# Funciones del MATLAB para resolver problemas de Optimización.

# Ejemplos con aplicaciones en Finanzas

Maria Gulnara Baldoquin de la Peña
Universidad EAFIT

## Introducción

Este documento complementa un folleto previo sobre el uso del MATLAB como herramienta para resolver problemas de la Matemática.

En particular se abordan aquellas funciones para resolver diferentes problemas de Optimización, desde la optimización de funciones de una variable al caso de varias variables, con o sin restricciones, lineales o no.

Se trata de destacar además cómo la Matemática es necesaria para complementar el uso de esta herramienta.

Por ejemplo, en muchos casos las funciones son multimodales, o sea, tienen varios óptimos locales, o se busca un óptimo condicionado por determinadas restricciones, donde debe dar un intervalo o punto inicial. Además, el software solo le garantiza encontrar óptimos locales. Los conocimientos recibidos de la Matemática le permiten tomar decisiones de manera conjunta con el uso del software.

Para cada problema puede usar una o combinación de varios comandos, en función del problema a resolver. Por ejemplo, en ocasiones es adecuado combinar con gráficos adecuados.

Otro aspecto importante es saber modelar el problema real planteado. El MATLAB, como otros softwares que ayudan a resolver problemas de Optimización, requieren el modelo matemático que representa el problema, y ese modelo lo debe introducir el usuario. Lo que resuelve el software es el modelo, no el problema. Si el modelo es incorrecto, el problema no se resolverá correctamente.

**Casos a considerar:**
1. Funciones de una variable, en un intervalo dado (**Comando fminbnd, Página 3**).
2. Funciones de varias variables, sin restricciones (**Comando fminsearch, Página 6**).
3. Funciones de varias variables, con restricciones, donde la función a optimizar (llamada función objetivo) y las restricciones son lineales (**Comandos linprog, intlinprog, Página 8**).
4. Funciones de varias variables, con restricciones, donde al menos la función objetivo o alguna restricción es no lineal (**Comando fmincon, Página 11**).
5. Considerar datos tomados de bases de datos en Excel (**Comando xlsread, Página 15**).

**1. Funciones de una variable, en un intervalo dado.**
Algunas sintaxis posibles:
x = **fminbnd**(f,a,b)
[x fval] = **fminbnd**(f,a,b)

Retorna en x un valor **mínimo local** de la función f, que es continua en el intervalo [a, b]. En el segundo caso asigna en fval el valor de la función f en el valor óptimo encontrado x.

Observaciones:
1. Los comandos del MATLAB para encontrar óptimos considera **SOLO PROBLEMAS DE MINIMIZAR**. En el caso de problemas de maximizar debe convertir el problema a uno de minimizar, multiplicando por -1 la función a optimizar.
2. La convergencia al valor óptimo es muy lenta cuando el óptimo está muy cercano a uno de los puntos frontera del intervalo (a o b).
3. Los métodos numéricos en los cuales se basa esta función del MATLAB son el método de la sección áurea y una interpolación parabólica.
4. La función realmente necesita estar definida en el intervalo abierto (a,b), pues a menos que a y b estén muy próximos, nunca los considera en el proceso de la búsqueda del óptimo. Si el óptimo es a ó b, da como respuesta un punto interior a (a,b) muy cercano al mismo.

Posibles sintaxis para definir f:
1. Directamente en el comando, o previamente, iniciando la función con el símbolo @(x).
2. Generar con el editor de funciones dicha función, e invocarla en el comando con el nombre con le dio, precedida con el símbolo @.

**Ejemplo 1:**
La función de costos (en millones de pesos) de una producción que depende de la cantidad de artículos x fabricados (en centenas) es: $f(x)=x^3/3-4.5x^2+18x$. ¿Cuál es el número de artículos a fabricar que minimice el costo, si sabemos que la cantidad de artículos a producir no debe ser menor que 300 y por restricciones de materia prima no debe producirse más de 10000? ¿Cuál sería ese costo?

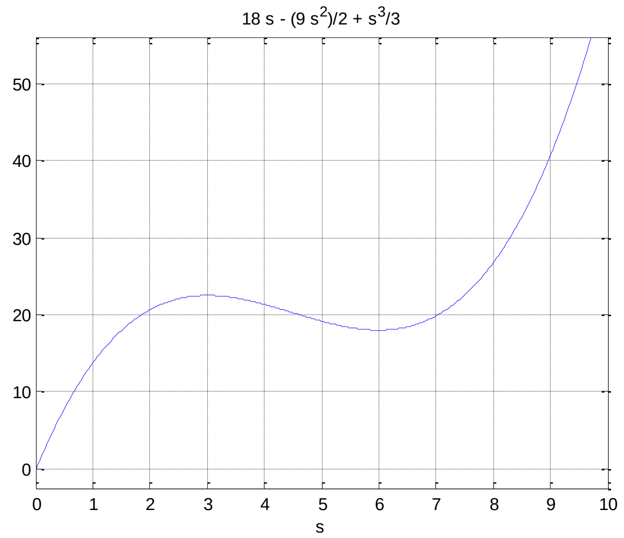
Una opción:
```
>> f=@(x)x^3/3-4.5*x^2+18*x;
>> [x fval]=fminbnd(f,3,100)
x = 6.0000
fval = 18.0000
```

O sea, sería producir 600 artículos y costaría $18 millones.

Observar en el gráfico de dicha función que lo obtenido es un óptimo local. Por ejemplo, para x=1 la función tiene un menor costo, pues f(1) = 13.83. Matemáticamente, lo puede comprobar con la función segunda derivada, con los siguientes comandos:

```
>> syms x
>> f=x^3/3-4.5*x^2+18*x;
>> fxx=diff(f,2)
```

fxx = 2*x – 9  (No es ni positiva ni negativa para todos los valores de x)




Otra opción:
Con el editor de funciones, generar la función que llamaremos ejemplo1:

```
function y=ejemplo1(x)
y=x^3/3-4.5*x^2+18*x;
end
```

Observar que es importante al escribir la función colocar un punto y coma al final, de lo contrario en la respuesta aparecerán todos los puntos que el método fue generando hasta obtener el óptimo propuesto.

Luego usar el comando:

```
>> [x fval]=fminbnd(@ejemplo1,3,10)
x = 6.0000
fval = 18.0000
```

**Ejemplo 2:**
La función de ganancia de una producción que depende de la cantidad de artículos x fabricados (en centenas) es: $f(x)=x^3/3-3x^2+5x$. ¿Cuál es el número de artículos a fabricar que maximice esa ganancia, si sabemos que la cantidad de artículos a producir no debe sobrepasar los 80? ¿Cuál sería esa ganancia?

```
>> f=@(x)-x^3/3+3*x^2-5*x;
>> [x fval]=fminbnd(f,0,8)
x = 1.0000
fval = -2.3333
```

O sea, f(1)= 2.3333
Observar que la función se multiplicó por -1, pues es un problema de maximizar.

En un problema real, ¿cómo decidir en qué intervalo buscar puntos de óptimo, si no tenemos información explícita al respecto? Es un problema similar al caso de funciones de varias variables donde debe darse un punto inicial de búsqueda, en lugar de un intervalo, ¿cómo elegirlo? Esta situación es importante cuando la función es multimodal, o sea, tiene varios puntos de extremo local, y el valor de la función a optimizar en ellos puede ser muy diferentes.

Hay varias posibilidades:

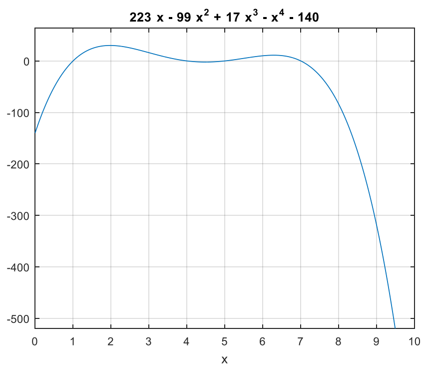
1. Graficar la función (caso de funciones de una variable) en intervalos donde usted sabe que puede "moverse" su variable. Por ejemplo, si representa unidades a producir y usted sabe que no puede producir más de 10000 artículos, y trabaja en miles de unidades, grafica en el intervalo (0,10).
2. Buscando los puntos críticos de la función, suponiendo que la función es derivable y su derivada continua, y graficando cerca de esos puntos.

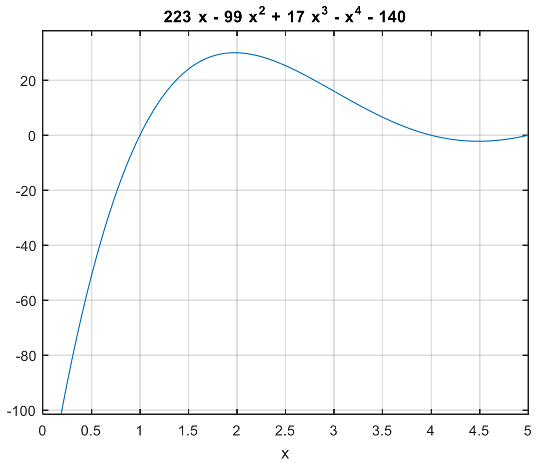
Veamos el siguiente ejemplo de una función multimodal:

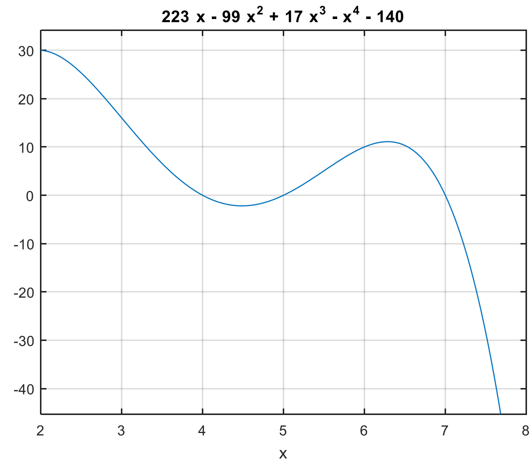
**Ejemplo 3**

Determinar el valor máximo de $f(x) = - x^4 + 17x^3 - 99x^2 + 223x – 140$

Observe el gráfico de la función f en el intervalo (0,10), y luego en los subintervalos (0,5) y (2,8). Se puede observar que en (0,10) la función tiene dos máximos locales y un mínimo local. Los máximos locales están cerca de 2 y de 6.




Usamos los comandos:

```
>> [x fval]=fminbnd(@(x)x^4 - 17*x^3 + 99*x^2 - 223*x +140,0,3)
x = 1.9765
fval = -30.0118
```

```
>> [x fval]=fminbnd(@(x)x^4 - 17*x^3 + 99*x^2 - 223*x +140,2,8)
x = 6.2873
fval = -11.0954
```

Observe que se multiplicó la función por -1 pues se está buscando un máximo. Luego los máximos locales encontrados son: x = 1.9765 y x = 6.2873, siendo el primero el máximo global pues el valor de la función objetivo es mayor, 30.0118.

¿Cómo podemos comprobar que para otros valores positivos, si son los que nos interesan, no hay otro máximo local, e incluso sea global?

Varios métodos:
1. Buscar los puntos críticos de f (de la función original), condición necesaria de óptimo, pues la función es un polinomio, o sea, es indefinidamente continua y derivable.

Con el MATLAB:

```
>> syms x
>> f= - x^4 + 17*x^3 - 99*x^2 + 223*x – 140;
>> h=double(solve(diff(f)))
h =
   1.9765
   4.4861
   6.2873
```

Efectivamente, solo tiene 3 puntos críticos, los vistos gráficamente de dos máximos locales y un mínimo local.

Observe que en un solo comando se hicieron tres operaciones, de adentro hacia afuera:
- ✓ Derivar f
- ✓ Resolver f '(x)=0
- ✓ Convertir las constantes simbólicas en numéricas.

2. Ver el signo de la función primera derivada (de la función original), que nos dice si crece (f '>0) o decrece (f '<0)
En este caso si deriva la función, el resultado será:
$f'(x) = -4x^3+51x^2-18x+223$

Puede comprobar lo que va viendo gráficamente, que la función a partir del punto de máximo local x=6.2873 empieza a decrecer, porque la función derivada tendrá signo negativo para valores mayores que este. Observe que $-4x^3$ tiene signo negativo para valores de x positivos, y el término $4x^3$ será mayor que el positivo $51x^2$, a partir de ciertos valores de x, por ejemplo, 6.2873, luego f '(x) será negativo, independiente de los otros términos que aparecen.

**2. Funciones de varias variables, sin restricciones.**
**La función fminsearch**

**Sintaxis posibles:**
x=fminsearch(f,x0)
[x fval]= fminsearch(f,x0)

En este caso, se debe introducir, además de la función a optimizar, un punto estimado donde el método utilizado comienza la búsqueda del óptimo.

Dicho punto es un vector que depende de la cantidad de variables del problema. Por ejemplo, en el caso de una función de dos variables, podría ser [0 0], en el caso de 3 variables, [0 0 0].

La función puede tener algunas discontinuidades, en particular si las mismas no ocurren cerca del óptimo buscado, que también será un mínimo local.
El método que usa el MATLAB es el método de Nelder-Mead o del Simplex (2), el cual no está basado en el uso de gradientes.

**Ejemplo 4:**
(Decisiones sobre fijación de precios, tomado de referencia (3))
La Corporación de cremas dentífricas orgánicas produce crema para dientes en dos tamaños, de 100 y 150 mililitros. El costo de producción de cada tubo de cada tamaño es de 60¢ y 90¢, respectivamente. Las demandas semanales x1 y x2 (en miles) para los dos tamaños son de
x1=3(p2-p1), x2=320+3p1-5p2, donde p1 y p2 son los precios en centavos de los tubos. Determine los precios p1 y p2 que maximizarían las utilidades de la compañía.

Puede comprobar que la función de utilidad en función de los precios p1 y p2 será:
$P = -3p1^2-5p2^2+6p1p2-90p1+590p2-28800$

Introducimos los comandos:
```
>> P=@(x)3*x(1)^2+5*x(2)^2-6*x(1)*x(2)+90*x(1)-590*x(2)+28800;
>> [X fval]=fminsearch(P,[0 0])
X = 110.0000  125.0000
fval  = -3.1250e+03
```

O sea, p1=110¢, p2=125¢

Observe que:
1. Al introducir P, x es un vector, cuyas componentes son x(1), x(2), siendo la primera p1 y la segunda p2.
Si tiene 3 variables, sean x,y,z, entonces la x que aparece en @(x) tendría las componentes x(1), x(2), x(3).

2. La función P se multiplicó por -1 porque era un problema de maximizar, por lo que la utilidad máxima es de $3.1250x10^3$, es decir, 3125 (en miles de centavos).

Con estos valores de p1 y p2, las demandas son de x1=45 y x2=25 (en miles)

3. ¿Los valores óptimos obtenidos hubieran sido diferentes si el punto inicial fuera otro? En este caso no, porque la función es cóncava, y por tanto tiene un único punto extremo que es un máximo.

Con el MATLAB podríamos comprobarlo hallando el hessiano de P (acá es la original) y ver que los menores impares son negativos y el único menor par, que es el propio determinante de la matriz hessiana, es positivo.

```
syms  x y
>> P = -3*x^2-5*y^2+6*x*y-90*x+590*y-28800;
>> H=hessian(P)
H =
[ -6,   6]
[  6, -10]
```

```
>> d=det(H)
d = 24
```

**3. Funciones de varias variables, con restricciones, donde la función objetivo y las restricciones son lineales**.

Son los conocidos como problemas de Programación Lineal (PL).

El comando linprog resuelve problemas de Programación Lineal que pueden modelarse como
min f '(x)
Con restricciones del tipo
$Ax \leq b$ (1)
$A_1x = b_1$ (2)
$x \geq lb$ (3)
$x \leq ub$ (4)

donde lb, ub son cotas inferiores (lb) y superiores (ub) de las variables.

Observe que:

1. x es un vector, con tantas componentes como variables tiene el modelo.
2. Puesto que el sistema es lineal, solo se introducen matrices y vectores. f será el vector de los coeficientes de las variables en la función objetivo, si el problema es de minimizar. Si es de maximizar es el vector opuesto a f, o sea, -f. A y $A_1$ son las matrices del sistema de restricciones, cuyo orden está dado por el número de restricciones de tipo (1) en A (de tipo (2) en $A_1$) y el número de variables. b y b1 son vectores columnas, que representan las partes derechas de las restricciones. lb y ub son vectores filas, con los valores de las cotas, si existen, para cada una de las variables.

En el caso de tener solo restricciones de tipo (1) la sintaxis a utilizar es:
x = linprog(f,A,b)   ó [x fval] = linprog(f,A,b)

En el caso de tener solo restricciones de tipo (2) la sintaxis a utilizar es:
x = linprog(f,A,b,Aeq,beq)   o [x fval] = linprog(f,A,b,Aeq,beq)
donde A, b se sustituyen por [], []

En el caso de tener restricciones que involucran cotas inferiores o superiores de las variables (de tipo (3) ó (4)) la sintaxis a utilizar es:

x = linprog(f,A,b,Aeq,beq,lb,ub)  o   [x fval] = linprog(f,A,b,Aeq,beq,lb,ub)

Si no se tienen restricciones del tipo (1) se sustituyen A, b por [],[].  Si no se tienen restricciones del tipo (2) se sustituyen Aeq, beq por [],[].

**Ejemplo 5**
Una microempresa de inversiones se dedica a administrar las carteras de acciones de varios clientes. Un nuevo cliente ha solicitado que la compañía se haga cargo de administrar para él una cartera de $100.000. A dicho cliente le interesa restringir la cartera mezclando solo 3 tipos de acciones, como se muestra en la siguiente tabla.

| ACCIONES | PRECIO POR ACCION ($) | RENDIMIENTO ANUAL POR ACCION ($) | INVERSIÓN MÁXIMA POSIBLE ($) |
|---|---|---|---|
| Gofer Crude | 60 | 7 | 60.000 |
| Can Oil | 25 | 3 | 25.000 |
| Sloth Petroleum | 20 | 3 | 30.000 |

A usted como asesor de la microempresa le solicitan proponer cuántas acciones de cada tipo ofertar para maximizar el rendimiento anual total estimado de esa cartera.

El modelo matemático para este problema sería:

Variables:
Sean x,y, z el # de acciones de Gofer Crude, Can Oil y Sltoh Petroleum, respectivamente

Entonces:
Max $f(x,y,z)= 7x+3y+3z$
s.a. $60x+25y+20z \leq 100000$
$60x\leq 60000$
$25y \leq 25000$
$20z \leq 30000$
$x,y,z \geq 0$

Es un problema de PL de maximizar, solo con restricciones de desigualdad, y cotas inferiores para todas las variables.

Utilizaríamos la sintaxis de linprog:

```
>> [x fval]= linprog([-7 -3 -3],[60 25 20;60 0 0;0 25 0;0 0 20],[100000;60000;25000;30000],[],[],[0 0 0],[])
Optimization terminated.
x =
   1.0e+03 *
    0.7500
    1.0000
    1.5000

fval = -1.2750e+04
```

En MATLAB el símbolo e+n es la base 10 elevado a la potencia n (si tiene signo – es a la –n). O sea, e+03 significa $10^3$.

También puede hacer:

```
f=[-7 -3 -3];
A=[60 25 20;60 0 0;0 25 0;0 0 20];
b=[100000;60000;25000;30000];
lb=[0 0 0];
```

y usar: [x fval] = linprog(f,A,b,[],[],lb,[])

O sea, el número de inversiones en Gofer Crude, Can Oil y Sltoh Petroleum deben ser de 750, 1000 y 1500 respectivamente, con un máximo rendimiento anual total de $12750.

**Ejemplo 6**

El siguiente ejemplo es característico de un tipo de problema donde un planificador tiene una función U (*x*, *y*) a optimizar, que representa el valor social del producto en la economía, y unas o varias ecuaciones lineales del tipo p*x* + q*y* = a (o p*x* + q*y* ≤ a), que representan restricciones de recursos en la capacidad de la economía para producir *x* y *y*.

max f(x,y)=4x+7y
s.a. x+3y=34
2x+y=18
x,y ≥0

En este problema de optimización lineal (función a optimizar y restricciones lineales) solo se tienen restricciones de igualdad, y cotas inferiores para las variables, luego la sintaxis a utilizar de linprog sería:

[x fval] = linprog(f,[],[],A,b,lb,[])

Resolviéndolo con el MATLAB:

```
>> f=[-4 -7];
>> lb=[0 0];
>> A=[1 3;2 1];
>> b=[34;18];
>> [x fval] = linprog(f,[],[],A,b,lb,[])
Optimization terminated.
x =
    4.0000
   10.0000
fval = -86.0000
```

Luego la solución es (4,10) con un valor de la función a optimizar de 86, para esos valores óptimos de x,y.

¿Qué pasaría si algunas variables (al menos una) debe ser entera?

El comando linprog no lo garantiza, para ello debe usar el comando **intlinprog**

Las sintaxis posibles son (o con [x fval] en la parte izquierda, donde se asignan los resultados):
x = intlinprog(f,intcon,A,b)
x = intlinprog(f,intcon,A,b,Aeq,beq)
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)

intcon es un vector que tiene los números que representan aquellas variables que deben ser enteras.
Por ejemplo, suponga que resuelve un problema de PL con 10 variables.
Si la segunda debe ser entera, entonces intcon=2.
Si las enteras deben ser la segunda y la quinta, entonces intcon = [2 5].
Si deben ser todas las variables enteras, intcon = [1 2 3 4 5 6 7 8 9 10].

**Ejemplo 7**
Un taller tiene tres tipos de máquinas A, B y C y puede fabricar dos productos. Todos los productos tienen que ir a cada máquina y cada uno va en el mismo orden: Primero a la máquina A, luego a la B y luego a la C. La Tabla muestra:
1. Las horas requeridas en cada máquina, por unidad de producto.

2. Las horas totales disponibles para cada máquina, por semana.
3. La ganancia por unidad vendida de cada producto.

| Tipo de máquina | Producto 1 | Producto 2 | Horas disp. x semana |
|---|---|---|---|
| A | 2 | 1 | 19 |
| B | 2 | 3 | 42 |
| C | 3 | 1 | 24 |
| Ganancia | 3 | 2 | |

¿Qué cantidad de cada producto se debe manufacturar cada semana, para obtener la máxima ganancia?

Compruebe que el modelo de PL asociado este problema es:

Variables:
$x$: Cantidad de unidades a producir del producto A semanalmente
$y$: Cantidad de unidades a producir del producto B semanalmente

max $3x + 2y$
s.a. $2x + y \leq 19$
$2x + 3y \leq 42$
$3x + y \leq 24$
$x, y \geq 0$, enteras

Los datos serían:
f=[-3 -2]; A=[2 1;2 3;3 1]; b=[19;42;24]; lb=[0 0];

Si usáramos el comando linprog el resultado sería:

```
>> [x fval] = linprog(f,A,b,[],[],lb,[])
x =
    3.7500
   11.5000
fval = -34.2500
```

Las cantidades de producto a producir semanalmente dan fraccionarias, solución no factible al problema.

Nota: No trate de redondear, es peligroso, a veces el redondeo o da soluciones infactibles, o una lejana a la óptima. Por ejemplo, en este caso si redondea a x=4, y=12, la primera restricción no se cumple, pues 2(4)+12=20>19

Usando el comando intlinprog:

```
>> intcon=[1 2];
>> [x fval] = intlinprog(f,intcon,A,b,[],[],lb,[])
x =
    4.0000
   11.0000
fval = -34
```

Observe que dio una solución factible, (4,11) y que el valor de la función a optimizar fue ligeramente inferior al obtenido si las variables no requirieran ser enteras (34 contra 34.25) a costa de lograr la condición de que las variables sean enteras.

**4. Funciones de varias variables, con restricciones, donde al menos la función objetivo o alguna restricción es no lineal.**

En este caso se usa la función fmincon del MATLAB.
Las posibles sintaxis son (o con [x fval] en la parte izquierda, donde se asignan los resultados):

x = fmincon(f,x0,A,b)
x = fmincon(f,x0,A,b,Aeq,beq)
x = fmincon(f,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(f,x0,A,b,Aeq,beq,lb,ub,nonlcon)

Las diferencias con el comando linprog son:
Debe introducir la función a optimizar f explícitamente, como se hace con fminsearch.
Debe introducir un punto inicial x0 donde comenzar la búsqueda.
A,b,Aeq,Aeq aparecen si tiene restricciones lineales de desigualdad o igualdad. Si tiene restricciones no lineales las introduce explícitamente con el editor de funciones en una función llamada nonlcon, donde asume que las restricciones son de la forma g(x)<0 o g(x)=0, o ambas, y solo introduce las funciones g. En los ejemplos observará cómo distinguir las diferentes restricciones no lineales de igualdad y de desigualdad en una misma función.

Una observación importante: este comando tiene sentido usarlo cuando f es no lineal o al menos una de las restricciones es no lineal. También es para el caso de minimizar.

**Ejemplo 8**
Una compañía tiene 3 fábricas y todas elaboran el mismo producto. Las fábricas A, B y C producen x,y y z unidades del producto, respectivamente, a un costo de producción de $3x^2+200$ USD, $y^2+400$ USD y $2z^2+300$ USD, respectivamente. Si recibe un pedido exacto de 1100 unidades, determine cómo debe distribuirse la producción entre las 3 fábricas con el fin de minimizar el costo total.

El modelo matemático sería:
minimizar $C=(3x^2+200)+(y^2+400)+(2z^2+300)= 3x^2+y^2+2z^2+1100$

con las restricciones:
$x+y+z=1100$
$x, y, z \geq 0$

La sintaxis a utilizar sería:
```
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[0 0 0],[],[],[1 1 1],[1100],[0 0 0],[])
```

Local minimum found that satisfies the constraints.
x = 200.0000  600.0000  300.0000

fval = 6.6110e+05

O sea, las fábricas A, B y C deben producir 200, 600 y 300 unidades respectivamente, a un costo de 6.6110.105, es decir, $661.100

¿Pudiéramos detectar si ese mínimo es global, o es sensible al punto inicial que le dimos?

La función es convexa, (como se demuestra al buscar la matriz hessiana con el MATLAB, detallado al final del ejemplo), pero su mínimo es (0,0,0), y buscamos un mínimo condicionado, dado por x+y+z=1100. Luego no es garantía comprobar solo si la función es convexa.

Compruebe entonces al usar los siguientes comandos, donde solo se varía el punto inicial, no importa si satisface la restricción el punto inicial, que nos llevan todos al mismo resultado.

```
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[400 400 300],[],[],[1 1 1],[1100],[0 0 0],[])
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[400 300 400],[],[],[1 1 1],[1100],[0 0 0],[])
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[50 50 50],[],[],[1 1 1],[1100],[0 0
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[1100 0 0],[],[],[1 1 1],[1100],[0 0 0],[])
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[0 1100 0],[],[],[1 1 1],[1100],[0 0 0],[])
>> [x fval]= fmincon(@(x)3*x(1)^2+x(2)^2+2*x(3)^2+1100,[0 0 1100],[],[],[1 1 1],[1100],[0 0 0],[])
```

Comprobando que f es convexa (aunque no es lo que se requiere en este ejemplo).

```
>> syms x y z
>> f= 3*x^2+y^2+2*z^2+1100;
>> H=hessian(f)
H =
[ 6, 0, 0]
[ 0, 2, 0]
[ 0, 0, 4]

>> D=det(H)
D = 48
```

Por ser la matriz hessiana diagonal, es fácil ver que todos los menores principales son positivos, incluso el determinante de H, que en el caso de una matriz diagonal es el producto de la diagonal principal.

**Ejemplo 9**

Determinemos el valor mínimo de la siguiente función no lineal, sujeta a una restricción no lineal.

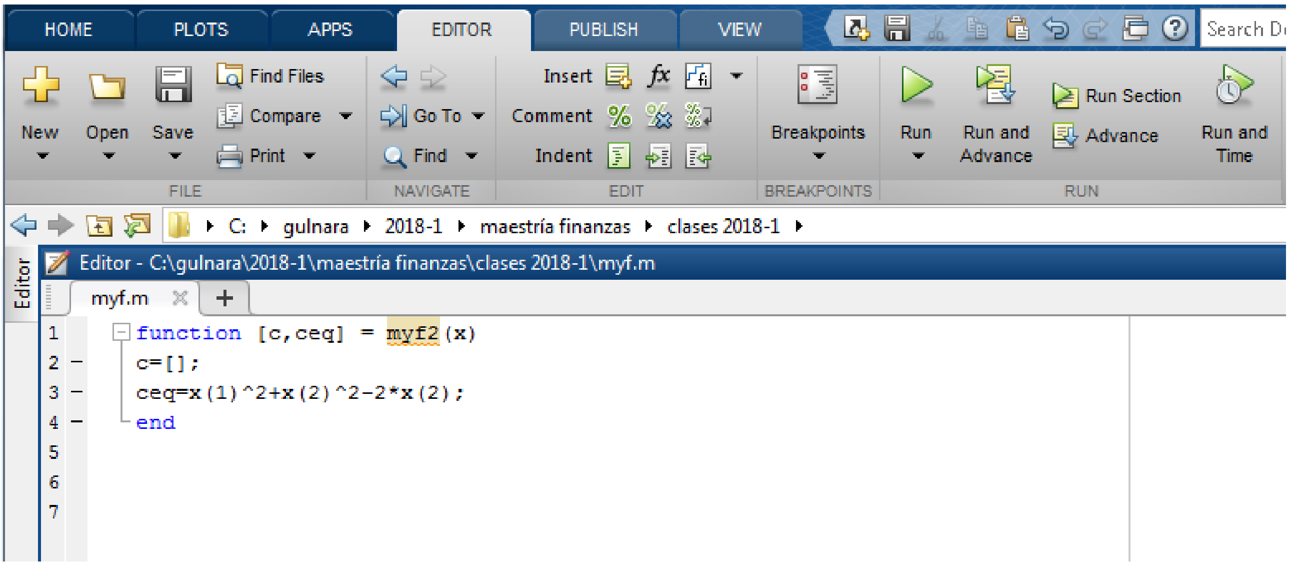
$f(x,y)=4x^2+2y^2+5$

s.a. $x^2+y^2-2y=0$

**Usamos el comando del MATLAB fmincon con la siguiente sintaxis:**

```
>> [X fval]=fmincon(@(x)4*x(1)^2+2*x(2)^2+5,[0 0],[],[],[],[],[],[],@myf)
```

Siendo myf definida en el editor de funciones como se grafica debajo:




En c se colocan las restricciones no lineales de desigualdad, en ceq las no lineales de igualdad. Observe que como no hay restricciones no lineales de desigualdad, se asigna [] a c.

Como no hay restricciones lineales de ningún tipo (ni desigualdad ni igualdad) ni cotas inferiores exigidas a las variables, se colocan 6 doble corchetes cerrados [].

Observe dos soluciones mínimos locales entregadas (0,0) y (0,2), ambas satisfaciendo las restricciones, con valores de la función a optimizar 0 y 13, dependiendo del punto inicial dado, en el primer caso, (0,0), en el segundo (0,20).

```
>> [X fval]=fmincon(@(x)4*x(1)^2+2*x(2)^2+5 ,[0 0], [], [], [], [], [], [], @myf)
Initial point is a local minimum that satisfies the constraints.
X = 0     0
fval = 5
>> [X fval]=fmincon(@(x)4*x(1)^2+2*x(2)^2+5, [0 20], [], [], [], [], [], [],@myf)
Local minimum found that satisfies the constraints.
X = -0.0000    2.0000
fval = 13.0000
```

¿Cómo podíamos tener idea de cuántos y qué posibles valores darle al punto inicial?

Buscar los puntos críticos de la función Lagrangiana correspondiente, usando los siguientes comandos del MATLAB:

```
>> syms x y l
>> L=4*x^2+2*y^2+5-l*(x^2+y^2-2*y);
>> Lx=diff(L,x)
Lx = 8*x - 2*l*x
>> Ly=diff(L,y)
Ly = 4*y - l*(2*y - 2)
>> Ll=diff(L,l)
Ll = - x^2 - y^2 + 2*y

>> S=solve('8*x - 2*l*x=0', '4*y - l*(2*y - 2)=0','- x^2 - y^2 + 2*y=0')
S =
   l: [2x1 sym]
   x: [2x1 sym]
   y: [2x1 sym]
```

```
>> S.x
ans =
 0
 0
>> S.y
ans =
 0
 2

>> S.l
ans =
 0
 4
```

Del resultado obtenemos que las posibilidades de mínimos locales son (0,0) y (0,2).

**Ejemplo 10**

En este ejemplo veamos cómo se introducen restricciones no lineales cuando hay más de una de desigualdad (o de igualdad, es el mismo proceso)

Min $f(x,y)= (x-4)^2+(y-4)^2$
s.a. $(x-1)^2+ y^2 \leq 1$
$x^2+ (y -1)^2 \leq 1$
$x,y \geq 0$

Observar que tanto la función a optimizar como las restricciones son no lineales, siendo ambas restricciones del tipo ≤, y además hay cotas inferiores para las variables, luego la sintaxis es:

```
>> [X fval]=fmincon(@(x)(x(1)-4)^2+(x(2)-4)^2,[0 0],[],[],[],[],[0 0],[],@myf1)
```

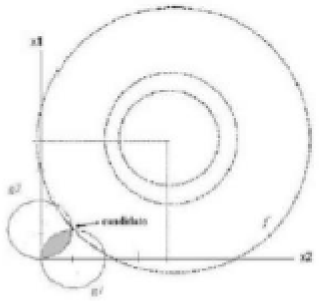
Local minimum found that satisfies the constraints.
X = 1.0000    1.0000
fval = 18.0000

¿Cómo se declara myf1?

```
function [c,ceq] = myf1(x)
c=[(x(1)-1)^2+x(2)^2-1,x(1)^2+(x(2)-1)^2-1];
ceq=[];
end
```

Observar que como hay dos restricciones de desigualdad, en c se declara un vector de dos componentes, donde en cada uno se escribe cada una de las restricciones, previamente colocadas en la forma $g(x1,x2)\leq 0$, y solo se introduce  $g(x1,x2)$.

Compruebe con diferentes puntos iniciales que es el único óptimo, y es global, pudiendo verlo en esta representación gráfica.




**5. Considerar datos tomados de bases de datos en Excel.**
Suponga que los datos los tiene archivados en tablas de Excel.

El comando xlsread del MATLAB le permite leer los datos de la misma. Los formatos más simples serían:
num = xlsread(filename)
num = xlsread(filename,sheet)

En ambos casos lee el documento de Excel con el nombre filename y lo escribe en la variable num de MATLAB. Si tiene varias hojas en el fichero y quiere referirse a una en particular, use la opción 2 colocando el número de esa hoja (denotado sheet).

Veamos el siguiente ejemplo de Optimización.

**Ejemplo 11**
Una compañía de inversiones tiene actualmente $10 millones de pesos disponibles para la inversión. La meta que trazó es maximizar la retribución esperada durante el siguiente año. La compañía requiere además que cuando menos 30% de los fondos tendrán que colocarse en acciones ordinarias y bonos de tesorería y que no más del 40% del dinero deberá invertirse en fondos de mercado y títulos municipales. ¿Cuánto dinero tendrá que invertir en cada instrumento? Tenga en cuenta los siguientes datos que dispone.

| Posibilidad de inversión | Retribución esperada (%) | Inversión máxima (millones de $) |
|---|---|---|
| Bonos de tesorería | 8 | 5 |
| Acciones ordinarias | 6 | 7 |
| Mercado de dinero | 12 | 2 |
| Títulos municipales | 9 | 4 |

Para responder con el comando adecuado de MATLAB, debe primero plantear el modelo matemático, que sería:

Variables:
x1: fondos en bonos de tesorería (en millones de dólares)
x2: fondos en acciones ordinarias (en millones de dólares)
x3: fondos en mercado de dinero (en millones de dólares)
x4: fondos en títulos municipales (en millones de dólares)

Función a optimizar:
Maximizar $z = 0.08x1+0.06x2+0.12x3+0.09x4$

Restricciones:
$x1+x2+x3+x4 \leq 10$

x1+x2 ≥ 3
x3+x4 ≤ 4
x1 ≤ 5
x2 ≤ 7
x3 ≤ 2
x4 ≤ 4
x1, x2, x3, x4 ≥ 0

Para buscar el comando apropiado: El modelo es lineal, tanto la función a optimizar como las restricciones. No exige variables enteras, luego sería linprog.

No tiene restricciones de igualdad, pero la segunda restricción de desigualdad es ≥, por lo que debe multiplicarse por -1 para que sea como las demás del tipo ≤.

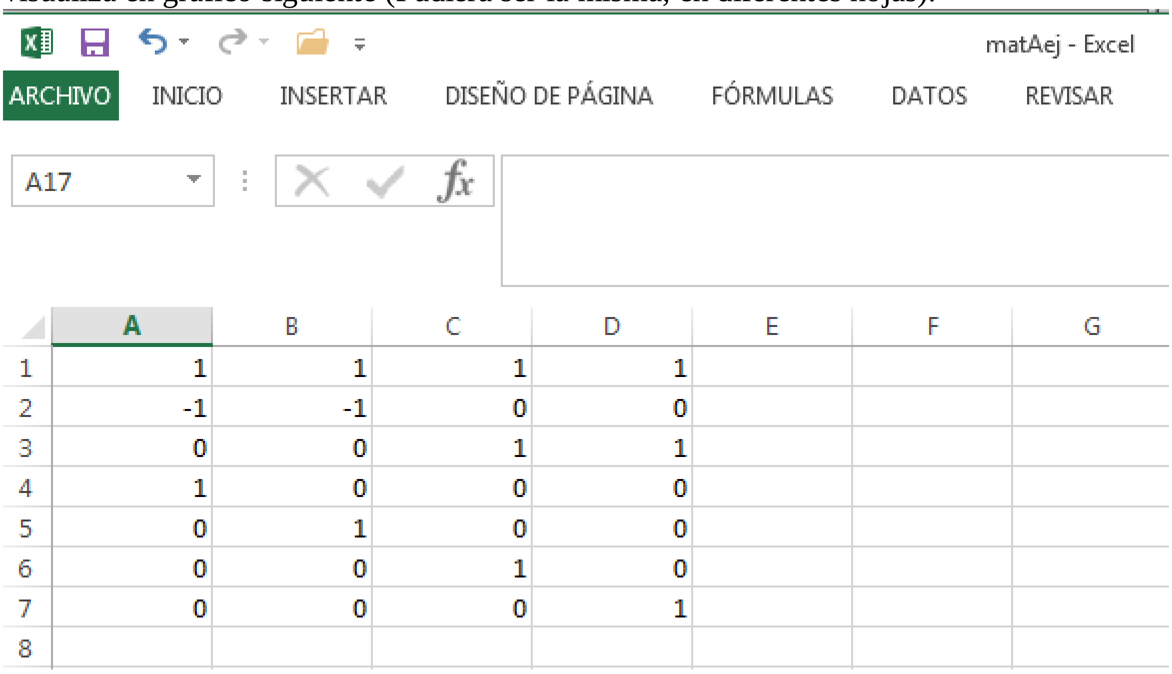
El problema es de maximizar, luego la función objetivo se multiplica por -1.

Hay restricciones de cotas inferiores para las variables (todas cero).

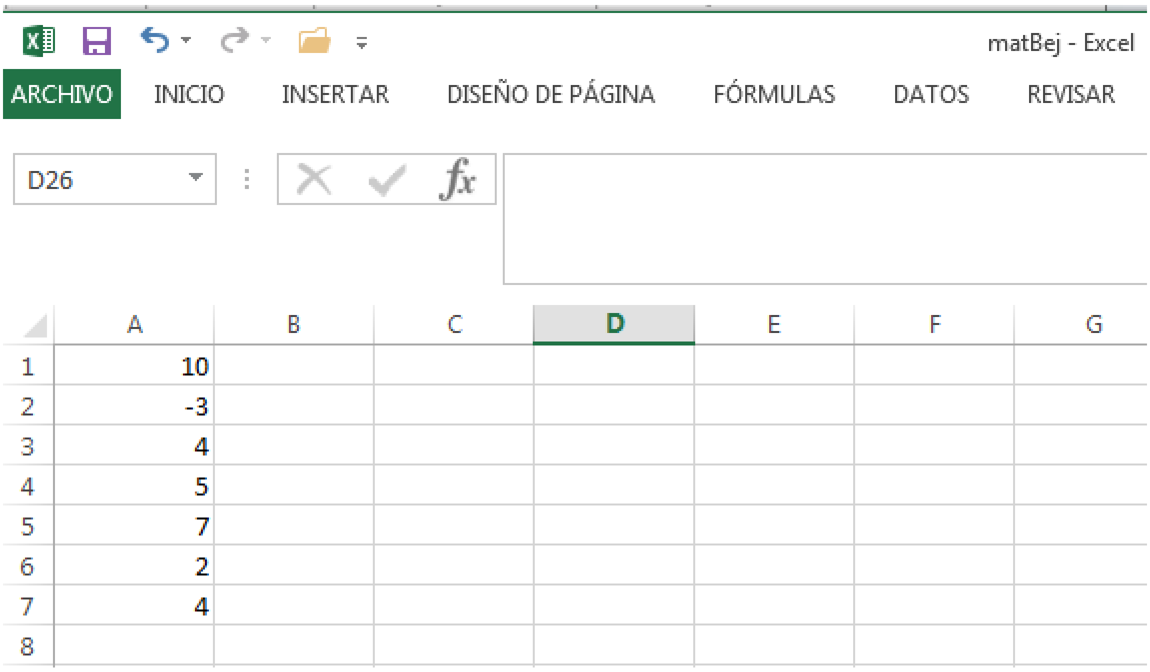
Luego la sintaxis apropiada sería:
[x fval]= linprog(-f,A,b,[],[],lb,[])

donde
f=[0.08 0.06 0.12 0.09];
lb=[0 0 0 0];

Supongamos que A y b están en dos tablas de Excel llamadas matAej.xlsx y matBej.xlsx, como se visualiza en gráfico siguiente (Pudiera ser la misma, en diferentes hojas).




| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | | | |
| 2 | -1 | -1 | 0 | 0 | | | |
| 3 | 0 | 0 | 1 | 1 | | | |
| 4 | 1 | 0 | 0 | 0 | | | |
| 5 | 0 | 1 | 0 | 0 | | | |
| 6 | 0 | 0 | 1 | 0 | | | |
| 7 | 0 | 0 | 0 | 1 | | | |
| 8 | | | | | | | |




Entonces se escribe en MATLAB la siguiente secuencia de comandos (el nombre del documento en .xlsx debe escribirse entre los símbolos ' '):

```
>> A=xlsread('matAej.xlsx');
>> b=xlsread('matBej.xlsx');
>> C=[0.08 0.06 0.12 0.09];
>> lb=[0  0  0  0];
>> [x fval]= linprog(-C,A,b,[],[],lb,[])
Optimization terminated.
x =
    5.0000
    1.0000
    2.0000
    2.0000
fval = -0.8800
```

Luego la respuesta es:
x1=5, x2=1, x3= 2, x4=2 y el valor de la retribución esperada será de 0.88 (en millones de dólares)

**Algunas referencias bibliográficas:**

1. Help del MATLAB.
2. Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *SIAM Journal of Optimization*, Vol. 9 Number 1, pp. 112-147, 1998.
3. Arya, J.C., Lardner, R.W. Matemáticas aplicada a la administración y economía, Quinta Edición, Prentice Hall, Copyright © 1993. ISBN 0-13-564287-6.