrige la degeneración sino que, cuando esta aparece, modifica la aproximación del jacobiano con la formula clásica de Broyden. Esto hace difícil compararlo con los anteriores a este respecto. Como hemos ya dicbo , el método de Gragg-Stewart corrige la de-

generación por un medio que, probablemente, logra un condicionamiento de la matriz de los Ax significativamente mejor que el anterior pero que es caro y no puede justificarse teóricamente.

b) Costo com-utacional: De acuerdo a los cálculos de la sección anterior, en orden creciente de costos , los métodos se ubican así:

1) MSS0; 2) MSS1; 3) GS1; 4) MSS2; 5) MSS3.

Sin embargo , recordemos que MSS0 no es un método estable.

c) Aprovechamiento de la información anterior: Si R_k es una buena aproximación del jacobiano en x^k, este hecho es explotado sólo por GS1, pues de acuerdo a [8], el updating de B_k se hace tomando B_{k+1} como la matriz más cercana a B_k entre las que verifican: $B_{k+1}\Delta x_k = \Delta F_k, \ldots, B_{k+1}\Delta x_{k-p+1} = \Delta F_{k-p+1}$. Esta diferencia se hace sentir cuando, por ejemplo, el método comienza tomando B_o como una discretización del jabobiano; o cuando se hacen ese tino de recomienzos por razones de singularidad de los incrementos dependientes.

d) Esparcimiento: Si existe alguna información sobre entradas del jacobiano de F (por ejemplo, que gran parte de ellas son ceros); dicha información puede ser aprovechada por MSS3 y nor GS1, pues estos métodos trabajan con aproximaciones directas del jacobiano: pero no por los otros métodos, que separan los incrementos independientes de los dependientes.

E) Implementaciones con relajación curvilínea: Si B_k es una aproximación del jacobiano de F en x^k, entonces $2B_k^t F(x^k)$ es una aproximación del gradiente de $\|F\|^2$ en x^k. Esta aproximación es utilizada exitosamente por métodos de tino Quasi-Newton para generar un camino que, comenzando por $x^k - B_k^{-1}F(x^k)$ acaba en x^k en una curva tangente a la dirección $-B_k^t F(x^k)$ (ver [13]). Los métodos MSS2, MSS3 y GS1 aquí expuestos son susceptibles de ese tino de implementación, sin significativo trahajo adicional. En cambio MSS0 y MSS1 necesitarían de $O(n^3)$ operaciones adicionales nara generar una aproximación a la dirección del gradiente (la inversión de la matriz de los Δx , que en los otros métodos es dada factorizada).

f) Memoria: El requerimiento de memoria en arreglos de los distintos métodos es el siguiente:

MSS0: $2.5n^2 + O(n)$ posiciones.

MSS1: $2.5n^2 + O(n)$ posiciones.

GS1: $2.5n^2 + O(n)$ posiciones.

MSS2: $3n^2 + O(n)$ posiciones.

MSS3: $3n^2 + O(n)$ posiciones.

q) Velocidad de convergencia: Mientras MSS0-3 tienen R-orden de convergencia igual a la raíz positiva de $t^{n+1} - t^n - 1 = 0$, con un trabajo de una evaluación funcional por iteración (o dos en los casos de degeneración), el método GS1 tiene R-orden al menos la raíz positiva de $t^{2n} - t^{2n-1} - 1 = 0$ (menor que el anterior) con una hipótesis reçtrictiva. Esto parece reflejar el hecho de que las implementaciones del método secante secuencial tratan de reflejar con más fidelidad la historia anterior del proceso iterativo que el método de Gay-Schnabel, lo mismo que explicaría la superioridad del método de Gay-Schnabel respecto al clásico de Broyden (ver [8]). No obstante, tal vez el resultado de orden del método de Gay -.Schnabel pueda ser mejorado.

5.- Convergencia global

Existen muchas maneras como un algoritmo de àlto orden de convergencia local puede dar origen a un algoritmo globalmente convergente (ver [16] y [17]). En esta sección se describe una de ellas.

Algoritmo GLOB

Sea α tal que $0 < \alpha < 1$; m un entero mayor o igual que 1; $K>0$

$h: (0, \varepsilon_1) \Rightarrow (0, \varepsilon_2)$ tal que $h(\varepsilon) \neq 0$ para todo ε,

$$\lim_{\varepsilon \to 0} h(\varepsilon) = 0$$

$$\lim_{\varepsilon \to 0} \varepsilon / h(\varepsilon) = 0 ,$$

$x^k \in R^n$, $F_k = F(x^k)$, $K > 0$.

Los pasos del algoritmo son:

1.- Si $F_k = 0$, parar;

$\lambda_k = 2(1 - \alpha)$

2.- Si k es múltiplo de n, saltar a 4.

3.- Utilizar un algoritmo A (por ejemplo MSS0-3 ó GS1 con algún método de relajación como interpolación parabólica p/ej) para obtener un punto x^{k+1} tal que:

$$\|F(x^{k+1})\|^2 \leq \|F(x^k)\|^2$$

$$\|x^{k+1} - x^k\| \leq K\|F(x^k)\| .$$

k = k + 1; Saltar a 1.

4.- Si $\|F_k\|^2 < \alpha \|F_{k-m}\|^2$, saltar a 3.

5.- Iteración especial.

Sean $\{v_1,\ldots, v_p\}$un conjunto de vectores tales que

$\|v_j\| \leq K\|F_k\|$ para todo $j = 1,\ldots, p$, $p \geq n$, entre los cuales hay n de ellos $\{w_1,\ldots,w_n\} \subset \{v_1,\ldots,v_p\}$tales que

$$|\det (w_1,\ldots,w_n)| \geq d > 0.$$

Ejecutar los pasos 5.1 hasta 5.4.

5.1.- $\varepsilon = 1 - c$.

5.2.- Para $j = 1,\ldots,p$, ejecutar 5.2.1 y 5.2.2.

5.2.1.- Si $\|F(x^k + h(\varepsilon)v_j\|^2 < (1 - \varepsilon)\|F_k\|^2$, saltar a 5.4

5.2.2.- Si $\|F(x^k - h(\varepsilon)v_j\|^2 < (1 - \varepsilon)\|F_k\|^2$, saltar a 5.4.

5.3.- $\lambda_k = \varepsilon$; $k = k + 1$; $\varepsilon = \varepsilon/2$. Saltar a 5.2.

5.4.- $\lambda_k = 2\varepsilon$

$x^{k+1} = x^k \pm h(\varepsilon)v_j$; (el $\pm$ determinado por 5.2.1 ó 5.2.2).

$F_{k+1} = F(x^k \pm h(\varepsilon)v_j)$,

$k = k+1$

Saltar a 1.

Teorema 6.I-

Sea $\{x^k\}_{k\in K_1}$ una subsucesión de una sucesión generada por GLOB; convergente a un punto $x \in R^n$. Entonces:

a) $J^t(x)F(x) = (1/2)\nabla\|F(x)\|^2 = 0$.

b) Si $\limsup \lambda_k \neq 0$ para $k \in K_1$; entonces $F(x) = 0$, y además $\lim_{k\to\infty} F(x^k) = 0$.

c) Si, en el caso h), x es un cero aislado de F, entonces $\lim_{k\to\infty} x^k = x$.

Demostración.

a) Supongamos que $\lim h_k = \bar{h} \neq 0$ para $h \in K_1$. Entonces , existe K_2 un subconjunto infinito de K_1 tal que; llamando $f(x) = \|F(x)\|^2$:

$$f(x^k + h(\lambda_k)w_j^k) \geq (1 - \lambda_k)f(x^k) \leq f(x^k - h(\lambda_k)w_j^k)$$

para todo $j = 1,\ldots,n$; $k \in K_2$. Por lo tanto:

$$(f(x^k + h(\lambda_k)w_j^k) - f(x^k))/h(\lambda_k) \geq (-\lambda_k/h(\lambda_k))f(x^k)$$

$$(f(x^k - h(\lambda_k)w_j^k) - f(x^k))/h(\lambda_k) \geq (-\lambda_k/h(\lambda_k))f(x^k)$$

para todo $k \in K_2$, $j = 1,\ldots,n$.

Luego, por el teorema de valor medio, existen θ_k, θ'_k entre 0 y 1, para $k \in K_2$, tales que:

$$\langle w_j^k \ , \ f(x^k + \theta_k h(\lambda_k) w_j^k \rangle \geq (\lambda_k / h(\lambda_k)) f(x^k) \qquad (6.1)$$

$$\langle w_j^k \ , \ f(x^k - \theta_k' h(\lambda_k) w_j^k \rangle \geq (-\lambda_k / h(\lambda_k)) f(x^k) \qquad (6.2)$$

Ahora, $\{x^k\}_{k \in K_1}$ es convergente, en consecuencia acotado: luego, por continuidad, existe $M > 0$ tal que

$$f(x^k) \leq M \text{ para todo } k \in K_1.$$

Por lo tanto

$$\| w_j^k \| \leq KM \text{ y } |\det(w_1^k, \ldots, w_n^k)| \geq d > 0 \text{ para todo } k \in K_1,$$

$j = 1, \ldots, n$; luego, por compacidad, se puede extraer $K_3 \subset K_2$, K_3 infinito tal que:

$(w_1^k, \ldots, w_n^k) \longrightarrow (w_1, \ldots, w_n)$ para $k \in K_3$; $w_1, \ldots, w_n$ linealmente independientes. Entonces, pasando al limite en (6.1) y (6.2), nara $k \in K_3$:

$$\langle w_j \ , \ f(x) \rangle = 0$$

para todo $j = 1, \ldots, n$; luego $f(x) = 0$.

La parte a) del teorema para el caso lim sup $\lambda_k > 0$ se deriva de la parte b), pues $f(x) = 0$ implica en este caso $\nabla f(x) = 0$. Pasemos a probar entonces la parte b).

Dicha prueba es directa. Si lim sup $\lambda_k > 0$ nara $k \in K_1$, entonces existe $K_2 \subset K_1$, K_3 infinito: m_k enteros mayores o iguales que 1 para $k \in K_2$ tales que:

$$f(x^k) < (1 - \tfrac{1}{2}\lambda_k) f(x^{k-m_k}) \quad \text{nara todo } k \in K_2,$$

expresión de la cual se deduce:

$$\lim f(x^k) = 0 \text{ para } k \in K_2;$$

luego $f(x) = 0$.

c) El algoritmo GLOB cumple la siguiente propiedad:
Para todo $\varepsilon > 0$, existe $\delta > 0$ tal que $\|F(x^k)\| < \delta$ implica $\|x^{k+1} - x^k\| < \varepsilon$
En consecuencia, se le pueden aplicar los teoremas de estabilidad de [14] obteniéndose c).

6.- Experimentos numéricos

a) Versiones sin relajación:

Se probaron los algoritmos MSS1, MSS2, MSS3 y GS1 con funciones test generadas de la siguiente manera:

$$F_i(x) = \sum_{j=1}^{n} A_{ij} \operatorname{sen} x_j + B_{ij} \cos x_j - E_i + 30 x_i$$

$i = 1,\ldots,n$; donde A_{ij} y C_{ij} son números aleatorios entre -100 y 100; la solución del problema, x^* es generada al azar entre $-\pi$ y π, y :

$$E_i = \sum_{j=1}^{n} A_{ij} \operatorname{sen} x_7^* + B_{ij} \cos x_3^* + 30x_i^*.$$

Estas funciones son análogas a las trigonométricas de Fletcher-Powell; la principal diferencia es la adición del término $30x$ que agregamos para evitar convergencia a soluc ones distintas de la prevista (hecho que, de ocurrir, es difícil de evaluar en términas de éxito o fracaso). Son funciones bastantes representativas de "casos reales" debido a sus variacioncs de pendiente, curvaturas, amplitudes de onda, etc. Por otra parte, no exhiben ninguna patología digna de mención, genéricaménte hablando.

En todos los algoritmos se tomó el primer paso como una iteración del método de Newton discretizado (determinado por la elección inicial de las matrices); y se previó volver a este paso toda vez que se registrara una singularidad de los AF (lo que nunca ocurrió en esta tanda de tests). El criterio de parada es $\| F(x) \| \leq 10^{-4}$, y se permitieron *30* iteraciones$_i$, rotulándose como "divergencia" los casos en que este número de iteraciones no fue suficiente para alcanzar convergencia. Debido a la elección del primer paso, el número de evaluaciones de función es siempre n + 1 + número de iteraciones, donde n es la dimensión del problema. Los resultados están clasificados en dos grupos, de acuerdo a la elección del punto inicial. Bajo Ia sigla del método escribiremos"C,j"en caso de que el método en cuestión haya convergido en j iteracioneç; y D en caso de divergencia.

Grupo 1:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$, con ε_i aleatorio (-0.01,0.01).

n	Prob.Nº	MSS1	MSS2	MSS3	GS1
5	1	C,4	C,4	C,4	C,4
5	2	C,3	C,3	C,3	C,3
5	3	C,4	C,4	C,3	C,3
10	1	C,5	C,5	C,5	C,4
10	2	C,3	C,3	C,3	C,3
10	3	C,3	C,3	C,4	C,3
20	1	C,6	C,6	C,5	C,5
20	2	C,4	C,4	C,4	C,4
20	3	C,7	C,7	C,4	C,4

Grupo 2:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$, con ε_i aleatorio (-0.1 , 0.1).

n	Prob.Nº	MSS1	MSS2	MSS3	GS1
5	1	C,11	C,11	C,15	C,9
5	2	C,15	C,15	C,20	C,15
5	3	C,7	C,7	C,7	C,6
10	1	C,12	C,12	C,10	C,9
10	2	C,6	C,6	C,5	C,5
10	3	C,8	C,8	C,10	C,7
20	1	C,28	C,29	C,22	C,17
20	2	C,8	C,8	C,16	C,7
20	3	D	D	D	C,15

Obsérvese que la perfomance de GS1 fue superior a la de los demás. Tal superioridad es atribuíble a la elección del paso inicial. En efecto, en la segunda iteración la aproximación que utiliza GS1 del jacobiano es la más cercana a B_o entre todas las que cumplen:

$$B_1(x^1 - x^o) = F(x^1) - F(x^o);$$

luego, en este paso, y tal vez en algunos de los siguientes, B_1 puede ser una representación más adecuada de $J(x^1)$ que las utilizadas por los métodos secantes secuenciales, ya que, en todos los casos R_o es una aproximación discretizada de $J(x^o)$.

b) Versiones con relajación:

Se probaron los algoritmos MSS11 y GS11 con las mismas funciones test que en a).

MSS11 (respectivamente GS11) es una versión con relajación de MS31 (respectivamente GS1). La novedad consiste en lo siguiente: una vez computado δx_k ; se testea si $\|F(x^k + \delta x_k)\|^2 \leq \|F(x^k)\|^2$; si esto no es así, se somete δx_k a un proceso de interpolación parabólica para lograr esa condición. Si al cabo de 8 relajaciones aún no se logró el descenso, se remplaza δx_k por los distintos ejes coordenados y se relaja en los mismos con el propósito de descender. En este caso, lógrese el descenso o no, se reinicia el algoritmo con un paso del Newton discretizado. Escribiremos, bajo la sigla del método; C,j,k para indicar que el método convergió en j iteraciones con k evaluaciones de *i*, y E en *caso* de que , tras (n + 1)100 evaluaciones de función, no se haya logrado convergencia.

Los resultados están clasificadoç en 5 grupos de acuerdo a la elección del punto inicial.

Grupo 1:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$. con E_i aleatorio entre -0.1 y 0.1

n	Prob.Nro.	MSS11	GS11
5	1	C,11,17	C,9,15
5	2	C,19,86	C,23,103
5	3	C,7,13	C,6,12
10	1	C,9,22	C,9,20
10	2	C,6,17	C,5,16
10	3	C,8,19	C,7,18
20	1	C,18,48	C,13,38
20	2	C,8,29	C,7,28
20	3	C,32,123	C,23,107

Grupo 2:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$ con ε_i aleatorio entre -0.3 y 0.3

n	Prob.Nro.	MSS11	GS11
5	1	C,39,127	E
5	2	C,12,21	C,15,35
5	3	C,10,20	C,10,19
10	1	C,22,55	C,13,26
10	2	C,9,23	C,8,19
10	3	C,15,33	C,15,31
20	1	C,41,162	C,26,72
20	2	C,17,50	C,12,36
20	3	C,127,495	C,111,747

Grupo 3:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$ con ε_i aleatorio entre -0.5 y0.

n	Prob.Nro.	MSS11	GS11
5	1	E	E
5	2	C,16,33	C,35,152
5	3	C,21,54	C,12,20
10	1	C,33,89	C,29,104
10	2	C,14,39	C,10,21
10	3	C,49,205	C,31,79
20	1	C,168,840	C,84,510
20	2	C,31,95	C,13,47
20	3	E	E

Grupo 4:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$ con ε_i aleatorio (-0.7 , 0,7)

n	Prob.Nro.	MSS11	GS11
5	1	E	E
5	2	C,62,335	C,25,95
5	3	C,42,146	E
10	1	C,31,139	C,84,313
10	2	C,20,39	C,17,39
10	3	C,47,145	C,44,133
20	1	C,124,534	E
20	2	E	E
20	3	C,191,907	C,174,1076

Grupo 5:

Punto inicial: $x_i^o = x_i^*(1 + \varepsilon_i)$ con ε_i aleatorio (-0.9 , 0.9)

n	Prob.Nro.	MSS11	GS11
5	1	E	E
5	2	C,29,76	C,27,82
5	3	C,44,234	C,26,77
10	1	E	E
10	2	C,47,168	C,39,123
10	3	E	C,83,407
20	1	E	C,126,497
20	2	C,65,361	E
20	3	E	E

Cuando aumenta la distancia del punto inicial respecto de la solución, la ventaja de conservar aproximaciones parecidas a la inicial de GS1 pierde parte de su significación. En los nuevos tests GS11 divergió en 11 ocasiones y MSS11 en 9, todas en casos en que la aproximación inicial tenía un desvío del 30% o más respecto de la solución. No obstante , algo de esa ventaja se sigue manteniendo debido a los recomienzos ocasionales. Como balance, el tiempo de CPU gastado por MSS11 en la totalidad de los tests fue un 94% del utilizado por GS11, lo que de todos modos no es muy significativo debido a que el programa de MSS11 estaba recargado con muchos más controles de impresión que el de GS11.

7.- Consideraciones finales

En este trabajo nos propusimos llamar la atención sobre el método secante secuencial para resolver sistemas no lineales sin derivadas y mostrar que, probablemente, implementaciones inteligentes de este método lo hacen asaz competitivo con los harto estudiados Quasi-Newton. En particular, las dificultades tradicionales apuntadas de este método pueden ser eficientemente superadas con recursos inherentes a los algoritmos mismos. En el futuro será necesario desarrollar un gran esfuerzo experimental con el fin de evaluar la real ubicación de este tipo de métodos entre los que se proponen fines similares. En particular: a) Se debe variar la elección inicial de las matrices; b) Se deben implementar métodos eficientes de relajación, e incorporar las salvaguardas eficaces de convergencia; c) Se debe trabajar en implementaciones que aprovechen estructuras particulares de la matriz jacobiana (esparciniento); d) Se debe trabajar en la extensión de la idea "secante secuencial" a cuadrados mínimos no lineales.

Referencias

1.- J.G.P. Barnes (1965), An algorithm for solving nonlinear equations based on the secant method, Comput.J., R, pp 66-72.

2.- R.H. Bartels y G.H. Golub: The simplex method of linear programming using LU decomposition, Comm. ACM, 12, 266-268, 1969.

3.- R.H. Bartels, G.H. Golub y M.A. Saunders (1970): Numerical techniques in Mathematical Programming, en Nonlinear Programming, J.B. Rosen, O.~.Mangasarian y R, Ritter (editores), Academic Press Inc, >E,

4.- A. Bjorck (1967): Solving least-squares problems by Gram-Schmidt orthogonalization, BIT, 7, pp 1-21.

5.- C.G. Broyden (1967): Quasi-Newton methods and their application to function minimization, Math.Comp., 21, pp 368-381.

6.- J.W. Daniel, W.B. Gragg, L. Kaufman y G.W. Stewart (1976): Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, Math.Comp., 30, pp 772-795.

7.- J.E. Dennis y J.J. Moré (1974): Quasi-Newton methods: Motivation and theory, TR 74-217, Dept. Com-. Sci, Cornell University.

8.- D.M. Gay y R.B. Schnabel (1977): Solving systems of nonlinear equations by Broyden's method with projected updates, Working paper Nº 169, National Bureau of Economic Research.

9.- P.E. Gill, W. Murray, G.H. Golub y M.A. Saunders (1974): Methods for modifying matrix factorizations, Math.Comp., 28, pp 505-535.

10.- J.W. Givens (1958): Computation of plane unitary rotations transforming a general matrix to triangular form, J.Soc.Ind.Appl.Math., 6, pp 26-50.

11.- W.B. Gragg y G.W. Stewart (1976): A stable variant of the secant method for solving nonlinear equations, SIAM J. Numer. Anal., 13, pp

12.- J.M. Martínez (1977): Esta tesiç, Capítulo II

13.- J.J. Moré y J. Tragenstein(1974): On the global convergence of Broyden's method, Math.Comp., 30, pp 523-540.

14.- J.M. Ortega (1972): Stability of difference equations and convergence of iterative processes, TR 191, Comp.Sci. Center, University of Maryland.

15.- J.M. Ortega y W.C. Rheinboldt (1970): Iterative solution of nonlinear equations in several variables, Academic Press, N.Y..

16.- Polak, E. (1971): Computational methods in optimization. A Unified Approach, Academic Press, N.Y..

17.- Polak, E. (1974): A globally converging secant method with applications to boundary value problems, SIAM J. Numer. Anal., 11, pp 529-537.

18.- H, Schwetlik (1970), Algorithm 12: a discrete method for the solution of finite dimensional systems of nonlinear equations, Comp., 5, pp. 82-08,

19.- P. Wolfe (1959); The secant method for solving nonlinear equations, Com, A C M, 2, pp. 12-13,