ANÁLISIS APLICADO
PROYECTO I

Condiciones para entregar el proyecto

1. Cada equipo debe tener de 3 a 4 miembros.

Por correo electrónico deben registrar el equipo con los nombres de los miembros y el nombre de la función que van a optimizar. Si la función ya esta tomada de otro equipo, entonces les tengo que pedir que escogen otra.

a) Himmelblau.
b) Styblinski-Tang
c) Booth
d) Beale
e) Matyas
f) Goldstein–Price function
g) McCormick
h) Easom
i) Three-Hump Camel
j) Lévi function N.13
k) Six-Hump Camel
l) GRIEWANK

2. Fecha de entrega: Jueves 2 de Abril a las 23:55 en comunidad.itam.

3. Resolver las tareas abajo.

4. Entregar el código completo en un archivo comprimido de formato `.zip`.
Este debe incluir

- las funciones `pDogleg.m`, `pCauchy.m`, `mRC1.m` y `mRC2.m`. En cualquier caso van a aproximar (o evaluar) gradientes y hessianas. Hay que entregar esas funciones, también.
- Para cada tabla de resultados en su documentación un *script* que reproduce el los números en la tabla.
- Para cada gráfica en su documentación un *script* que reproduce la gráfica desde el mismo punto de vista. Ver `help view`.

El código de los métodos tiene que respectar Laboratorio 4 (ver comunidad.itam. → deConfianza).

5. Entregar un documento en formato `.pdf` que junta y comenta los resultados obtenidos. No quiero código en el documento.

Dr. Andreas Wachtel
Semestre: Primavera 2020.

El proyecto

1. Escribir funciones en MatLab u Octave.

Parámetros centrales:

$$\eta = 0.1 \,, \qquad tol = 10^{-5} \,, \qquad \Delta_{max} = 1.5 \,.$$

Los comentarios están en Ingles, pues MatLab no me acepta acentos.

- Una función que calcula el punto *dogleg*:

```
function [p] = pDogLeg( B, g, delta )
% In :  B     ... an s.p.d. matrix that approximates the hessian of  f  in  xk
%       g     ... (vector) gradient of  f  in  xk
%       delta ... trust region radius
%
% Out:  p     ... The  dogleg  point
```

- Una función que calcula el punto de *Cauchy*:

```
function [pC] = pCauchy( B, g, delta )
% In :  B     ... (symmetric matrix) approximates the hessian of  f  in  xk
%       g     ... (vector) gradient of  f  in  xk
%       delta ... trust region radius
%
% Out:  pC    ... The  Cauchy  point
```

- Una función que implementa el siguiente método de región de confianza.
  Este método debe usar el punto de *Cauchy* y para el modelo cuadrático se usa $B_k \stackrel{\text{def}}{=} \nabla^2 f(\boldsymbol{x}_k)$.

```
function [x, msg] = mRC1( f, x0, itmax )
% Trust region method using the  Cauchy  point.
%
% In :  f     ... (handle) function to be optimized
%       x0    ... (vector) initial point
%       itmax ... (natural number) upper bound for number of iterations
%
% Out:  x   ... (vector) last approximation of a stationary point
%       msg ... (string) message that says whether (or not) a minimum was found
```

- Una función que implementa el siguiente método de región de confianza.
  Este método debe usar el punto *dogleg*. Para asegurar que la hessiana B_k del modelo cuadrático siempre es s.p.d. la definimos de la siguiente manera (en cada paso): Usando `eigs` se determina solamente el eigenvalor más chico, *i.e.*, $\lambda_1 \stackrel{\text{def}}{=} \lambda_{min}\big(\nabla^2 f(\boldsymbol{x}_k)\big)$. Después, se usa

$$B_k \stackrel{\text{def}}{=} \begin{cases} \nabla^2 f(\boldsymbol{x}_k) & , \text{si } \lambda_1 > 0 \\ \nabla^2 f(\boldsymbol{x}_k) + sI & , \text{si } \lambda_1 \leq 0 \end{cases} \qquad \text{donde} \qquad s \stackrel{\text{def}}{=} 10^{-12} - \frac{9}{8}\lambda_1 \,.$$

```
function [x, msg] = mRC2( f, x0, itmax )
% Trust region method using the  dogleg  point.
%   Same arguments and results as the  mRC1  method.
```

⋆ Para definir la respuesta `msg` se debe ver si el eigenvalor mínimo de $\nabla^2 f$ es no-negativo (`help eigs`). Alternativamente, pueden hacer que MatLab intente sacar una factorización Cholesky (`help chol`).

2. Tareas (aplicar su código).

2.1. Dibujar en 3d. Sea $f\colon \mathbb{R}^2 \to \mathbb{R}$ y el punto inicial $\boldsymbol{x}_0$ cerca de un mínimo local $\boldsymbol{x}^\star$. La matriz Hessiana en $\boldsymbol{x}_0$ sea simétrica y (semi-)definida positiva. Escoge una región de confianza con $\Delta > 0$. Dibuje la gráfica en $\mathbb{R}^3$ del modelo cuadrático en la región de la confianza.

Ayuda: ver ejemplo `exampleCircular.m`: (comunidad.itam $\to$ deConfianza $\to$ codeRC.zip)

2.2. Dibujar direcciones en 2d. Sea $f\colon \mathbb{R}^2 \to \mathbb{R}$ y el punto inicial $\boldsymbol{x}_0$ cerca de un mínimo local $\boldsymbol{x}^\star$. La matriz Hessiana en $\boldsymbol{x}_0$ sea simétrica y (semi-)definida positiva y se usa para definir el modelo cuadratico en $\boldsymbol{x}_0$. Escoge una región de confianza con $\Delta > 0$.

Luego haga un *plot* (en dos dimensiones) que contiene

- la frontera de la región de confianza
- algunos conjuntos de nivel en $\mathbb{R}^2$ del modelo cuadrático en la región de la confianza.
- los tres direcciones *Newton, Cauchy, dogleg.* Para obtenerlas use sus funciones `pDogLeg`, `pCauchy`.

Ayuda 1: ver ejemplo en comunidad.itam $\to$ deDescenso $\to$ `scrLevelSet.m`
Ayuda 2: se puede usar `quiver( 1, 2, -.5,-1, 'r', 'LineWidth',2);` para hacer un "vector" rojo que empieza en $(1,2)^T$ y tiene dirección y longitud definidas por $(-1/2,-1)^T$.
Ayuda 3: Con `axis equal;` la frontera de la región de confianza debe ser un circulo.

2.3. Iteraciones y error. Para su función escoge un punto inicial tal que $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2 > \Delta_{max}$ y en el cual la función tiene una Hessiana no positiva definida. En el caso, que la función es globalmente convexa, escoge una distancia mayor que $3\Delta_{max}$. Después, para cada metodo haga una tabla que contiene los utimos 8 iteraciones con su numero, y los valores $\|\nabla f(\boldsymbol{x}_k)\|_2$ y $f(\boldsymbol{x}_k)$.

Además, en el caso que la solución óptima es conocida mide los ultimos 5 errores: $\|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2$.

2.4. Visualizar el proceso. Haz una gráfica similar a las en el Lab. 2 para visualizar el camino que tomo la iteración.

Ayuda: ver ejemplo `scrLevelSet.m`, semana 2.