Optimization-Based Modeling and Design of Electronic Circuits
Assignment 1

Prof. J. E. Rayas-Sánchez
January 2019

1) Calculate "by hand" the Hessian matrix of the bowl and Rosenbrock functions described in the lecture notes. Determine if these two functions have a positive definite Hessian everywhere, if not, determine some regions where these functions have a non-positive definite Hessian.

2) Develop a Matlab function to calculate the derivative of a unidimensional scalar function `fun(x)` at a point `x0`. The function should have the following functionality:

```
% Usage: D = Derivative(fun,x0)
%        fun: name of the scalar function (string).
%        x0: point of interest (scalar).
%        D: derivative of fun at x0 (scalar).
```

Indicate the reference(s) used to write the algorithm. Test your Matlab function with the cases shown in the following table (calculate D and the error $|y'(x_0)-D|$ for each case).

$y(x)$	$y'(x)$	x_0	$y'(x_0)$
$\sin(x)$	$\cos(x)$	0	1
		π	-1
xe^x	$e^x(x+1)$	0	1
		-1	0
		30	$331.280712\times10^{+12}$
x^9	$9x^8$	0	0
		1	9
		-1	9
		2	2,304

3) Taking the previous function as a basis for construction, develop a Matlab function to calculate the gradient of a multidimensional scalar function. The function should have the following functionality:

```
% Usage: g = Grad(fun,x0)
%        fun: name of the multidimensional scalar function (string). This
%             function takes a vector argument of length n and returns a
%             scalar.
%        x0: point of interest (vector of length n).
%        g: vector containing the gradient of fun at x0. The size(g)=
%           size(x0).
```

Indicate the reference(s) used in writing the algorithm. Test your Matlab function with the cases shown in the following table (calculate $\boldsymbol{g}$ and the scalar error $\|\nabla y(\boldsymbol{x}_0)-\boldsymbol{g}\|_2$ for each case).

Periférico Sur 8585
45604 Tlaquepaque, Jal., México
Tel +52 33 3669 3598
www.iteso.mx

$y(\boldsymbol{x})$	$\nabla y(\boldsymbol{x})$	$\boldsymbol{x}_0$	$\nabla y(\boldsymbol{x}_0)$
$(x_1-6)^2+\frac{1}{25}(x_2-4.5)^4$ (bowl function)	$\begin{bmatrix} 2(x_1-6) \\ \frac{4}{25}(x_2-4.5)^3 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -12 \\ -14.58 \end{bmatrix}$
		$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -14 \\ -6.86 \end{bmatrix}$
$100(x_2-x_1^2)^2+(1-x_1)^2$ (Rosenbrock function)	$\begin{bmatrix} -400x_1(x_2-x_1^2)+2(x_1-1) \\ 200(x_2-x_1^2) \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -2 \\ 0 \end{bmatrix}$
		$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

4) Taking the previous function as a basis for construction, develop a Matlab function to calculate the Jacobian of a multidimensional vector function. The function should have the following functionality:

```
% Usage: J = Jacobian(fun,x0)
%        fun: name of the multidimensional vector function (string). This
%             function takes a vector argument of lenght n and returns a
%             vector of length m.
%        x0: point of interest (vector of length n).
%        J: Jacobian of fun at x0 (matrix with m rows and n columns).
```

Indicate the reference(s) used in writing the algorithm. Test your Matlab function with the cases shown in the following table (calculate $\boldsymbol{J}_{\text{Matlab}}$ and the scalar error $\left\| \boldsymbol{J}(\boldsymbol{y}(\boldsymbol{x}_0)) - \boldsymbol{J}_{\text{Matlab}} \right\|_{\text{F}}$ for each case).

$\boldsymbol{y}(\boldsymbol{x})$	$\boldsymbol{J}(\boldsymbol{y}(\boldsymbol{x}))$	$\boldsymbol{x}_0$	$\boldsymbol{J}(\boldsymbol{y}(\boldsymbol{x}_0))$
$\begin{bmatrix} 2x_1^2+3x_2+5\sin x_3 \end{bmatrix}$	$\begin{bmatrix} 4x_1 & 3 & 5\cos x_3 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 3 & 5 \end{bmatrix}$
		$\begin{bmatrix} -100 \\ 40 \\ \pi/2 \end{bmatrix}$	$\begin{bmatrix} -400 & 3 & 0 \end{bmatrix}$
$\begin{bmatrix} 2x_1^2+3x_2 \\ 2x_1x_2+5x_1^3+2x_1x_2^3 \\ -e^{x_1} \end{bmatrix}$	$\begin{bmatrix} 4x_1 & 3 \\ 2x_2+15x_1^2+2x_2^3 & 2x_1+6x_1x_2^2 \\ -e^{x_1} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 3 \\ 0 & 0 \\ -1 & 0 \end{bmatrix}$
		$\begin{bmatrix} 3 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 12 & 3 \\ 115 & 78 \\ -20.085536923188 & 0 \end{bmatrix}$

Periférico Sur 8585
45604 Tlaquepaque, Jal., México
Tel +52 33 3669 3598
www.iteso.mx

5) Taking the two previous functions as a basis for construction, develop a Matlab function to calculate the Hessian of a multidimensional scalar function. The function should have the following functionality:

```
% Usage: H = Hessian(fun,x0)
%        fun: name of the multidimensional scalar function (string). This
%             function takes a vector argument of length n and returns a
%             scalar.
%        x0: point of interest (vector of length n).
%        H: Hessian of fun at x0 (n by n symmetric matrix).
```

Indicate the reference(s) used in writing the algorithm. Test your Matlab function with the cases shown in the following table and show the corresponding results (calculate $\boldsymbol{H}_{\text{Matlab}}$ and the scalar error $\left\|\boldsymbol{H}(y(\boldsymbol{x}_0))-\boldsymbol{H}_{\text{Matlab}}\right\|_{\text{F}}$ for each case).

$y(\boldsymbol{x})$	$\boldsymbol{H}(y(\boldsymbol{x}))$	$\boldsymbol{x}_0$	$\boldsymbol{H}(y(\boldsymbol{x}_0))$
$(x_1-6)^2+\frac{1}{25}(x_2-4.5)^4$ (bowl function)	?	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}2 & 0\\0 & 9.72\end{bmatrix}$
		$\begin{bmatrix}6\\4.5\end{bmatrix}$	$\begin{bmatrix}2 & 0\\0 & 0\end{bmatrix}$
$100(x_2-x_1^2)^2+(1-x_1)^2$ (Rosenbrock function)	?	$\begin{bmatrix}0\\0\end{bmatrix}$	$\begin{bmatrix}2 & 0\\0 & 200\end{bmatrix}$
		$\begin{bmatrix}1\\1\end{bmatrix}$	$\begin{bmatrix}802 & -400\\-400 & 200\end{bmatrix}$

Submission deadline: February 13, 2019

Periférico Sur 8585
45604 Tlaquepaque, Jal., México
Tel +52 33 3669 3598
www.iteso.mx