

Tarea 7 - Optimización

Erick Salvador Alvarez Valencia, CIMAT A.C.

I. INTRODUCCIÓN

En el presente reporte se mostrarán los resultados obtenidos con respecto a la tarea 7 de optimización. En la cual se aplicó el método de gradiente conjugado a una función de suavizado, así como la versión no lineal del método a diferentes funciones.

En cada sección se mostrarán gráficas elaboradas con información de las iteraciones del método (norma del gradiente, valor de la función, etc.) así como unas conclusiones del mismo.

mds

Marzo 27, 2018

II. EJERCICIO 1 - GRADIENTE CONJUGADO LINEAL

En esta parte se implementó el método de gradiente conjugado para optimizar la siguiente función:

$$f(x) = \sum_{i=1}^n (x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 \quad (1)$$

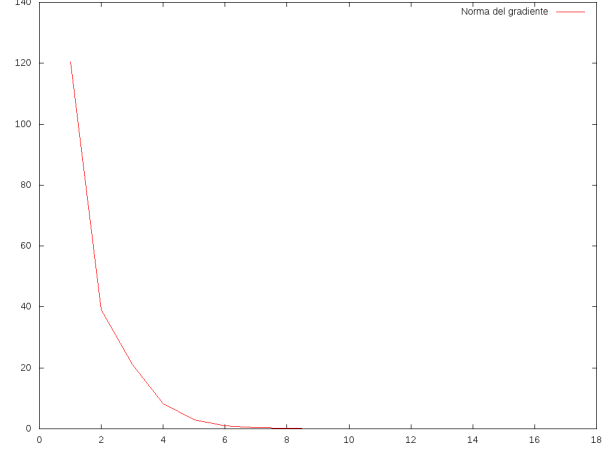
En dicha función el punto y_i y el valor de λ son conocidos. El gradiente de la función anterior es:

$$\nabla f(x) = [2(x_1 - y_1) - 2\lambda(x_2 - x_1) \dots 2(x_i - y_i) + 2\lambda(x_i - x_{i-1}) - 2\lambda(x_{i+1} - x_i) \dots 2(x_n - y_n) + 2\lambda(x_n - x_{n-1})]^T \quad (2)$$

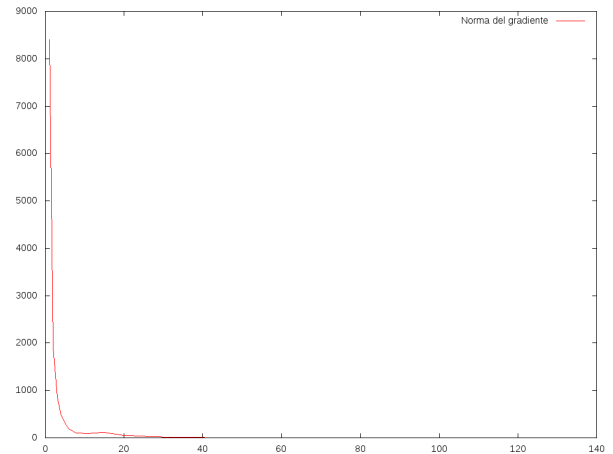
El Hessiano de la función es:

$$\nabla^2 f(x) = \begin{bmatrix} 2 + 2\lambda & -2\lambda & 0 & 0 & \dots & 0 \\ -2\lambda & 2 + 4\lambda & -2\lambda & 0 & \dots & 0 \\ 0 & -2\lambda & 2 + 4\lambda & -2\lambda & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & 0 & -2\lambda & 2 + 2\lambda \end{bmatrix} \quad (3)$$

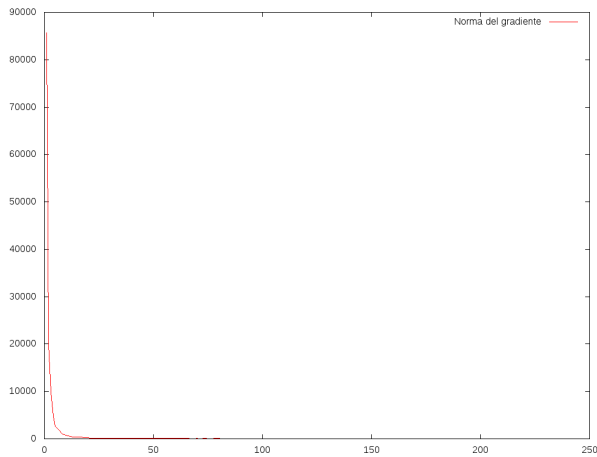
Al aplicarse el método de gradiente conjugado a la función anterior se utilizaron diferentes valores de lambda para ver los resultados. Como este modelo es basado en un suavizado, entre más grande el valor de lambda más suave tiene que verse la función. A continuación se muestran unas gráficas con los resultados obtenidos al aplicar el método de GC con los valores de $\lambda = \{1, 100, 1000\}$, en dichas gráficas se mostrará la evolución de la norma del gradiente a través de las ejecuciones, además de otras que muestran la evolución de la función objetivo a lo largo de las iteraciones.



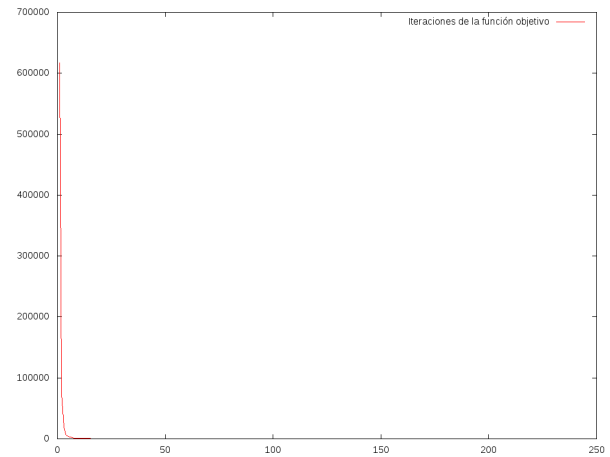
(a) Figura 1. Evolución de la norma del gradiente de la función con $\lambda = 1$.



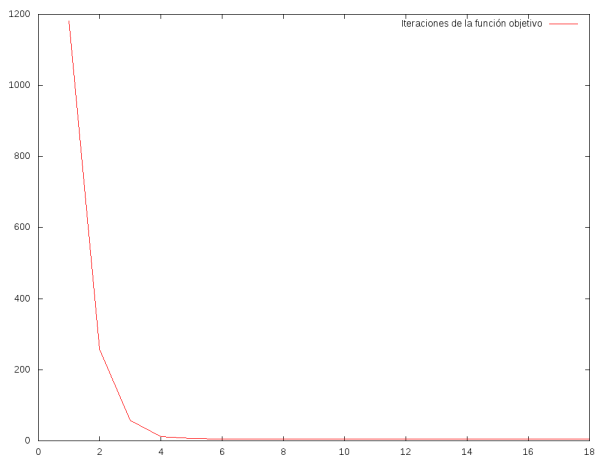
(b) Figura 2. Evolución de la norma del gradiente de la función con $\lambda = 100$.



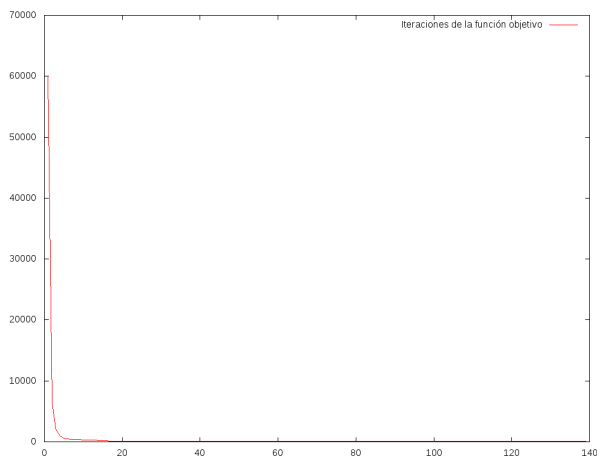
(c) Figura 3. Evolución de la norma gradiente de la función con $\lambda = 1000$.



(f) Figura 6. Evolución del valor de la función con $\lambda = 1000$.



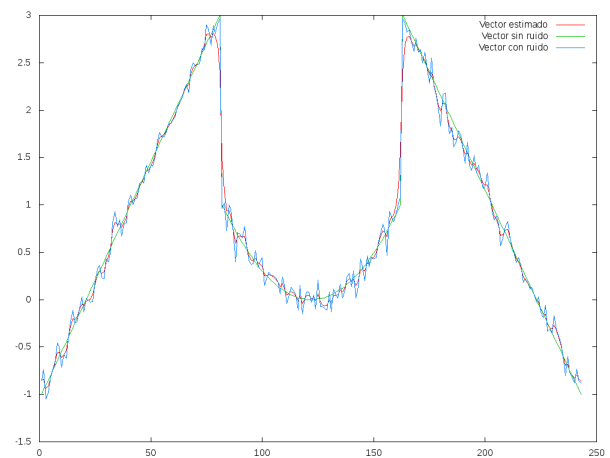
(d) Figura 4. Evolución del valor de la función con $\lambda = 1$.



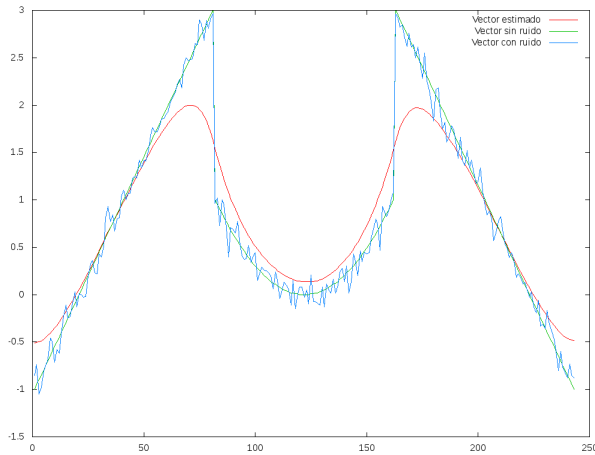
(e) Figura 5. Evolución del valor de la función con $\lambda = 100$.

En las tres gráficas podemos ver la evolución de la norma del gradiente de la función a lo largo de la ejecución del algoritmo con los valores de lambda indicados previamente. Lo primero que podemos notar es que en los tres casos la norma fue descendiendo muy rápido independientemente de que haya iniciado en valores muy altos como es el caso con $\lambda = 1000$, a lo largo de la iteración 20 este ya estaba muy cerca de cero. Otra cosa que podemos notar es que entre más iba creciendo el valor de lambda, más iteraciones necesitaba el algoritmo para converger. Como se sabe el método de gradiente conjugado necesita a lo más n iteraciones para lograr convergencia donde n es la dimensión del problema, y en efecto, cuando $\lambda = 1000$ se puede ver en la Figura 3. que se necesitaron 243 iteraciones para encontrar el punto crítico. Hay que darse cuenta que debido a la simetría de los datos, la gráfica fue contruida hasta 250 en el eje X pero en realidad sólo se realizaron 243 iteraciones.

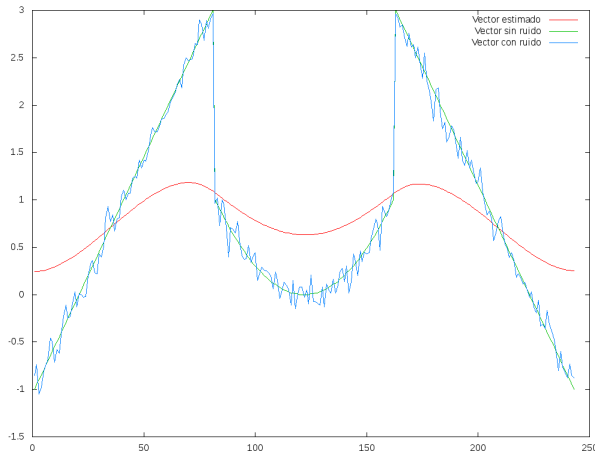
Ahora se mostrarán las gráficas que contienen tanto el vector estimado \hat{x} como el vector sin ruido x y el vector con ruido y .



(g) Figura 7. Vectores \hat{x} , x y y con $\lambda = 1$.



(h) Figura 8. Vectores \hat{x} , x y y con $\lambda = 100$.



(i) Figura 9. Vectores \hat{x} , x y y con $\lambda = 1000$.

En las tres Figuras anteriores podemos observar los tres vectores ya mencionados con la diferencia que se fue variando el valor de λ de 1 a 1000 y de esta forma podemos observar cómo es que entre más grande lambda, la función se va haciendo más suave y se va despegando más de la solución sin ruido x .

III. EJERCICIO 2 - GRADIENTE CONJUGADO NO LINEAL

Para esta parte se implementó la versión no lineal del gradiente conjugado, donde se sabe que en esta versión no se tiene asegurada la convergencia, ni tampoco el hecho de que el algoritmo converja en a lo más n iteraciones.

El principal cambio de esta versión con la lineal es el cálculo de la variable β el cual se tiene varias formas de hacer. Para este ejercicio se implementaron las siguientes Betas:

1. Beta de Fletcher-Reeves :

$$\beta^{FR} = \frac{\nabla^T f(x_{k+1}) \nabla f(x_{k+1})}{\nabla^T f(x_k) \nabla f(x_k)}$$

2. Beta de Polak-Ribiere :

$$\beta^{PR} = \frac{\nabla^T f(x_{k+1}) (\nabla f(x_{k+1}) - \nabla f(x_k))}{\nabla^T f(x_k) \nabla f(x_k)}$$

3. Beta de Hestnes-Stiefel :

$$\beta^{HS} = \frac{\nabla^T f(x_{k+1}) (\nabla f(x_{k+1}) - \nabla f(x_k))}{(\nabla f(x_{k+1}) - \nabla f(x_k))^T d_k}$$

4. Beta de Fletcher-Reeves y Polak-Ribiere : $-\beta^{FR}$ si $\beta^{PR} < -\beta^{FR}$. β^{PR} si $|\beta^{PR}| \leq \beta^{FR}$. β^{FR} si $\beta^{PR} > \beta^{FR}$.

Además de esto el algoritmo se probó con 4 funciones diferentes:

1. Rosenbrock :

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

2. Wood :

$$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10,1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19,8(x_2 - 1)(x_4 - 1)$$

3. Convexa 1 : $n = 1000$

$$f(x) = \sum_{i=1}^n (e^{x_i} - x_i)$$

4. Convexa 2 : $n = 1000$

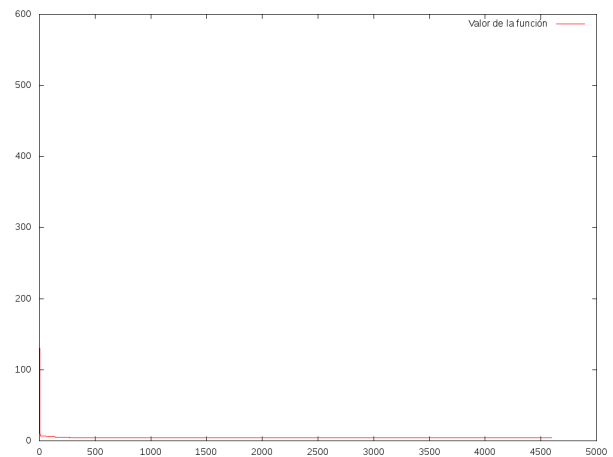
$$f(x) = \sum_{i=1}^n \frac{i}{n} (e^{x_i} - x_i)$$

De las funciones anteriores como dice el nombre de las últimas dos son convexas, por lo cual se sabe que tienen un óptimo global, además de que se puede apreciar que no son lineales.

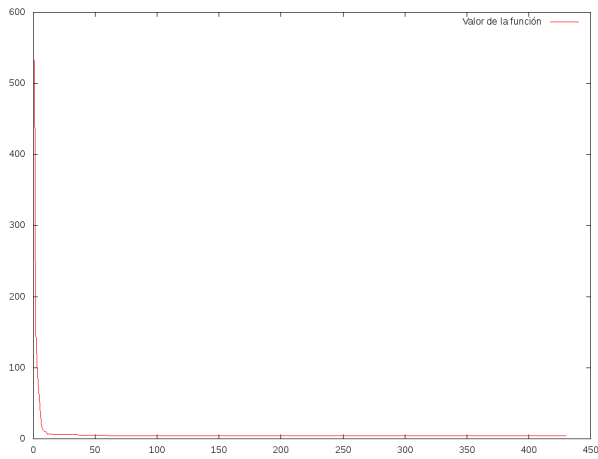
Para el cálculo del tamaño de paso α se utilizó el método de backtracking con las condiciones fuertes de Wolfe para asegurar suficiente descenso.

A continuación se mostrarán las gráficas que corresponden a la evolución de la norma del gradiente, el valor de la función y la norma de la dirección en base a las iteraciones realizadas por el algoritmo. Esto anterior con respecto a cada función y cada valor de Beta usado.

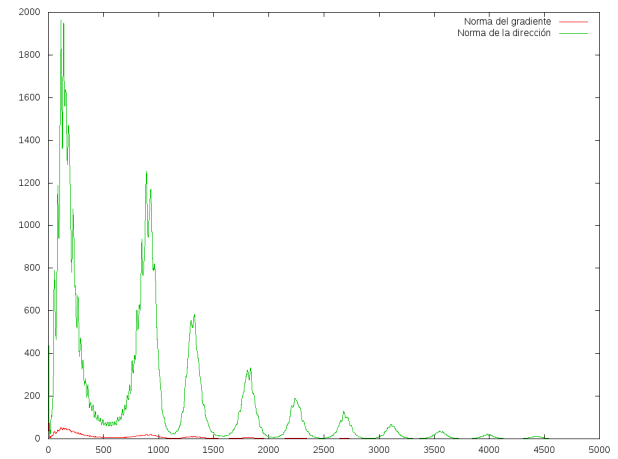
Primero se colocarán las gráficas de la función de Rosenbrock.



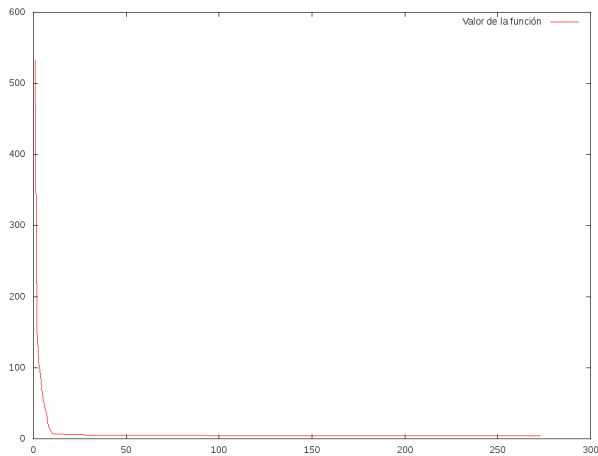
(j) Figura 10. Evolución del valor de la función Rosenbrock usando la Beta de FR.



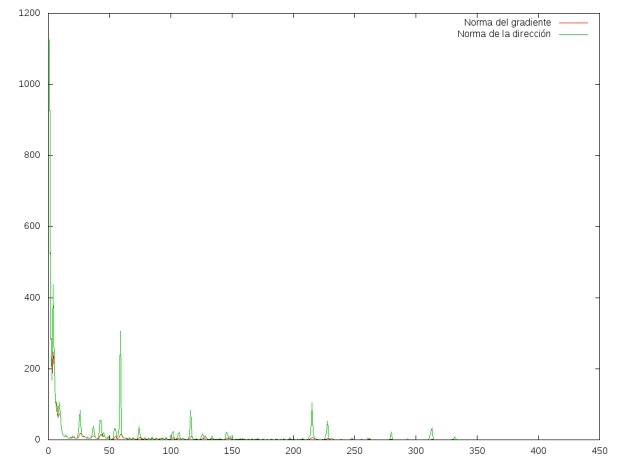
(k) Figura 11. Evolución del valor de la función Rosenbrock usando la Beta de PR.



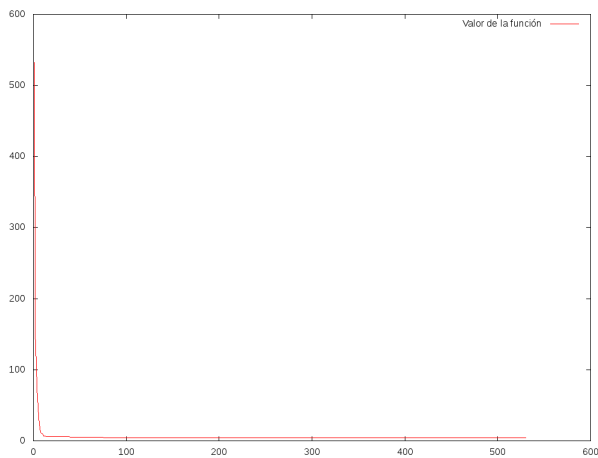
(n) Figura 14. Evolución de la norma del gradiente y la dirección de la función Rosenbrock usando la Beta de FR.



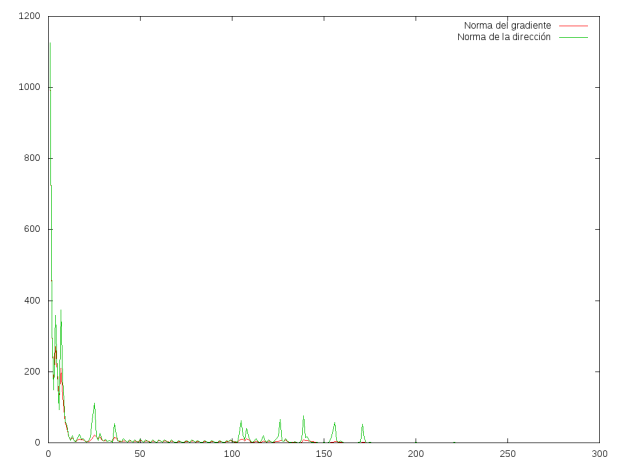
(l) Figura 12. Evolución del valor de la función Rosenbrock usando la Beta de HS.



(o) Figura 15. Evolución de la norma del gradiente y la dirección de la función Rosenbrock usando la Beta de PR.

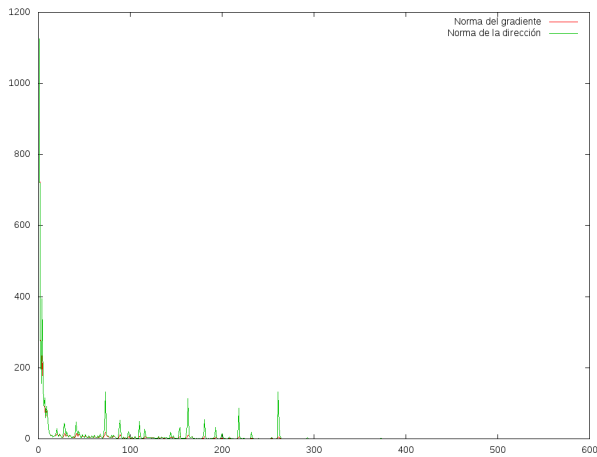


(m) Figura 13. Evolución del valor de la función Rosenbrock usando la Beta de FR-PR.

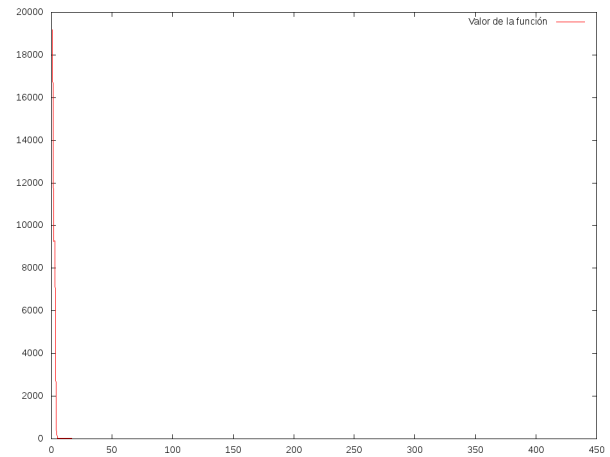


(p) Figura 16. Evolución de la norma del gradiente y la dirección de la función Rosenbrock usando la Beta de HS.

Podemos ver que para cada Beta usada el valor de la función descendió muy rápido (alrededor de 50 iteraciones ya se tenía cercanía a cero).



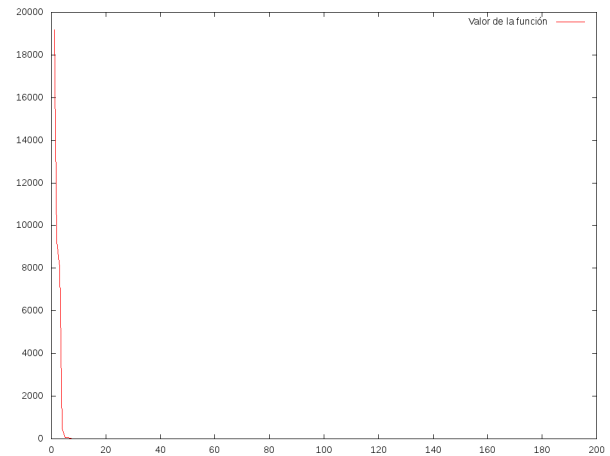
(q) Figura 17. Evolución de la norma del gradiente y la dirección de la función de Rosenbrock usando la Beta de FR-PR.



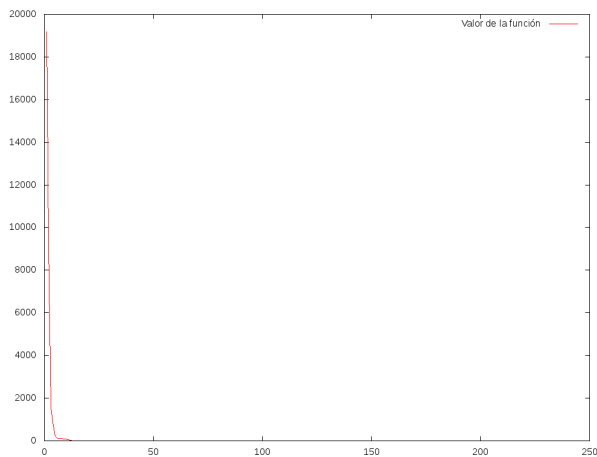
(s) Figura 19. Evolución del valor de la función Wood usando la Beta de PR.

De las gráficas anteriores se puede notar que tanto la norma del gradiente como la de la dirección fueron descendiendo, excepto que para la norma de la dirección (color amarillo) fue oscilando a lo largo de las iteraciones. Esto anterior se aprecia muy bien en la gráfica relacionada a la Beta de Fletcher-Reeves.

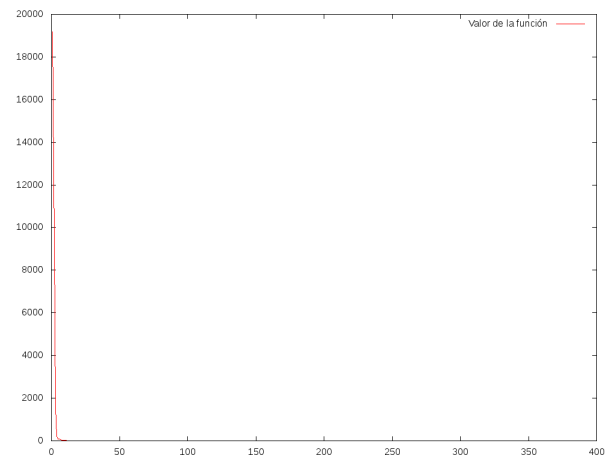
Ahora se mostrarán las gráficas para la función de Wood.



(t) Figura 20. Evolución del valor de la función Wood usando la Beta de HS.



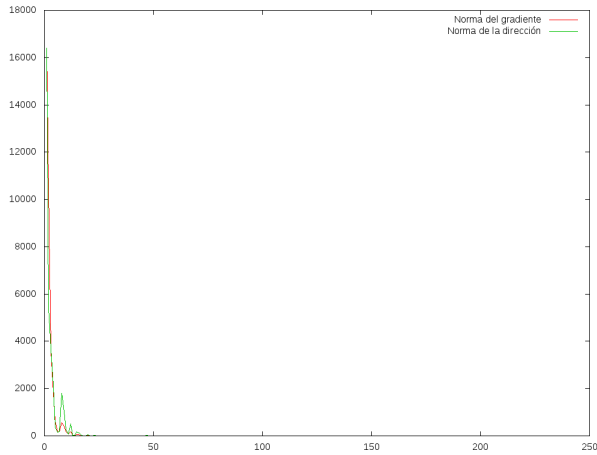
(r) Figura 18. Evolución del valor de la función Wood usando la Beta de FR.



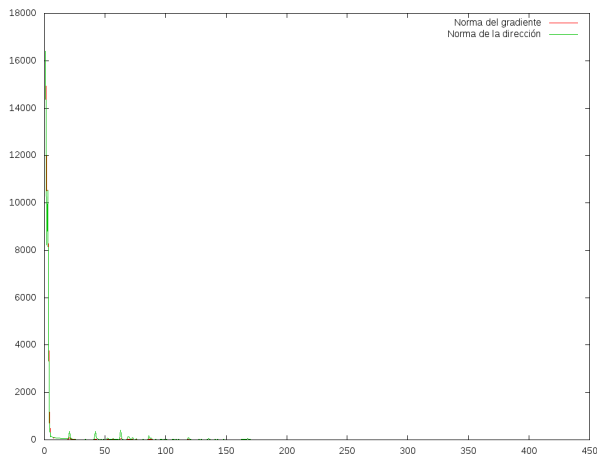
(u) Figura 21. Evolución del valor de la función Wood usando la Beta de FR-PR.

En las 4 gráficas anteriores se puede ver que el valor de la función a lo largo del algoritmo fue descendiendo muy rápido (alrededor de la iteración 20 ya se tenía cercanía a cero). Y

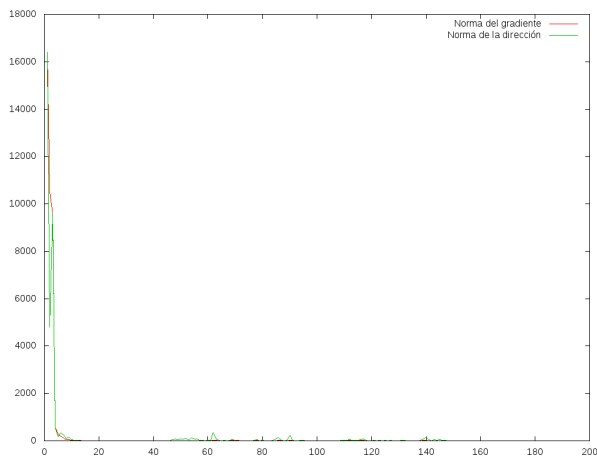
de ahí en adelante esta descendía más despacio hasta cumplir las restricciones de tolerancia.



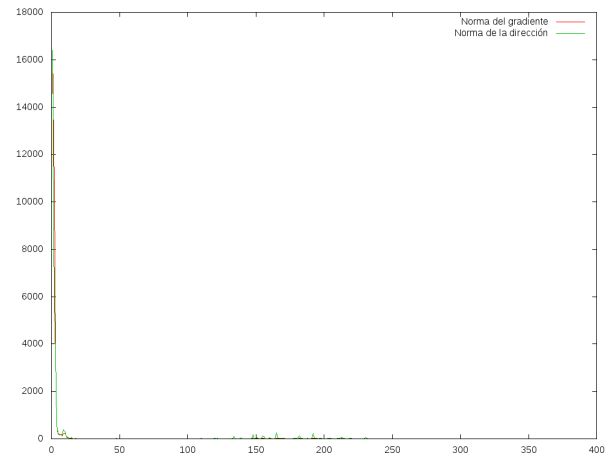
(v) Figura 22. Evolución de la norma del gradiente y la dirección de la función Wood usando la Beta de FR.



(w) Figura 23. Evolución de la norma del gradiente y la dirección de la función Wood usando la Beta de PR.



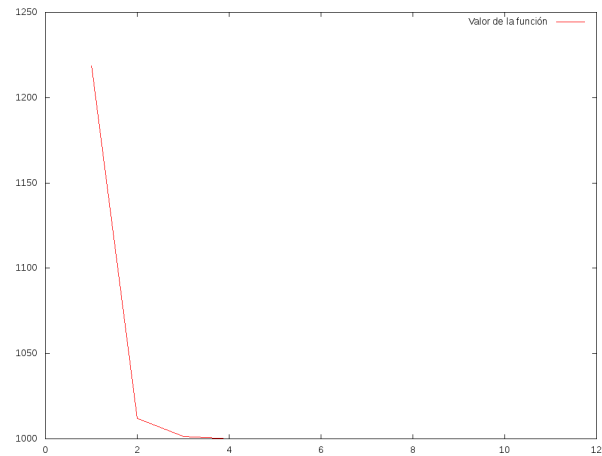
(x) Figura 24. Evolución de la norma del gradiente y la dirección de la función Wood usando la Beta de HS.



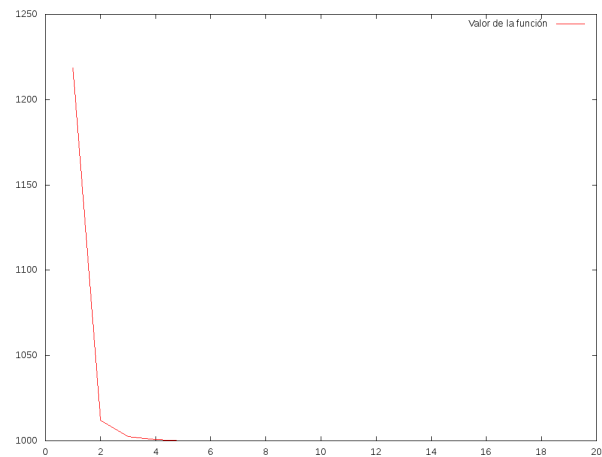
(y) Figura 25. Evolución de la norma del gradiente y la dirección de la función Wood usando la Beta de FR-PR.

Para las 4 gráficas anteriores podemos apreciar que tanto la norma del gradiente como la de la dirección fueron descendiendo muy rápido a través de las primeras iteraciones. Y a diferencia de la función anterior, estas no oscilaron tanto.

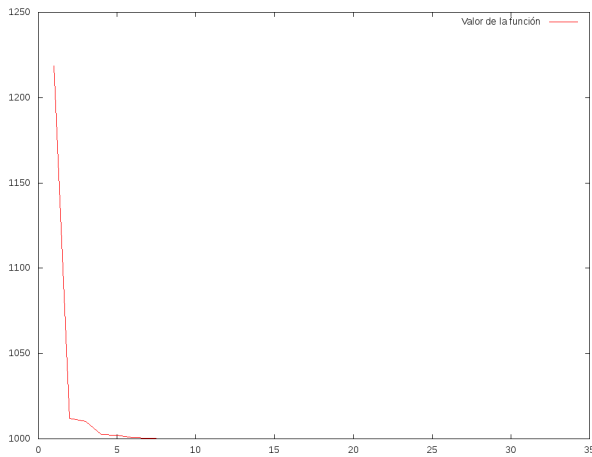
Ahora se mostrarán las gráficas para la función Convexa 1.



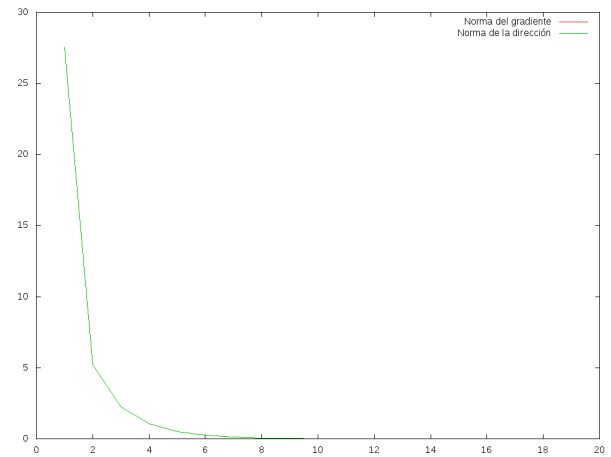
(z) Figura 26. Evolución del valor de la función Convexa 1 usando la Beta de FR.



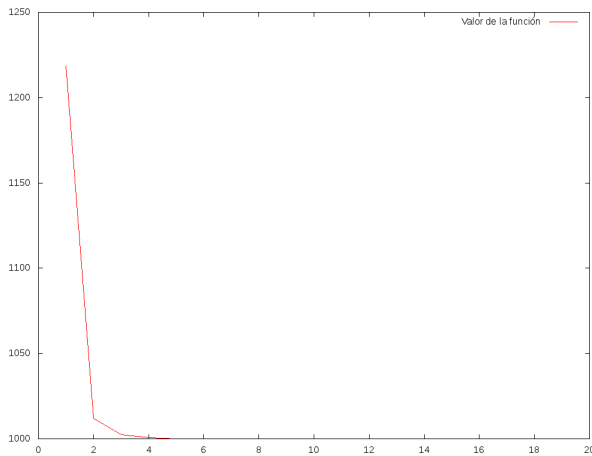
(aa) Figura 27. Evolución del valor de la función Convexa 1 usando la Beta de PR.



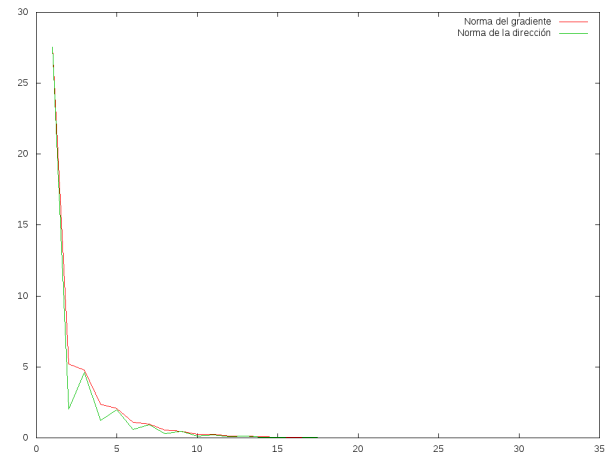
(ab) Figura 28. Evolución del valor de la función Convexa 1 usando la Beta de HS.



(ae) Figura 31. Evolución de la norma del gradiente y la dirección de la función Convexa 1 usando la Beta de PR.

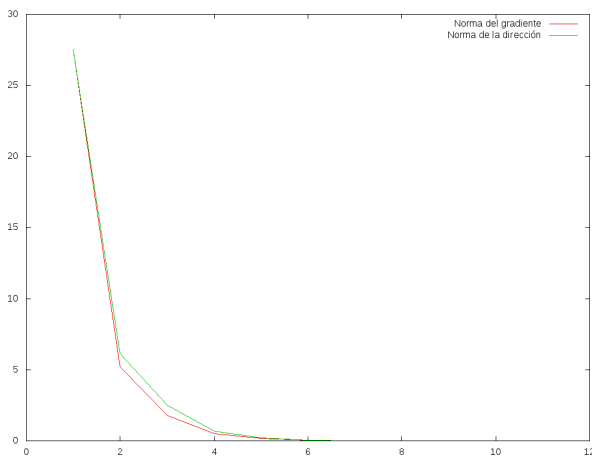


(ac) Figura 29. Evolución del valor de la función Convexa 1 usando la Beta de FR-PR.

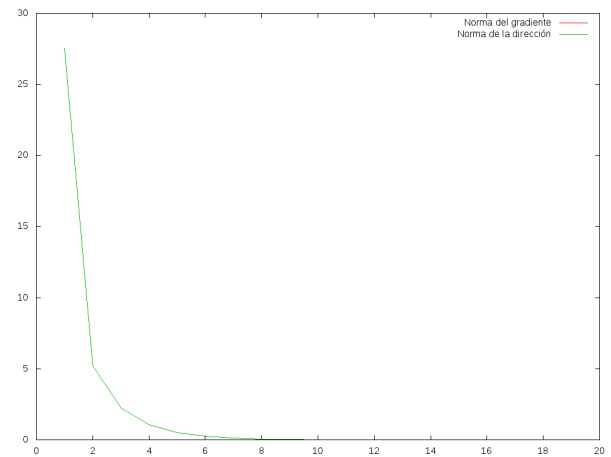


(af) Figura 32. Evolución de la norma del gradiente y la dirección de la función Convexa 1 usando la Beta de HS.

En las 4 gráficas anteriores podemos ver que en este caso el algoritmo convergió bastante rápido para las diferentes Betas, y se puede notar que este convergió hacia el valor 1000 como se tenía supuesto.



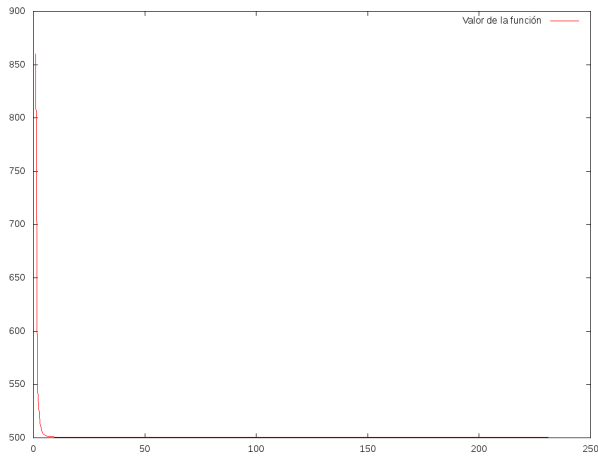
(ad) Figura 30. Evolución de la norma del gradiente y la dirección de la función Convexa 1 usando la Beta de FR.



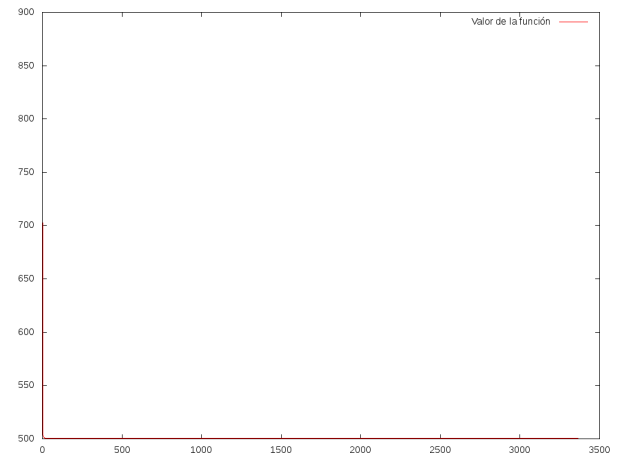
(ag) Figura 33. Evolución de la norma del gradiente y la dirección de la función Convexa 1 usando la Beta de FR-PR.

En las 4 gráficas anteriores se aprecia la evolución de la norma del gradiente y la dirección a lo largo de las iteraciones, estas descendieron de manera muy rápida y a excepción de la Beta de HS no hubo tantas oscilaciones.

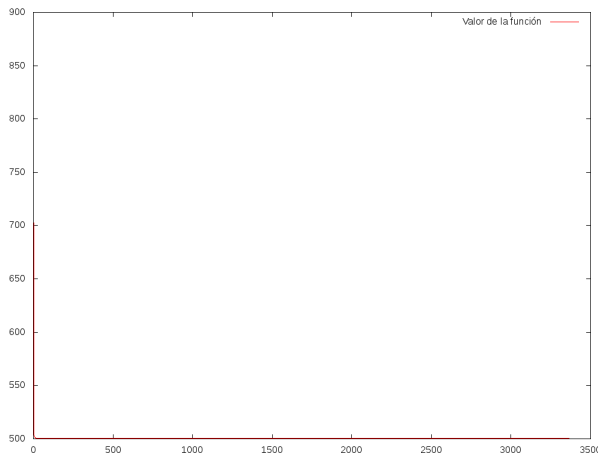
Ahora se mostrarán las gráficas para la función Convexa 2.



(ah) Figura 34. Evolución del valor de la función Convexa 2 usando la Beta de FR.

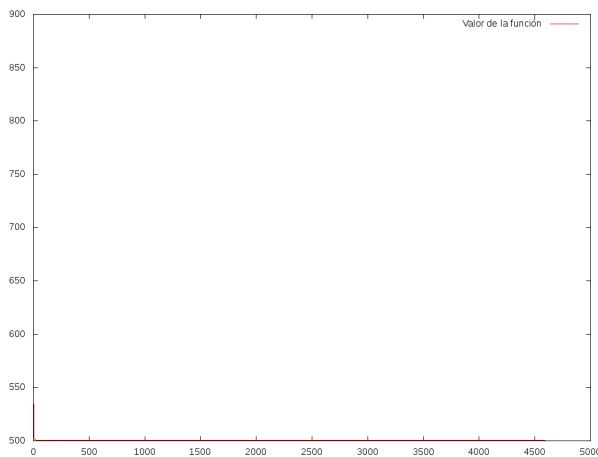


(ak) Figura 37. Evolución del valor de la función Convexa 2 usando la Beta de FR-PR.

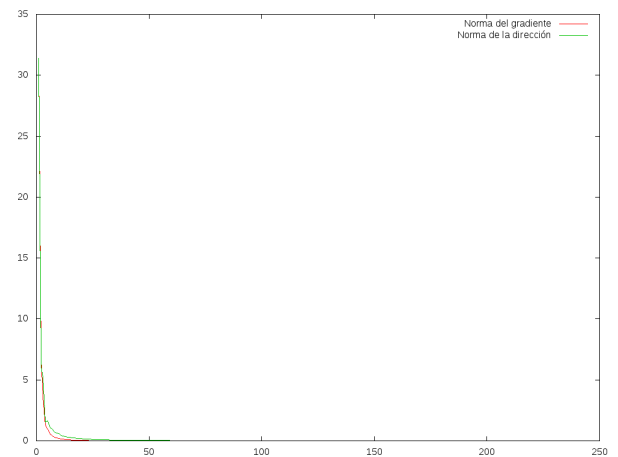


(ai) Figura 35. Evolución del valor de la función Convexa 2 usando la Beta de PR.

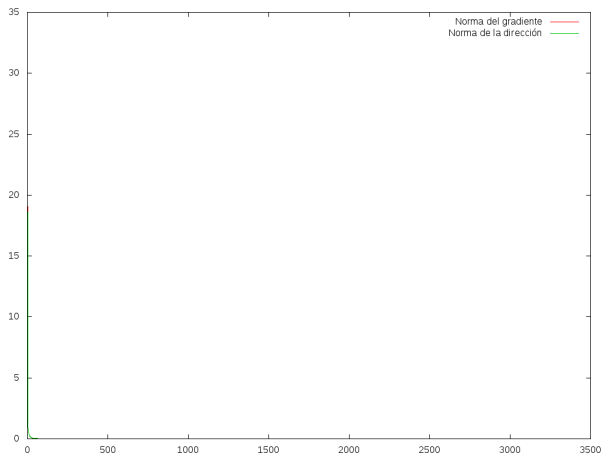
Finalmente tenemos la evolución del valor de la función convexa 2. Para este caso podemos ver que el algoritmo tardó muchas iteraciones en converger, esto debido a que las condiciones fuertes de Wolfe generaron alfas muy pequeñas y el avance era lento, en este caso lo que se hizo fue usar las condiciones de Armijo con un alfa inicial de 1.0 en donde se logró convergencia en menos iteraciones.



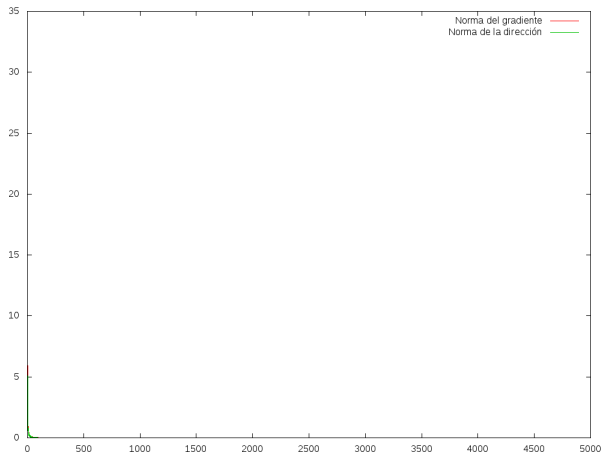
(aj) Figura 36. Evolución del valor de la función Convexa 2 usando la Beta de HS.



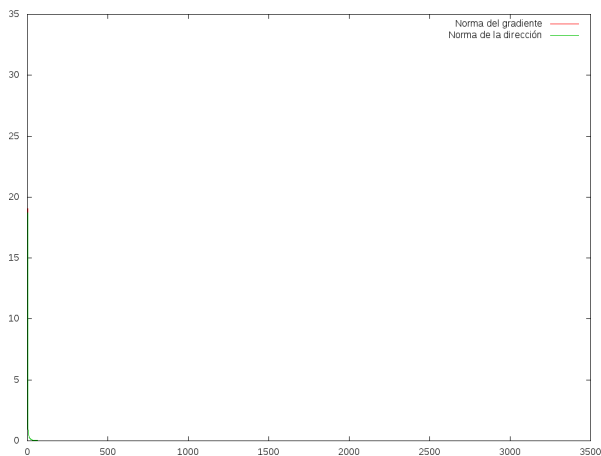
(al) Figura 38. Evolución de la norma del gradiente y la dirección de la función Convexa 2 usando la Beta de FR.



(am) Figura 39. Evolución de la norma del gradiente y la dirección de la función Convexa 2 usando la Beta de PR.



(an) Figura 40. Evolución de la norma del gradiente y la dirección de la función Convexa 2 usando la Beta de HS.



(ao) Figura 41. Evolución de la norma del gradiente y la dirección de la función Convexa 2 usando la Beta de FR-PR.

En las 4 Figuras anteriores se muestra la norma del gradiente y la dirección para la función convexa 2. Podemos ver que el descenso a valores cercanos a cero fue muy rápido pero luego se mantuvo constante esto debido a que en esas partes y por lo explicado anteriormente el descenso fue mucho más rápido.

IV. CONCLUSION

En el presente reporte se analizó el algoritmo de gradiente conjugado en su versión lineal y no lineal. Se pudo ver que para el caso lineal este algoritmo funciona bien ya que se asegura convergencia en a lo más n iteraciones donde n es la dimensión de la función. Para el caso lineal esto no es así ya que no se tiene asegurada la convergencia, además de que el cálculo del tamaño de paso puede generar diversos resultados, desde una convergencia muy rápida hasta disparar el valor de la función a valores muy grandes y lograr que este no converja.