

**Universidad de Guanajuato**  
**F.I.M.E.E.**  
**Laboratorio de Cálculo I**  
 Prof. Ing. Daniel Arturo Razo Montes  
 Práctica 4: Cómputo de valores Extremos de Funciones

---

## I. Introducción

En esta práctica se verá como calcular los valores extremos (máximos y mínimos) de una función usando el *toolbox* de matemática simbólica de MatLab.

## II. Desarrollo

Teclée los siguientes listados en su editor de archivos .m. Las salidas para cada uno de los listados se verán en la ventana de comandos (*Command Window*). No olvide teclear previamente el listado A de la práctica 3 al inicio de cada programa de ésta práctica.

### Listado 1

```
v = []; % variable auxiliar
% ----- Ejemplo 1 -----
y_11 = x^4+4*x^3-2*x^2-12*x; % funcion dada (simbolica)
disp('----- Ejemplo 1 -----');
disp('y_1(x) = ')
pretty(y_11)
disp('derivada de la funcion y_1(x) = ')
der_1 = diff(y_11,x); % derivada de una funcion
pretty(der_1)
disp('valores criticos de y_1(x) = ')
vc_1 = solve(der_1); % valores criticos, solve, hace igual a 0 la
% expresion en su argumento

pretty(vc_1)
v = vc_1;
y_12 = v.^4+4*v.^3-2*v.^2-12*v; % funcion dada, igual que la simbolica,
% se utilizara como auxiliar para evaluar
% la funcion en sus valores criticos

disp('-----');
figure('Name','Ejemplo 1')
subplot(1,2,1) % varios ejes coordenados en una misma figura
ezplot(y_11)
hold on
plot(double(vc_1(1)),double(y_12(1)),'rx') % marcas en la grafica para
plot(double(vc_1(2)),double(y_12(2)),'rs') % los valores criticos
plot(double(vc_1(3)),double(y_12(3)),'rd')
title('y_{1} = x^4+4*x^3-2*x^2-12*x')
grid
subplot(1,2,2)
ezplot(der_1)
title('y\prime_{1}(x)')
grid
% -----
```

### Salida del Listado 1:

```
----- Ejemplo 1 -----
y_1(x) =

          4          3          2
         x  + 4 x  - 2 x  - 12 x

derivada de la funcion y_1(x) =

          3          2
         4 x  + 12 x  - 4 x - 12
```

valores criticos de  $y_1(x) =$

[-1 1 -3]

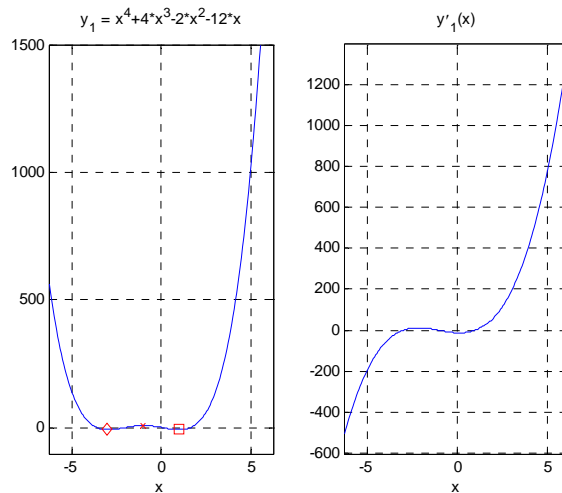


Figura 1. Gráfica de salida del listado 1

## Listado 2

```
% ----- Ejemplo 2 -----
y_21 = (w^3-3*w^2+4)^(1/3); % funcion dada (simbolica)
disp('----- Ejemplo 2 -----');
disp('y_2(w) = ')
pretty(y_21)
disp('derivada de la funcion y_2(w) = ')
der_2 = diff(y_21,w); % derivada de una funcion
pretty(der_2)
[num_1 ans] = numden (der_2);% regresa el numerador de una expresion
[ans den_1] = numden (der_2);% regresa el denominador de una funcion
disp('valores criticos de y_2(w) = ')
vc_21 = solve(num_1,'w'); % valores criticos, solve, hace igual a 0 la
% expresion en su argumento
vc_22 = solve(den_1,'w'); % valores criticos, solve, hace igual a 0 la
% expresion en su argumento

pretty(vc_21)
pretty(vc_22)
v = [vc_21' vc_22'];
y_22 = (v.^3-3*v.^2+4).^(1/3); % funcion dada, igual que la simbolica,
% se utilizara como auxiliar para evaluar
% la funcion en sus valores criticos

disp('-----');
figure('Name','Ejemplo 2')
subplot(1,2,1) % varios ejes coordenados en una misma figura
ezplot(y_21)
hold on
plot(double(vc_21(1)),double(y_22(1)),'rd') % marcas en la grafica para
plot(double(vc_21(2)),double(y_22(2)),'rs') % los valores criticos
plot(double(vc_22(1)),double(y_22(3)),'rd')
title('y_{2} = (w^3-3*w^2+4)^(1/3)')
grid
subplot(1,2,2)
ezplot(der_2)
title('y\prime_{2}(w)')
grid
% -----
```

## Salida del listado 2:

----- Ejemplo 2 -----

y\_2(w) =

$$(w^3 - 3w^2 + 4)^{1/3}$$

derivada de la funcion y\_2(w) =

$$\frac{1}{3} \frac{2}{(w^3 - 3w^2 + 4)^{2/3}}$$

valores criticos de y\_2(w) =

$$[0 \quad 2]$$

$$[-1 \quad 2 \quad 2]$$

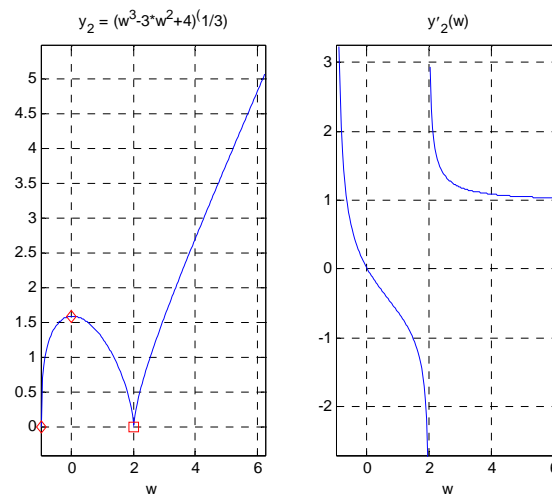


Figura 2. Gráfica de salida del listado 2

### Listado 3

```
% ----- Ejemplo 3 -----
y_31 = (x^2+2*x+5)/(x-1); % funcion dada (simbolica)
disp('----- Ejemplo 3 -----');
disp('y_3(x) = ')
pretty(y_31)
disp('derivada de la funcion y_3(x) = ')
der_3 = diff(y_31,x); % derivada de una funcion
der_3 = simplify(der_3);
pretty(der_3)
disp('valores criticos de y_3(x) = ')
vc_3 = solve(der_3); % valores criticos, solve, hace igual a 0 la
% expresion en su argumento

pretty(vc_3)
v = vc_3;
% v = vpa(v);
y_32 = (v.^2+2*v+5)./(v-1); % funcion dada, igual que la simbolica,
% se utilizara como auxiliar para evaluar
% la funcion en sus valores criticos

disp('-----');
figure('Name','Ejemplo 3')
subplot(1,2,1) % varios ejes coordenados en una misma figura
ezplot(y_31)
hold on
plot(double(vc_3(1)),double(y_32(1)),'rd') % marcas en la grafica para
plot(double(vc_3(2)),double(y_32(2)),'rs') % los valores criticos
title('y_{3} = (x^2+2*x+5)/(x-1)')
grid
```

```

subplot(1,2,2)
ezplot(der_3)
title('y\prime_{3}(x)')
grid
% -----

```

### Salida del listado 3:

```

----- Ejemplo 3 -----
y_3(x) =

      2
      x  + 2 x + 5
      -----
      x - 1

derivada de la funcion y_3(x) =

      2
      x  - 7 - 2 x
      -----
      2
      (x - 1)

valores criticos de y_3(x) =

      [      1/2      1/2]
      [1 + 2 2      1 - 2 2 ]
-----

```

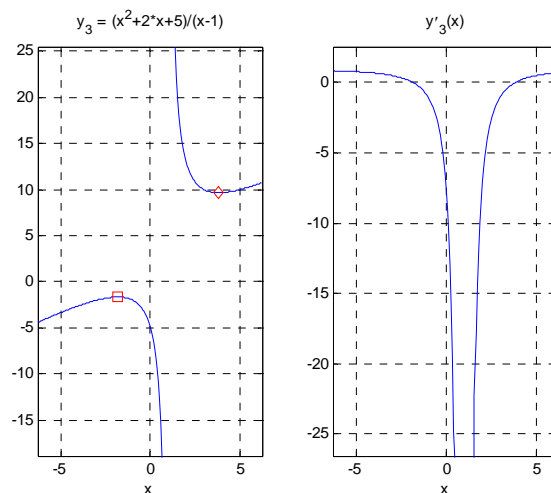


Figura 3. Gráfica de salida del listado 3

### Listado 4

```

% ----- Ejemplo 4 -----
y_41 = sin(2*x) + cos(2*x); % funcion dada (simbolica)
disp('----- Ejemplo 4 -----');
disp('y_4(x) = ')
pretty(y_41)
disp('derivada de la funcion y_4(x) = ')
der_4 = diff(y_41,x); % derivada de una funcion
der_4 = simplify(der_4);
pretty(der_4)
disp('valores criticos de y_4(x) = ')
vc_4 = solve(der_4'); % valores criticos, solve, hace igual a 0 la
% expresion en su argumento

pretty(vc_4')
v = vc_4;
% v = vpa(v);
y_42 = sin(2*v) + cos(2*v); % funcion dada, igual que la simbolica,
% se utilizara como auxiliar para evaluar

```

```

                                % la funcion en sus valores criticos
disp('----- Ejemplo 4 -----');
figure('Name','Ejemplo 4')
subplot(1,2,1)                    % varios ejes coordenados en una misma figura
ezplot(y_41)
hold on
for n = -10:2:10
    plot(double(vc_4(1))-pi*(n-1)/2,double(-y_42(1)),'rd')
    plot(double(vc_4(1))+pi*n/2,double(y_42(1)),'rs')
end
title('y_{4} = sin(2*x) + cos(2*x)')
grid
subplot(1,2,2)
ezplot(der_4)
title('y\prime_{4}(x)')
grid
% -----

```

#### Salida del listado 4:

```

----- Ejemplo 4 -----
y_4(x) =

                sin(2 x) + cos(2 x)
derivada de la funcion y_4(x) =

                2 cos(2 x) - 2 sin(2 x)
valores criticos de y_4(x) =

                [1/8 pi   - 3/8 pi]
-----

```

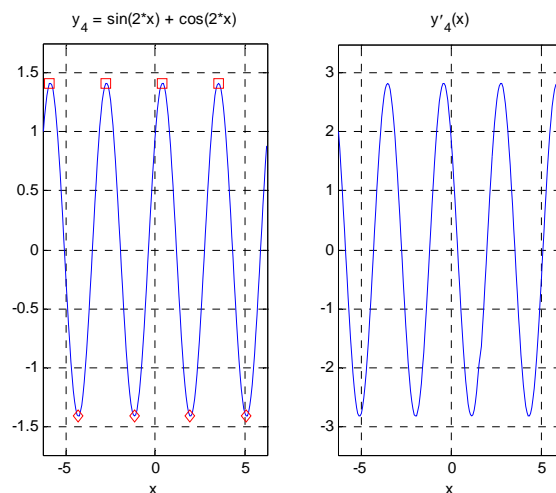


Figura 4. Gráfica de salida del listado 4

#### Listado 5

```

% ----- Ejemplo 5 -----
y_51 = abs(4-x^2);                % funcion dada (simbolica)
disp('----- Ejemplo 5 -----');
disp('y_5(x) = ')
pretty(y_51)
disp('derivada de la funcion y_5(x) = ')
der_5 = diff(y_51,x);             % derivada de una funcion
der_5 = simplify(der_5);
pretty(der_5)
disp('valores criticos de y_5(x) = ')
vc_51 = solve(der_5);            % valores criticos, solve, hace igual a 0 la
vc_52 = solve('4-x^2');          % expresion en su argumento

```

```

vc_5 = [vc_51' vc_52'];
pretty(vc_5)
v = vc_5;
% v = vpa(v);
y_52 = abs(4-v.^2);           % funcion dada, igual que la simbolica,
                              % se utilizara como auxiliar para evaluar
                              % la funcion en sus valores criticos

disp('-----');
figure('Name','Ejemplo 5')
subplot(1,2,1)               % varios ejes coordenados en una misma figura
ezplot(y_51)
hold on
plot(double(vc_5(1)),double(y_52(1)),'rs') % marcas en la grafica para
plot(double(vc_5(2)),double(y_52(2)),'rs') % los valores criticos
plot(double(vc_5(3)),double(y_52(2)),'rs')
title('y_{5} = abs(4-x^2)')
grid
subplot(1,2,2)
% ezplot(der_5)
ezplot('2*x*abs(-4+x^2)')
title('y\prime_{5}(x)')
grid
% -----

```

Salida del listado 5:

```

----- Ejemplo 5 -----
y_5(x) =

                2
            | -4 + x |

derivada de la funcion y_5(x) =

                2
            2 abs(1, -4 + x ) x

valores criticos de y_5(x) =

            [0    -2    2]
-----

```

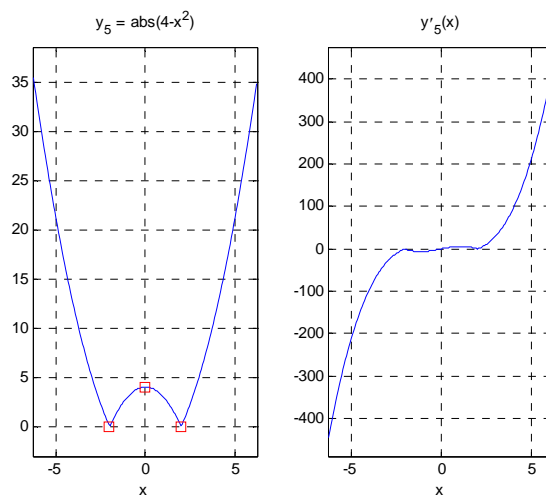


Figura 5. Gráfica de salida del listado 5

### III. Ejercicios

Encuentre los valores críticos de la función dada

1.  $f(x) = x^3 + x - 2$
2.  $f(x) = x^4 - 4x^3 + 7$

$$3. f(x) = \frac{x}{x^2 + 2}$$

$$4. f(x) = x^{2/3} + x$$

$$5. f(x) = \frac{x+4}{\sqrt[3]{x+1}}$$

$$6. f(x) = \cos 4x$$

Encuentre los extremos absolutos de la función dada en el intervalo indicado

$$7. f(x) = (x-1)^2; \quad [2,5]$$

$$8. f(x) = x^{2/3}(x^2 - 1); \quad [-1,1]$$

#### Ayuda.

El comando `solve()` por defecto iguala a cero la expresión algebraica contenida en su argumento, a menos que se especifique otra igualación por ejemplo `solve('2*x + 4 = 10')`, lo cual es lo mismo que `solve('2*x + 4 - 10')`. Puede ver que más se puede hacer con éste comando tecleando desde la ventana de comandos `help solve`. El comando `double()` convierte el formato del número en su argumento a formato `double()`, en esta práctica se convirtió una variable con formato simbólico (`sym`) a doble (`double`), debido a que el comando `plot()` no acepta variables del tipo simbólico (`sym`). Puede ver como funciona el comando `numden()` tecleando en la ventana de comandos `help numden`. En MatLab el operador `*` considera la multiplicación como matricial, lo mismo sucede para los operadores `/`, `^`, para realizar una multiplicación escalar y en general una división o elevar a una potencia (elemento a elemento) se escribe un punto antes del operador, por ejemplo `.*`, `./`, `.^`.

#### IV. Escriba sus conclusiones y observaciones

Nota: Reporte únicamente los ejercicios (sección III), poniendo el código fuente y los resultados de salida para cada uno (gráficas y/o expresiones algebraicas). Escriba sus conclusiones generales sobre la práctica.