

# Resolución Numérica de Sistemas No Lineales de Gran Escala

José Mario Martínez

[www.ime.unicamp.br/~martinez](http://www.ime.unicamp.br/~martinez)

Departamento de Matemática Aplicada, Universidad Estadual de Campinas, Brasil

2012

## Definición y Notación

Se trata de resolver  $F(x) = 0$ , donde  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ .

Denotaremos

$$F(x) = (f_1(x), \dots, f_n(x))^t.$$

$$J(x) \equiv F'(x) = (\nabla f_1(x), \dots, \nabla f_n(x))^t \in \mathbb{R}^{n \times n}.$$

$$(F'(x))_{ij}(x) = \frac{\partial f_i}{\partial x_j}(x).$$

## Particularidad del Problema

El problema de resolver un sistema no lineal “determinado” es un caso particular de :

- “Resolver” sobredeterminado (Cuadrados mínimos, en general).
- Minimizar una función (la norma).
- Resolver sin suposiciones de derivabilidad.
- Minimización global.
- Minimización con diferentes tipos de restricciones.

## Generalidad del Problema

Todos los problemas anteriores suelen ser resueltos con esquemas iterativos del tipo  $x^{k+1} = G(x^k)$ .

Todos, y muchos otros, pueden ser considerados “Problemas de Punto Fijo”

$$x = G(x)$$

o sea  $x - G(x) = 0$ .

Luego, escribiendo  $F(x) = x - G(x)$  podemos pensar que “Sistemas No Lineales” es el problema más general posible.

# Libros

J. M. Ortega and W. C. Rheinboldt, *Iterative solutions of nonlinear equations in several variables*, Academic Press, 1970.

J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice Hall, 1983.

+ Ostrowski, Schwetlick, Kelley ...

## Artículos

J. E. Dennis and J. J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Review* 19, pp. 46–89, 1977.

J. M. Martínez. Fixed-point quasi-Newton methods. *SIAM Journal on Numerical Analysis* 29, pp. 1413-1434, 1992.

W. La Cruz, J. M. Martínez and M. Raydan. Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of Computation* 75, pp. 1429-1448, 2006.

# Ejemplos

- Problemas originados en la discretización de EDP, o problemas de contorno en EDO.
- Sistemas no lineales en grafos (Ejemplo, Cálculo de flujos y presiones en cañerías).
- Sistemas con estructuras adicionales, por ejemplo, oriundos de problemas de optimización.
- Problemas de Punto Fijo.

## Ejemplo de un problema importante de punto fijo

Cálculo de Estructuras Electrónicas.

Dados los núcleos de un sistema, calcular los electrones.

Ecuación de Schrodinger, Principio de Pauli, Formulación Hartree-Fock.

Electrones : Matriz de coeficientes  $C \in \mathbb{R}^{K \times N}$  con columnas ortonormales, o Matriz de densidad  $P \in \mathbb{R}^{K \times K}$  dada por  $P = CC^t$ .  
Matriz de Fock :  $Fock(P) \in \mathbb{R}^{K \times K}$  (complicada, cerrada, enorme).  
Problema SCF (Self-Consistent Field) : Encontrar  $P \in \mathbb{R}^{K \times K}$  tal que

La Matriz de Proyeccion en el subespacio generado por los autovectores asociados a los  $N$  menores autovalores de  $Fock(P)$  coincide con  $P$

Observen que esto es “Punto Fijo” : Dado  $P$  calculo  $Fock(P)$ , después calculo los autovectores de  $Fock(P)$ , tomo aquellos correspondientes a los  $N$  menores autovalores ( $C \in \mathbb{R}^{K \times N}$ ), formo  $G(P) \equiv P_{nueva} = CC^t$  y mi objetivo es conseguir  $P$  tal que  $P = P_{nueva}$ .

O sea, el problema es  $F(P) = 0$ , donde  $F(P) = P - G(P)$ .

Ver

L. D. Marks, Robust mixing for ab initio quantum mechanical calculations, Physical Review B 78 (7).

# El método de Newton

Quiero resolver  $F(x) = 0$ , que es difícil.

Uso que  $F(x) \approx F(x^k) + J(x^k)(x - x^k)$ .

Calculo  $x^{k+1}$  tal que  $J(x^k)(x^{k+1} - x^k) = -F(x^k)$ .

Así, tengo un método iterativo al cual llamamos “Método de Newton”.

## Convergencia local cuadrática de Newton

Sea  $x^*$  tal que  $F(x^*) = 0$  y  $J(x^*)$  es no singular. Supongamos que

$$\|J(x) - J(x^*)\| \leq \gamma \|x - x^*\|$$

para todo  $x$  en un entorno de  $x^*$ . Entonces existe un entorno de  $x^*$  tal que, si  $x^0$  pertenece a ese entorno, la iteración de Newton está bien definida, converge a  $x^*$  y satisface

$$\|x^{k+1} - x^*\| = O(\|x^k - x^*\|^2).$$

## Invariancia por transformaciones lineales

Invariancia en el contradominio :

Si aplico Newton al sistema  $AF(x) = 0$  (con  $A$  no singular) obtengo exactamente las mismas iteraciones que si aplico Newton a  $F(x) = 0$ .

Invariancia en el dominio :

Sea  $A$  no singular. Si aplico Newton al sistema  $H(y) \equiv F(Ay + b) = 0$  las iteraciones  $y^k$  satisfacen  $x^k = Ay^k + b$ , donde las iteraciones  $x^k$  son las correspondientes a aplicar Newton a  $F(x) = 0$ .

Estas propiedades son una consecuencia de la Regla de la Cadena.

## Propiedad de descenso de Newton

Supongamos que  $f_i(x^k) > 0$  y que  $x^{k+1}$  es obtenido por Newton. Entonces, de  $F'(x^k)(x^{k+1} - x^k) = -F(x^k)$ , deduzco :

$$\nabla f_i(x^k)^t (x^{k+1} - x^k) = -f_i(x^k) < 0.$$

O sea, la derivada direccional de  $f_i$  a lo largo de la dirección de Newton es negativa.

Análogamente, si  $f_i(x^k) < 0$ , la derivada direccional es positiva.

Esto significa que Newton tiende a mejorar todas las componentes del sistema.

## Newton es Muy Bueno

La convergencia cuadrática, la invariancia por transformaciones lineales y la propiedad de descenso para todas las componentes nos llevan a afirmar que **Newton es muy bueno!**.

Además muchas veces podemos confiar en Newton aún cuando temporariamente empeora. Por ejemplo, pensemos en el caso, en una variable, en donde  $F(x)$  es convexa. (Ver dibujo.)

Este caso está estudiado en el libro de Ortega y Rheinboldt y generalizado en :

M. A. Gomes-Ruggiero, V. L. R. Lopes and J. M. Martínez. Global convergence of Newton-like methods for nonsmooth systems. *Numerical Functional Analysis and Optimization* (9-10), pp. 959-969 (1997).

## Un libro que precisa ser actualizado

J. M. Martínez and S. A. Santos. *Métodos computacionais de otimização*. XX Colóquio Brasileiro de Matemática, IMPA (ISBN 85-244-0092-7, 256 pages) (1995).

Este libro desarrolla lo esencial de la optimización numérica partiendo de “sistemas no lineales”, más precisamente, de la “teoría de convergencia local para métodos que resuelven sistemas no lineales”.

Es un punto de vista válido, porque esa teoría subyace toda la optimización continua.

Copia electrónica más confiable de este libro que la versión original impresa puede ser solicitada a [sandra@ime.unicamp.br](mailto:sandra@ime.unicamp.br).

## Newton Discreto

Cuando, por el motivo que sea (incluyendo la pereza) no tenemos las derivadas de  $F$  podemos aplicar el Método de Newton Discreto, en el cual reemplazamos el Jacobiano por su aproximación por diferencias finitas.

En este caso, la iteración tiene la forma :

$$\left( [F(x^k + he^1) - F(x^k)]/h, \dots, [F(x^k + he^n) - F(x^k)]/h \right) (x^{k+1} - x^k) = -F(x^k).$$

Observen que hay  $n$  evaluaciones adicionales de  $F$  por iteración pero no hay evaluaciones del Jacobiano.

Cuando  $h$  es chico, esto se aproxima mucho a Newton. Pero la diferenciación numérica es un arma de doble filo.

## Newton Inexacto

Para implementar el método de Newton en problemas de gran escala, lo más común es usar subrutinas que resuelven sistemas lineales esparsos por métodos directos (factorización LU) como MA27 o MA57, de HSL.

Cuando el tamaño o la estructura de estos sistemas hacen que hasta estas subrutinas sean inutilizables, podemos emplear métodos iterativos para resolver los sistemas lineales newtonianos.

Por ejemplo, GMRES, LSQR.

En ese caso, uno debe decidir “cuándo parar”.

Lo más común es parar el método iterativo cuando

$$\|J(x^k)s + F(x^k)\| \leq r_k \|F(x^k)\|.$$

Si  $r_k$  es fijo (digamos 0.1) uno obtiene convergencia local lineal.

Si  $r_k \rightarrow 0$  uno obtiene convergencia local superlineal.

Si  $r_k = O(\|F(x^k)\|)$  uno obtiene convergencia local cuadrática.

R. Dembo, S. C. Eisenstat and T. Steihaug, *Inexact Newton Methods*, SIAM Journal on Numerical Analysis 19, 400–408, 1982.

Una cuestión de terminología :

Truncado e Inexacto.

Inexacto : Asociado al criterio de parada. Admite la interpretación de que, por ejemplo, la resolución de un problema discretizado es la resolución inexacta de un problema continuo.

Truncado : Cuando un método iterativo lineal es usado para resolver el sistema lineal Newtoniano.

## Productos Matriz-Vector e Incrementos

Cuando se hace Newton Truncado, el método iterativo lineal suele trabajar haciendo productos Jacobian  $\times$  vector.

Ahora bien,

$$F(x^k + v) \approx F(x^k) + J(x^k)v,$$

$$J(x^k)v \approx F(x^k + v) - F(x^k)$$

Luego puedo reemplazar cada producto matriz-vector por una evaluación adicional del residuo.

## Discreto y Esparso = CPR

CPR (Curtis-Powell-Reid) es una técnica para obtener Jacobianos discretos esparcos sin realizar  $n$  evaluaciones auxiliares por iteración. La idea es aprovechar la estructura.

Por ejemplo, supongamos que sabemos que el Jacobiano es una matriz diagonal (caso ciertamente extremo). En ese caso, podemos obtener su versión discreta con una única evaluación auxiliar por iteración.

En efecto, sea  $\mathbf{e} = (1, \dots, 1)^t$ . Entonces  $[F(x^k + h\mathbf{e}) - F(x^k)]/h$  es una aproximación de los elementos de la diagonal del Jacobiano.

# Convergencia de Newton Discreto

Propiedades parecidas a Newton Analítico.

Cuando  $h$ , el paso de discretización, es suficientemente chico.

Sin embargo, hay que tener cuidado con “ $h$  chico” porque la diferenciación numérica es muy riesgosa.

## Métodos Secantes Clásicos

Notación para el incremento : Usaremos siempre  $s^k = x^{k+1} - x^k$ .  
También :  $y^k = F(x^{k+1}) - F(x^k)$ .

En el libro de Ortega y Rheinboldt encontramos varios “métodos secantes clásicos”.

El más conspicuo es el “Método Secante Secuencial”.

Los métodos secantes se basan en la interpolación lineal del operador  $F(x)$ .

## Método Secante Secuencial

Estamos en  $x^k$ . Consideramos la función afín  $B(x - x^k) + c$  que coincide con  $F(x)$  en los  $(n + 1)$  puntos  $x^k, x^{k-1}, \dots, x^{k-n}$ .

Como queremos que  $B(x - x^k) + c = F(x^k)$  para  $x = x^k$ , debemos definir  $c = F(x^k)$ .

Ahora  $B(x^{k-1} - x^k) + F(x^k) = F(x^{k-1})$  implica que

$$B(x^k - x^{k-1}) = F(x^k) - F(x^{k-1}).$$

También

$$B(x^k - x^{k-2}) = F(x^k) - F(x^{k-2}).$$

Etcétera. Restando :

$$B(x^{k-1} - x^{k-2}) = F(x^{k-1}) - F(x^{k-2})$$

Etcétera.

Juntando todas esas condiciones tenemos :

$$Bs^{k-j} = y^{k-j}, j = 1, \dots, n.$$

Por lo tanto :

$$B = (y^{k-n}, \dots, y^{k-1})(s^{k-n}, \dots, s^{k-1})^{-1}.$$

Luego, la iteración del Método Secante Secuencial es viene dada por

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

en donde

$$B_k = (y^{k-n}, \dots, y^{k-1})(s^{k-n}, \dots, s^{k-1})^{-1}.$$

Observe virtudes y defectos de esta fórmula.

## Corregir Inestabilidades

El defecto del método secante secuencial es que los  $n + 1$  puntos en los cuales quiero interpolar pueden no ser afinmente independientes. O sea, la matriz de incrementos  $(s^{k-n}, \dots, s^{k-1})$  puede no ser invertible lo que hace que no haya una función afín que interpole en los  $n + 1$  últimos puntos.

Eso es corregido descartando el ultimo incremento dependiente y reemplazando por un incremento que envuelve una evaluación auxiliar.

Pero para eso tenemos que descubrir que el último incremento es combinación lineal de los anteriores. Y para eso tenemos que mantener y actualizar una factorización de la matriz de incrementos en el dominio.

## Métodos secantes modernos

También llamados Métodos Quasi-Newton, los métodos secantes modernos se originaron como métodos de minimización (Davidon 1959, Fletcher-Powell 1962).

Los primeros métodos secantes modernos para resolver sistemas fueron introducidos por C. G. Broyden.

C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* 19, 577-593, 1965.

## Broyden Bueno y Broyden Malo

La idea del Método de Broyden “Bueno” es la siguiente : Para calcular  $x^{k+1}$ , buscamos la función afín que interpola, no en todos los últimos  $n + 1$  puntos sino apenas en los dos últimos  $x^k$  y  $x^{k-1}$ . Esta función va a tener la fórmula  $B_k(x - x^k) + F(x^k)$ , como vimos antes. Y, como vimos antes, debe satisfacer la ecuación

$$B_k(x^k - x^{k-1}) = F(x^k) - F(x^{k-1}).$$

Pero no precisa satisfacer las “anteriores”.

Naturalmente, hay muchas matrices que satisfacen esta ecuación.

Por comodidad, en vez de  $B_k(x^k - x^{k-1}) = F(x^k) - F(x^{k-1})$ , escribimos

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

o sea

$$B_{k+1}s^k = y^k.$$

y llamamos a esto “Ecuación Secante”, aunque algunos autores prefieren llamarla “Ecuación Fundamental de los Métodos Quasi-Newton”.

Como hay muchas matrices  $B$  que satisfacen la ecuación secante, Broyden resolvió elegir  $B_{k+1}$  como aquella solución de la ecuación secante que está más cerca de  $B_k$ . En notas autobiográficas, Broyden se dijo motivado por un principio de economía de energía. De esa manera, Broyden definió  $B_{k+1}$  como la matriz  $B$  que resuelve el problema

$$\text{Minimizar } \|B - B_k\| \text{ sujeta a } Bs^k = y^k.$$

Tanto adoptando  $\|\cdot\| = \|\cdot\|_2$  como adoptando  $\|\cdot\| = \|\cdot\|_F$ , este problema tiene la siguiente solución :

$$B_{k+1} = B_k + \frac{(y^k - B_k s^k)(s^k)^t}{(s^k)^t s^k}.$$

La fórmula  $B_{k+1} = B_k + \frac{(y^k - B_k s^k)(s^k)^t}{(s^k)^t s^k}$  define al llamado “Método de Broyden Bueno”.

Observen que  $B_{k+1}$  es igual a  $B_k$  más una matriz de rango 1 (de la forma  $uv^t$ ). La inversa de  $B_{k+1}$  se puede obtener en función de la inversa de  $B_k$  :

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k)(s^k)^t B_k^{-1}}{(s^k)^t B_k^{-1} y^k}.$$

Otra solución de la ecuación secante es la provista por el Método de Broyden “Malo”, también sugerido en el paper de 1965.

En vez de pensar la ecuación secante como  $Bs^k = y^k$ , Broyden-Malo la piensa como  $Hy^k = s^k$ . Aplicando el principio de variación mínima se llega a :

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k)(y^k)^t}{(y^k)^t y^k},$$

con el consecuente método iterativo  $x^{k+1} = x^k - H_k F(x^k)$ .

## Métodos de Broyden en Sistemas Lineales

El Método de Newton tiene la obvia propiedad de resolver sistemas lineales en una única iteración. El Método Secante Secuencial resuelve sistemas lineales en  $n$  iteraciones.

En 1967 Broyden escribía que, lamentablemente, no tenía ninguna propiedad de ese tipo asociada a sus métodos quasi-Newton.

Solamente en 1979, D. M. Gay probó que los métodos de Broyden resuelven sistemas lineales en (lo máximo)  $2n$  iteraciones.

D. M. Gay, Some convergence properties of Broyden's method, *SIAM Journal of Numerical Analysis* 16, pp. 623-630, 1979.

## Condición Dennis-Moré

Consideremos la iteración de tipo Newton  $x^{k+1} = x^k - B_k^{-1}F(x^k)$ . Siempre se supo que, si  $\lim \|B_k - J(x^k)\| = 0$ , la convergencia de esta iteración es superlineal. La razón es Taylor :

$$\begin{aligned} F(x^{k+1}) &= F(x^k) + J(x^k)B_k^{-1}F(x^k) + O(\|F(x^k)\|^2) = \\ &= (I - J(x^k)B_k^{-1})F(x^k) + O(\|F(x^k)\|^2). \end{aligned}$$

Dennis y Moré descubrieron que lo que se precisa para esta propiedad es menos que el hecho de  $B_k$  aproximarse a  $J(x^k)$ . La fórmula arriba ya muestra eso.

Después de sucesivas aproximaciones, Dennis y Moré llegaron a la siguiente elegante formulación del principio mínimo necesario para garantizar que el esquema convergente  $x^{k+1} = x^k - B_k^{-1}F(x^k)$  converja superlinealmente :

$$\lim_{k \rightarrow \infty} \frac{[B_k - J(x^k)]s^k}{\|s^k\|} = 0.$$

## Relación entre la ecuación secante y la Condición Dennis-Moré

La ecuación secante dice

$$B_{k+1}s^k = y^k \quad (= F(x^{k+1}) - F(x^k) = J(x^k)s^k + O(\|s^k\|^2))$$

La condición Dennis-Moré dice

$$B_k s^k = J(x^k) s^k + o(\|s^k\|).$$

Luego, para que la Ecuación Secante implique la condición Dennis-Moré es suficiente que

$$\lim_{k \rightarrow \infty} \|B_{k+1} - B_k\| = 0.$$

## Consecuencia

Si un método iterativo de la forma  $x^{k+1} = x^k - B_k^{-1}F(x^k)$  es convergente a  $x^*$ , donde el Jacobiano es no-singular y satisface una condición de Lipschitz, si las  $B_k$  satisfacen siempre la ecuación secante y si  $\|B_{k+1} - B_k\|$  tiende a cero, entonces la convergencia es superlineal, o sea :

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

## Deterioración acotada

De lo anterior se deduce que para que un método secante convergente de la forma  $x^{k+1} = x^k - B_k^{-1}F(x^k)$  sea superlineal, es suficiente que la diferencia entre  $B'_k$ 's consecutivas tienda a cero. Por otro lado, habría que ver cuál es la condición para que un método de ese tipo sea efectivamente convergente. El principio de deterioración acotada sirve para eso.

## Deterioración acotada en Broyden

En Broyden uno obtiene  $B_{k+1}$  proyectando  $B_k$  en la variedad afín definida por  $Bs^k = y^k$ .

El Jacobiano en  $x^*$  no pertenece, en general, a esta variedad, pero está cerca de la misma.

(La matriz  $\int_0^1 J(x^k + ts^k) dt$  sí pertenece a la variedad (Teorema de Valor Medio).

Al proyectar uno se acerca de todos los elementos del conjunto donde proyecta. De ese modo, no se acerca necesariamente de  $J(x^*)$  pero, por lo menos, no se aleja demasiado. Eso es Deterioración Acotada.

Por la Deterioración Acotada, si  $B_0$  está suficientemente cerca de  $J(x^*)$ , nunca va a quedar demasiado lejos, y, por lo tanto, se va a parecer a Newton siempre, y el método va a converger.

Por otro lado, al proyectar, proyectar, proyectar, el Teorema de Pitágoras nos garantizará que

$$\|B_{k+1} - B_k\| \rightarrow 0.$$

De ese modo, tanto el Método de Broyden como todos los métodos basados en

Proyección en la variedad secante  
 tienen convergencia local superlineal.

Supongamos que  $F(x^*) = 0$ ,  $J(x^*)$  no singular y  $J(x)$  satisface una condición de Lipschitz. Entonces : Existen  $\delta, \varepsilon > 0$  tales que, si

$\|x^0 - x^*\| \leq \varepsilon$  y  $\|B_0 - J(x^*)\| \leq \delta$ , la sucesión definida por  $x^{k+1} = x^k - B_k^{-1} F(x^k)$  converge a  $x^*$  y satisface

$$\lim \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

## Newton, Refinamientos y Quasi-Newton

El método de Newton Con  $p$  Refinamientos consiste en, después de cada iteración de Newton, aplicar  $p$  iteraciones donde se repite el Jacobiano anterior. O sea :

$$x^{k+1} = x^k - J(x^k)^{-1}F(x^k) \text{ si } k \text{ es múltiplo de } p + 1,$$

$$x^{k+1} = x^k - J(x^{k-1})^{-1}F(x^k) \text{ si } k \text{ no es múltiplo de } p + 1$$

Re-definiendo  $x^1 \leftarrow x^{p+1}, x^2 \leftarrow x^{2(p+1)}, \dots$ , este método tiene convergencia superlineal con orden  $2 + p$ .

Una idea natural es hacer “algo mejor que esto” empleando quasi-Newton. Por ejemplo,

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

donde

$$B_k = J(x^k) \text{ si } k \text{ es múltiplo de } p + 1,$$

mientras que  $B_k$  es obtenida a partir de  $B_{k-1}$  por, digamos, Broyden, si  $k$  no es múltiplo de  $p + 1$ .

Las propiedades de convergencia local de “Newton con refinamientos quasi-Newton” no están bien estudiadas. Por un lado, el Teorema de Gay junto con el Teorema de convergencia de Newton indican que estos métodos tienen convergencia superlineal con orden  $4 = 2 + 2$  cada  $2n + 1$  iteraciones. (Si  $p = 2n$ .)

Pero esto es paradójicamente peor que el resultado para Newton con refinamientos simples, que establecería orden  $2 + 2n$  cada  $2n + 1$  iteraciones.

Esto parece contradecir el hecho de que refinamientos quasi-Newton deberían ser mejores que refinamientos simples.

## Quasi-Newton con memoria limitada

En problemas de gran tamaño uno no forma explícitamente las matrices Quasi-Newton ( $B_k$  o  $H_k$ ).

Como uno tiene, digamos, que

$$H_{k+1} = H_k + u^k (v^k)^t,$$

lo que uno hace es guardar los vectores  $u^k$  y  $v^k$  y efectuar los nuevos productos  $H_{k+j} F(x^{k+j})$  consecuentemente.

Por ejemplo,

$$\text{Calculo } x^1 = x^0 - H_0 F(x^0).$$

$$\text{Calculo } u^0, v^0.$$

$$\begin{aligned} \text{Calculo } x^2 = x^1 - H_1 F(x^1) &= x^1 - [H_0 + u^0(v^0)^t] F(x^1) \\ &= x^1 - H_0 F(x^1) - (v^0)^t F(x^1) u^0 \end{aligned}$$

y así sucesivamente.

Cuando el número de vectores  $u^k, v^k$  se me hace demasiado grande o la iteración se me hace demasiado costosa, recomienzo.

## Alternativas para Broyden

Los métodos de Actualización de una Columna por Iteración (Directo o Inverso) consisten en, en cada iteración, modificar solamente una columna de  $B_k$  (en el directo) o de  $H_k$  (en el inverso) para obtener  $B_{k+1}$  o  $H_{k+1}$  respectivamente. La columna que se cambia es la que provoca una menor variación en la aproximación del Jacobiano.

El Método de Actualización de Columna Inverso es así :

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

donde

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k)(e^i)^t}{(e^i)^t y^k}$$

y la columna  $i$  es aquella que maximiza  $|y_1^k|, \dots, |y_n^k|$ .

El Método de Actualización de Columna Inverso fue introducido en :  
J. M. Martínez and M. C. Zambaldi. An inverse column-updating  
method for solving large-scale nonlinear systems of equations.

*Optimization Methods and Software* 1, pp. 129-140 (1992).

Luksan y Vlcek afirman que este método fue el de mejor  
desempeño en estudio numérico exhaustivo publicado en :

L. Luksan and J. Vlcek, Computational experience with globally  
convergent descent methods for large sparse systems of nonlinear  
equations, *Optimization Methods and Software* 8, pp. 201-233,  
1998.

En principio, las propiedades de convergencia de los métodos e actualización de columna no son tan buenas como las de Broyden, porque los primeros no se basan en proyecciones ortogonales, sino en otro tipo de principio variacional.

Sin embargo, preservan algunas propiedades importantes, debidas al Teorema de Gay : Cuando convergen, lo hacen cuadráticamente cada  $2n$  iteraciones.

Al mismo tiempo, su costo computacional es el menor posible que permite preservar la ecuación secante.

# Los métodos quasi-Newton vistos desde 2012

Los métodos quasi-Newton tuvieron sus años de gloria hasta 1990, aproximadamente.

A partir de ahí la comunidad convenció que no había más nada que decir.

Sin embargo, dos potencialidades han sido subestimadas :

- 1 Los métodos quasi-Newton como refinadores naturales de Newton.
- 2 Los métodos quasi-Newton como aceleradores de iteraciones de punto fijo.

## Quasi-Newton y Punto Fijo

Muchos procesos iterativos para resolver cualquier problema son, o pueden ser interpretados como iteraciones de punto fijo :

$$x^{k+1} = G(x^k).$$

Escribiendo  $F(x) = x - G(x)$ , la iteración de punto fijo es :

$$x^{k+1} = x^k - (x^k - G(x^k)) = x^k - F(x^k) = x^k - H_0 F(x^k),$$

onde  $H_0 = I$ .

La consideración de la iteración  $x^{k+1} = x^k - H_k F(x^k)$  en el contexto de arriba, a través de modificaciones quasi-Newton de  $H_k$  es muy natural.

Esencialmente, su efecto a través de las fórmulas quasi-Newton tradicionales sería acelerar las iteraciones de punto fijo, tradicionalmente de convergencia lineal, para convertirlas en superlineales.

## Precondicionadores Secantes

Cuando uno usa el método de Newton Truncado, es bueno usar algún preconditionador razonable para el método iterativo lineal. Una posibilidad, que no tiene que ver con las características del problema, es usar preconditionadores secantes.

La idea es generar  $H_{k+1}$  usando una fórmula secante y obtener la iteración  $k + 1$  aplicando un método iterativo lineal a :

$$H_k J(x^k)(x^{k+1} - x^k) = -H_k F(x^k).$$

Así se obtiene convergencia superlineal.

J. M. Martínez. A theory of secant preconditioners. *Mathematics of Computation* 60, pp. 681-698 (1993).

O sea, en la primera iteración uno resuelve aproximadamente por un método iterativo lineal :

$$H_0 J(x^0) s^0 = -H_0 F(x^0).$$

Calculamos  $x^1 = x^0 + s^0$ ,  $F(x^1)$ ,  $y^1 = F(x^1) - F(x^0)$ , calculamos los vectores de la actualización secante  $u^0, v^0$ .

En la segunda iteración resolvemos aproximadamente por un método iterativo lineal

$$[H_0 + u^0(v^0)^t] J(x^1) s^1 = -[H_0 + u^0(v^0)^t] F(x^1).$$

y así sucesivamente.

Asintóticamente se tiene convergencia superlineal haciendo una única iteración del método iterativo lineal.

## Método del Residuo Espectral

Los sistemas no lineales oriundos de iteraciones de punto fijo son sistemas en los cuales la iteración

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

funciona bien.

El Método del Residuo Espectral (LaCruz-Martínez-Raydan) puede ser interpretado como una aceleración de esta iteración. La idea es calcular  $\alpha \in \mathbb{R}$  tal que la matriz  $\alpha_k I$  satisface la ecuación quasi-Newton de la mejor manera posible.

Minimizando  $\|\alpha s^{k-1} - y^{k-1}\|_2^2$  llegamos a

$$\alpha_k = \frac{(s^{k-1})^t y^{k-1}}{(s^{k-1})^t s^{k-1}},$$

lo que nos permite sugerir el método

$$x^{k+1} = x^k - \frac{1}{\alpha_k} F(x^k).$$

## Piedras en el Camino

- 1 La matriz  $J(x^k)$  es casi singular. Esto se detecta cuando uno hace la factorización LU, en general junto con una estimativa del número de condición. Remedio : Reemplazar  $J(x^k)$  por  $(J(x^k)^t J(x^k) + \mu_k I)^{-1} J(x^k)^t$ . O sea, reemplazar la iteración Newtoniana por

$$(J(x^k)^t J(x^k) + \mu_k I)(x^{k+1} - x^k) = -J(x^k)^t F(x^k).$$

- 2 Algún denominador en alguna fórmula quasi-Newton es cero. Remedio : Diagnosticar que un producto escalar es casi cero si el ángulo es próximo de 90 grados. Acoplar con una tolerancia “absoluta” para las normas de los vectores. Descartar la actualización.

## Métodos con búsqueda unidimensional

En los métodos puros  $x^{k+1} = x^k + s^k$  y no se hace ningún test sobre la calidad de  $x^{k+1}$ .

Varios motivos pueden llevar al uso de métodos impuros.

- 1 El incremento  $\|s^k\|$  es desproporcionadamente grande, lo que indica mal-condicionamiento no detectado antes.
- 2 El valor de  $\|F(x^k + s^k)\|$  no es satisfactorio.

## Control de Paso

Si  $\|s^k\|$  es desproporcionadamente grande, reemplazarlo por  $t_k s^k$  para un valor razonable de  $t_k > 0$  es sensato.

También es sensato proceder como si mal-condicionamiento hubiera sido detectado en la factorización.

## Crecimiento inadecuado de la norma del residuo

Si  $s^k$  es fruto del método de Newton, la propiedad de descenso de este método nos dice que, para  $t_k > 0$  suficientemente chico vamos a conseguir

$$\|F(x^k + t_k s^k)\| < \|F(x^k)\|.$$

Esto puede ser bueno, aunque ya sabemos que el método de Newton es suficientemente confiable como para tolerar algunos aumentos del residuo.

Hay diferentes estrategias “no monótonas” que permiten crecimiento ocasional de la norma del residuo, sin perjudicar el comportamiento general del método.

En el caso de iteraciones quasi-Newton el caso no es tan claro, porque no tenemos la propiedad de descenso.

A veces se intentan valores positivos y negativos de  $t_k$ , con módulo decreciente, en la tentativa de disminuir el valor de la norma del residuo.

No hay nada teóricamente apasionante en este sentido.

# Métodos Homotópicos

La idea de los métodos homotópicos para resolver  $F(x) = 0$  consiste en construir una homotopía  $H(x, t) = 0$  de manera que  $H(x, 0) = 0$  sea un problema fácil mientras que  $H(x, 1) = F(x)$ . La curva correspondiente es “integrada” entre 0 y 1 apostando que, para resolver para determinado  $t$ , un buen punto inicial es la solución para un  $t$  un poco menor.

## Homotopías populares

Las homotopías más populares son la Newtoniana :

$$H(x, t) = (1 - t)[F(x) - F(x^0)] + tF(x),$$

y la homotopía regularizante :

$$H(x, t) = (1 - t)(x - x^0) + tF(x).$$

El paquete Hompack está basado en la homotopía regularizante. L. T. Watson, S. C. Billups, and A. P. Morgan, Algorithm 652 : HOMPACT : a suite of codes for globally convergent homotopy algorithms, *ACM Transactions on Mathematical Software* 13, pp. 281-310, 1987.

El método de Newton funciona bien para problemas en los cuales las derivadas no existen en todos los puntos, más precisamente, en un conjunto de medida nula.

Los problemas en los cuales Newton “funciona” se llaman “semisuaves”.

Si un operador  $F$  es localmente Lipschitz entonces es derivable en casi todo punto.

$\partial F(x) =$  Los límites de los Jacobianos de las sucesiones que tienden a  $x$ .

Semisuave :

$$\lim_{h \rightarrow 0, V \in \partial F(x+h)} \frac{\|F(x+h) - F(x) - Vh\|}{h} = 0.$$

Para problemas semisuaves Newton es localmente superlineal.

L. Qi and J. Sun, A nonsmooth version of Newton's method, *Mathematical Programming* 58, pp. 353-368, 1993.

Los métodos de Newton Inexactos también funcionan en esas circunstancias. Aunque es obligatorio que  $r_k \rightarrow 0$ .

J. M. Martínez and L. Qi. Inexact Newton methods for solving nonsmooth equations. *Journal of Computational and Applied Mathematics* 60, pp. 127-145 (1995).

## Ejemplo de sistema semisuave

Considere el problema de optimización con restricciones :

$$\text{Minimizar } f(x) \text{ sujeta a } g(x) \leq 0.$$

Las condiciones KKT de optimalidad pueden ser formuladas así :

$$\nabla f(x) + \sum_{i=1}^m \mu_i g_i(x) = 0$$

$$\min\{\mu_i, g_i(x)\} = 0, \quad i = 1, \dots, m.$$

Métodos para minimización con restricciones pueden ser fundados en este hecho.

## Problemas sobredeterminados

Si  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  con  $m \geq n$  tengo un problema sobredeterminado.  
En vez de ser

$$J(x^k)(x - x^k) = -F(x^k),$$

Newton pasa a ser

$$J(x^k)^t J(x^k)(x - x^k) = -J(x^k)^t F(x^k)$$

y se llama Gauss-Newton (Minimiza  $\|J(x^k)(x - x^k) + F(x^k)\|_2^2$ ).  
Levenberg-Marquardt lidia bien con variables correlacionadas :

$$[J(x^k)^t J(x^k) + \mu_k I](x - x^k) = -J(x^k)^t F(x^k).$$

## Dos formas de globalizar

$$[J(x^k)^t J(x^k) + \mu_k I](x^{k+1} - x^k) = -J(x^k)^t F(x^k).$$

y

$$[J(x^k)^t J(x^k) + \mu_k I](x - x^k) = -J(x^k)^t F(x^k).$$

seguido de  $x^{k+1} = x^k + t_k(x - x^k)$ .

## Problemas indeterminados

Se trata de resolver  $F(x) = 0$  donde  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \leq n$ .

Cualquier problema de “viabilidad” es de este tipo.

Hay un Método de Newton Natural :

$$x^{k+1} = x^k - J(x^k)^\dagger F(x^k).$$

Hay Quasi-Newton, Hay Newton Inexacto.

# Sistemas No Lineales, Newton y sus amigos, dentro de un algoritmo de Optimización con Restricciones

A eso dedicaremos las próximas transparencias.

# Introducción

IPOPT es el algoritmo con más reputación para resolver problemas de optimización con restricciones, de gran escala.

El programa es libremente disponible en la Internet.

La principal referencia es :

A. Wächter and L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, *Mathematical Programming* 106, 25–57 (2006).

## Ipopt en 6 líneas

Ipopt minimiza  $f(x)$  sujeta a  $h(x) = 0$ ,  $x \geq 0$ . ( $\ell \leq x \leq u$ ).

Ipopt usa barrera logarítmica. En cada Outer Iteration minimiza aproximadamente  $f(x) - \mu \sum \log(x_i)$  sujeta a  $h(x) = 0$ .

Para resolver esos subproblemas (de minimización con restricciones de igualdad) usa **Newton con salvaguardas, detalles y cuidados**.

Para entender lo que Ipopt hace en general, vamos a analizar lo que hace en casos particulares.

Ipopt resuelve problemas de PNL dados en la forma :

$$\text{Minimizar } f(x) \text{ s. t. } h(x) = 0, \ell \leq x \leq u.$$

## Ipopt en el caso sin restricciones

Supongamos que usamos Ipopt para minimizar  $f(x)$ , sin restricciones (sin  $h(x) = 0$ , sin "bounds").

En este caso, Ipopt es el método de **Newton**, con las siguientes especificaciones :

Siendo Newton, en cada iteración, Ipopt "considera" el sistema lineal :

$$\nabla^2 f(x^k)d = -\nabla f(x^k).$$

Al factorizar la Hessiana, el método de factorización le permite obtener el número de autovalores positivos, negativos y ceros. (Sin calcularlos!)

Uno quiere que todos los autovalores sean positivos. Si esto ocurre, Ipopt resuelve de hecho el sistema y obtiene la dirección de búsqueda  $d \in \mathbb{R}^n$ .

Si hay menos que  $n$  autovalores positivos,  $l_{\text{popt}}$  aumenta la diagonal de la Hessiana hasta obtener una matriz definida positiva. Cuando consigue esto, resuelve el sistema y obtiene la dirección  $d$ .  $d$  va a ser una dirección “de descenso”.

## Cómo aumenta la Diagonal ?

Comienza sumando  $10^{-4}$  a todos los términos de la diagonal de la Hessiana.

Si con eso todavía no quedó definida positiva, aument  $10^{-2}$  a la diagonal de la Hessiana, después  $10^0$ ,  $100$ ,  $10^4$ , ... y así sucesivamente, hasta un máximo de valor espantoso. En algún momento consigue que la matriz sea definida positiva y, entonces, calcula  $d$ .

lpopt guarda “el aumento” de una iteración para la siguiente. En la iteración “siguiente”, lpopt trata de disminuir un poco el aumento usado en la iteración anterior (divide por 3).

## Cálculo del paso

En el caso irrestricto que estamos analizando, la próxima iteración es :

$$x^{k+1} = x^k + t_k d^k,$$

donde  $d^k$  es la dirección Newtoniana modificada obtenida arriba y  $t_k$  es obtenido por backtracking comenzando con el paso unitario. `lpopt` indicando `Convergência`, cuando la norma-infinito del gradiente es menor que  $\epsilon_{tol}$ .

Con esto, ya sabemos lo que es `lpopt` en el caso irrestricto.

## Ipopt con restricciones lineales de igualdad

Consideremos ahora Ipopt aplicado a :

$$\text{Minimizar } f(x) \text{ sujeta a } Ax = b.$$

Punto inicial viable tal que  $Ax^0 = b$ .

Ipopt no va a abandonar nunca la viabilidad.

Dado  $x^k$ , para obtener el próximo iterando, Ipopt de nuevo usa Newton. O sea, “considera” el sistema lineal :

$$\nabla^2 f(x^k)(x - x^k) + A^T(\lambda - \lambda^k) = -[\nabla f(x^k) + A^T \lambda^k]$$

$$A(x - x^k) = 0.$$

## Corrección de la Inercia

Para que este sistema lineal nos dé una dirección  $d = x - x^k$  que sea de descenso para la función objetivo, se necesita (condición suficiente) que la Hessiana sea “definida positiva en el espacio nulo de  $A$ ”. Para que el sistema tenga solución se precisa que el rango de  $A$  sea  $m$ . Esas dos condiciones juntas dicen que la matriz del sistema tiene inercia  $(n, m, 0)$ , o sea,  $n$  autovalores positivos,  $m$  negativos y ninguno nulo.

Factorizaciones LU avanzadas, como MA27, nos dan la inercia.

Consideremos primero el caso en que el rango de  $A$  es  $m$ . En este caso, `lpopt` va a llamar a MA27, y, si no encuentra la inercia correcta, va a aumentar la diagonal Noroeste usando los mismos procedimientos que en el caso irrestricto, hasta obtener la inercia correcta.

Mas também pode ser que a inércia esteja errada por haver autovalores nulos. Este caso corresponde a  $\text{posto}(A) < m$ . O remédio é colocar na diagonal Sudeste um número pequeno negativo.

## Complicações no Sudeste

Observem que “colocar um número pequeno na diagonal Sudeste” pode fazer perder a viabilidade do próximo iterando. No documento que eu li não se fala desta situação, porque ele trata diretamente de restrições não lineares onde a viabilidade se perde e isso não é nenhuma tragédia.

No caso de restrições lineares seria uma pena perder a viabilidade por tão pouca coisa.

## A direção de busca

Depois de conseguir a inércia correta, e supondo que as dificuldades do Sudeste foram adequadamente resolvidas sem perdas de viabilidade, a direção  $d$  no espaço  $x$  é uma direção de descida para  $f$ , e Ipopt procede a encontrar  $x^{k+1}$  como no caso irrestrito.

Com isto, sabemos o que Ipopt faz no caso de minimizar uma função com restrições lineares de igualdade, com a ressalva de que, em caso de que o posto de  $A$  seja menor que  $m$ , Ipopt passa a trabalhar como se as restrições fossem não lineares e a viabilidade pudesse ser perdida.

Também usamos, neste caso, que o ponto inicial é viável. Se não for, melhor encontrar um viável antes.

## Critério de parada escalado

No caso sem restrições vimos que o critério de parada é que a norma infinito do gradiente é menor que uma tolerância  $\epsilon_{tol}$ . No caso de minimização com somente com restrições  $Ax = b$ , considerando que os iterandos são sempre viáveis, o natural seria parar quando a norma infinito de  $\nabla f(x^k) + A^T \lambda^k$  é menor que  $\epsilon_{tol}$ . Entretanto, Ipopt observa que, quando há restrições, o tamanho dos multiplicadores de Lagrange pode ser espantoso, especialmente se os gradientes das restrições não são LI. Em consequência, Ipopt usa o critério de parada escalado :

$$\frac{\|\nabla f(x^k) + A^T \lambda^k\|_{\infty}}{s_d} \leq \epsilon_{tol},$$

onde

$$s_d = \max\{1, \|\lambda^k\|_1 / [100(m + n)]\}.$$

## Ipopt em caixas

Agora vamos ver o que faz Ipopt quando o problema é do tipo :  
 “Minimizar  $f(x)$  sujeita a  $\ell \leq x \leq u$ ”. Para simplificar a exposição vamos pensar no caso “Minimizar  $f(x)$  sujeita a  $x \geq 0$ ”. Neste caso, Ipopt emprega a Penalidade Logarítmica. Isto é, as restrições “de caixa” são “eliminadas” mediante :

$$\text{Minimizar } f(x) - \mu \log(x),$$

“sem restrições”, onde o parâmetro de barreira  $\mu > 0$  deve tender a zero, na forma que veremos.

Uma comparação de Ipopt com outros “solvers” no caso de minimização em caixas pode ser encontrada em paper 2012 de Birgin e Gentil, por aparecer em COAP.

## Newton e o truque para mal-condicionamento

Pensemos em Sua Majestade, o método de Newton, para resolver o problema barreirizado. A direção de busca viria de resolver o sistema linear :

$$\nabla^2(f(x^k) - \mu \log(x^k))d = -\nabla(f(x^k) - \mu \log(x^k)).$$

ou seja :

$$[\nabla^2 f(x^k) + \mu X_k^{-2}]d = -[\nabla f(x^k) - \mu X_k^{-1} e].$$

Este sistema é muito mal condicionado quando alguma coordenada de  $x^k$  está próxima de zero, mas, como acontece em Penalidade Externa, o mal-condicionamento é “consertável”.

Para isso, considerem que o que queremos é resolver

$$\nabla[f(x) - \mu \log(x)] \equiv \nabla f(x) - \mu X^{-1}e = 0,$$

e definamos as variáveis auxiliares  $z_i = \mu/x_i$ . Logo, o sistema não linear acima fica

$$\nabla f(x) - z = 0, \quad x_i z_i = \mu_i \quad \forall i.$$

Observem que o sistema resultante é uma  $\mu$ -perturbação das condições KKT.

Agora aplicamos Newton a este sistema, obtendo :

$$\nabla^2 f(x^k)(x - x^k) - (z - z^k) = -\nabla f(x^k) + z^k,$$

$$x_i^k(z_i - z_i^k) + z_i^k(x_i - x_i^k) = -x_i^k z_i^k + \mu \quad \forall i.$$

Colocando em evidência  $z$  no segundo bloco e substituindo no primeiro se obtém um sistema de  $n \times n$  que `lpsolve` resolve. Uma vez obtido o “novo”  $x$ , o novo  $z$  sai do segundo bloco.

Observem que, desta maneira, temos uma sequência para  $x$  e também uma sequência para as variáveis  $z$ . As variáveis  $z$  podem ser consideradas estimativas para os multiplicadores de Lagrange associados com  $x \geq 0$ .

## A aceptación do passo

Há um trial point para  $x$ , mas também há um trial point para  $z$ , que vem do sistema Newtoniano. Assim, há um passo sugerido para  $x$  e um passo sugerido para  $z$ . Como queremos ficar dentro da positividade, ambos passos devem ser suficientemente pequenos para garantir que o próximo iterando, tanto em  $x$  como em  $z$  seja positivo. O passo efetivo se obtém, para  $x$ , fazendo backtracking. Para  $z$  o passo efectivo é o passo máximo que a manutenção da positividade exige.

## Contas com $n = 1$

Problema barreirado : Minimizar  $f(x) - \mu \log(x)$ .

Gradiente igual a zero :  $f'(x) - \mu/x = 0$ . (1)

Newton básico :  $(f''(x_k) + \mu/x_k^2)(x - x_k) = -f'(x_k) + \mu/x_k$  mal condicionado.

Mudança de variáveis :  $z = \mu/x$  ou seja  $xz = \mu$ .

Problema equivalente a (1) :  $f'(x) - z = 0, xz = \mu$ . Bem condicionado.(2)

Newton para (2) :  $f''(x_k)(x - x_k) - z = -f'(x_k)$  ,

$z_k(x - x_k) + x_k(z - z_k) = \mu - x_k z_k$ , bem condicionado ( $2 \times 2$ ). (3)

Coloco em evidência  $z$  na segunda de (3) e substituo na primeira :

$(f''(x_k) + z_k/x_k)(x - x_k) = -f'(x_k) + \mu/x_k$ . Comparem com

Newton básico. Vejam que dados  $x_k, z_k$  obtemos  $x_{k+1}$  usando esta fórmula e, depois, obtemos  $z_{k+1}$  usando (3).

## Parada em cada problema de barreira

Vimos que usamos Newton com backtracking e correções de inércia para minimizar a função barreira. Agora vamos ver quando parar, declarando que já foi encontrada uma solução aproximada razoável para o problema barreira. Ipopt pára, devolvendo o controle para a modificação do parâmetro barreira, quando as seguintes condições são satisfeitas :

$$\frac{\|\nabla f(x^k) - z^k\|_\infty}{s_d} \leq 10\mu,$$

e

$$\frac{|x_i^k z_i^k - \mu|}{s_c} \leq 10\mu,$$

onde  $s_d = s_c = \max\{100, \|z^k\|_1/n\}/100$ . O escalamento com  $s_d$  ou  $s_c$  se deve à preocupação pela possível aparição de multiplicadores excessivamente grandes.

## Modificação do parâmetro de barreira $\mu$

O primeiro parâmetro barreira é igual a 0.1.

Depois de resolver cada problema de barreira, o parâmetro barreira  $\mu$  é reduzido usando a fórmula :

$$\mu_{novo} = \max\{\epsilon_{tol}/10, \min\{0.2\mu, \mu^{1.5}\}\}.$$

A modificação do parâmetro barreira está desenhada de maneira a preservar convergência superlinear do método.

## Ipopt com restrições lineares gerais

Para simplificar a exposição, vamos escrever as restrições lineares de igualdade + desigualdade na forma :

$$Ax = b, x \geq 0.$$

Neste caso, Ipopt “elimina” as restrições de desigualdade por meio de penalidade interna (barreira). Isto significa que o problema de minimizar  $f(x)$  sujeito a  $Ax = b, x \geq 0$  se transforma em

$$\text{Minimizar } f(x) - \mu \log(x) \text{ sujeito a } Ax = b.$$

onde  $\mu$  é um parâmetro de penalidade positivo que deve tender a zero.

O truque para corrigir mal-condicionamento do sistema Newtoniano se usa neste caso em todo seu esplendor. Omite detalhes.

## Cuidados que devem ser tomados pelas desigualdades

Tudo o que se faz neste caso é igual ao que se faz no caso de restrições lineares sem desigualdades. A inércia da matriz adequada é calculada e talvez corrigida, a direção de Newton-modificado é computada e se faz backtracking com Armijo.

Entretanto, o ponto inicial deve ser interior, ou seja, deve satisfazer  $x > 0$ , e todos os iterandos também. Isto significa que, em cada iteração, deve-se cuidar de não violar as desigualdades e, em caso de que um “trial point” tiver componentes negativas, deve-se tomar um passo máximo, e um pouquinho menos que isso, para garantir que os pontos fiquem dentro.

# Ipopt somente com restrições (não lineares) de igualdade

Antes de considerar Ipopt em toda sua generalidade, vamos ver o que ele faz no problema “Minimizar  $f(x)$  sujeita a  $h(x) = 0$ ”.  
Observem que este é o primeiro caso onde contemplamos a possibilidade de perder a viabilidade. Em todos os outros casos somente se trabalhava com pontos viáveis.

## Newton, sempre Newton

Dado o ponto corrente  $x^k \in \mathbb{R}^n$  (não necessariamente viável) e a aproximação corrente aos multiplicadores  $\lambda^k \in \mathbb{R}^m$ , Ipopt considera o sistema KKT

$$\nabla f(x) + \nabla h(x)\lambda = 0, \quad h(x) = 0,$$

e resolve o sistema linear Newtoniano associado, com correções de inércia como as já comentadas. Assim, obtemos uma direção primal  $d_x^k$  e uma direção para os multiplicadores  $d_\lambda^k$ . Como resultado do processo de busca unidimensional,  $d_x^k$  acabará multiplicada por um fator menor ou igual que 1. Ipopt multiplica  $d_\lambda^k$  pelo mesmo fator.

# Backtracking

O novo ponto e os novos multiplicadores vão ser obtidos por backtracking usando a direção de Newton. Vejamos as diferentes coisas que podem acontecer :

- (1) O primeiro trial point é testado em relação a algumas características desejáveis que veremos mais adiante. Se ele não satisfaz essas condições, é substituído por uma “second order correction” (SOC), cuja descrição veremos depois. Daqui em diante, chamaremos primeiro trial point tanto ao originado na direção de Newton ou a sua SOC.
- (2) Se um trial point é “pior” que algum iterando anterior tanto em termos de viabilidade como de valor da função objetivo, rejeite o trial point e divida o passo por dois para testar outro.

(3) O trial point  $x_{trial}$  satisfaz

$\|h(x_{trial})\|_1 \leq 10^{-4} \max\{1, \|h(x^0)\|_1\}$ , a direção primal de Newton é uma direção de descida para a função objetivo, mais ainda, mesmo multiplicada pelo “passo” continua com uma derivada direcional bastante negativa (propriedade que inevitavelmente vai desaparecer se o passo for muito pequeno) (ou seja, esta propriedade diz que o passo não foi ainda muito pequeno neste momento do backtrack) e um descenso do tipo Armijo da função objetivo se verifica. Neste caso, o trial point é gloriosamente aceito.

(4)  $\|h(x_{trial})\|_1 > 10^{-4} \max\{1, \|h(x^0)\|_1\}$ , ou a direção primal não é de descida para a função objetivo, ou (quando multiplicada pelo passo) a derivada direcional fica pequena. Neste caso, se  $\|h(x_{trial})\|_1 \leq 0.99999 \|h(x^k)\|_1$  ou a função objetivo em  $x_{trial}$  é menor que a função objetivo em  $x^k$  menos  $0.00001 \|h(x^k)\|_1$ , também aceitamos gloriosamente o trial point.

(5) Quando o trial point não é gloriosamente aceito, o passo se divide por 2, e se calcula outro trial point.

(6) Se, ao fazer backtracking o passo fica muito pequeno (fórmula complicada (23) para definir muito pequeno) abandonar o backtracking e executar uma Fase de Restauração.

## Second Order Correction (SOC)

A SOC é uma correção do passo Newtoniano que é acionada quando o primeiro trial point foi rejeitado (por qualquer motivo) e, ao mesmo tempo, este trial point não é melhor que  $x^k$  em termos de viabilidade. Neste caso, se trata de corrigir o primeiro trial point, e, portanto, a direção de busca, encontrando um outro trial point que promete ser mais viável. A SOC consiste, simplesmente, em fazer uma nova iteração de Newton partindo do trial point fracassado, repetindo a fatoração da matriz para não gastar muito. Se, depois de uma SOC também não se obtém um trial que preste, se executa outra, até um máximo de 4.

## Fase de Restauração

A fase de restauração se executa quando o passo do backtracking ficou excessivamente pequeno.

O processo usado para restauração se baseia na minimização da distância ao ponto corrente, sujeita à viabilidade do ponto restaurado. Para este processo, Ipopt usa Ipopt.

## Ipopt no Problema Geral

Ipopt resolve “Minimizar  $f(x)$  sujeita a  $h(x) = 0, \ell \leq x \leq u$ ”.  
Felizmente, não há nada essencial a acrescentar.

Ipopt resolve esse problema como uma sequência de problemas barreira, diminuindo a barreira com as regras faladas acima. A função de mérito vinculada à otimalidade é sempre a função objetivo original com a barreira logarítmica.

Cada problema barreira é resolvido como no caso “sem bounds” tomando os devidos cuidados com os bounds. Os detalhes desse procedimento são os explicados nas transpas anteriores.

## O Diabo mora nos detalhes (Deus também)

O título do paper Ipopt envolve as palavras “interior point” e “filtros”. De fato, a palavra mais importante não está no título. Esta palavra é “Newton”.

Ipopt é Newton, onde o problema dos bounds escolheu ser resolvido usando barreira.

A “convergência global” é conseguida (não sei às custas de que suposições) com o auxílio de “filtros”.

Ipopt tem uma programação muito cuidadosa e evidentemente exaustiva. Há (pelo menos) 16 parâmetros ajustados para melhor desempenho do método. Vários deles são “dimensionais”, ou seja, o desempenho do método não é invariante por escalamentos de função, restrições ou variáveis, mesmo mudanças de unidades.

## Conclusiones

- Resolver sistemas no lineales es un problema más general de lo que parece.
- Hay muchas aplicaciones importantes a Física, Química, Ingeniería, algunas de las cuales extremadamente desafiantes.
- Newton, Quasi-Newton, Newton Inexacto están vivos.
- Hay generalizaciones a sistemas indeterminados, sobredeterminados, con algún grado de no-diferenciabilidad, etc.
- Hay procedimientos de globalización, basados en optimización o en homotopías.
- Hay sistemas no lineales dentro de algoritmos de optimización con restricciones.