Examples of Using *fmincon* in MATLAB

fmincon finds a minimum of a constrained nonlinear multivariable function, and by default is based on the SQP (Sequential Quadratic Programming) algorithm.

fmincon solves problems of the form:

$$\min_x f(x) \text{ subject to:}$$

$$\left.\begin{array}{l} A*x \le B \\ Aeq*x = Beq \end{array}\right\} \text{ linear constraints}$$

$$\left.\begin{array}{l} C(x) \le 0 \\ Ceq(x) = 0 \end{array}\right\} \text{ (nonlinear constraints)}$$

$$LB \le x \le UB \quad \rightarrow \quad \text{bounding of variables}$$

Common Syntax:

x = fmincon(fun, x0, A, b)

x = fmincon(fun, x0, A, b, Aeq, Beq)

x = fmincon(fun, x0, A, b, Aeq, Beq, lb, ub)

x = fmincon(fun, x0, A, b, Aeq, Beq, lb, ub, nonlcon)

[x, fval, exitflag, output, lambda] = fmincon(fun, x0, A, b, Aeq, Beq, lb, ub, nonlcon)

where **x0** is the initial guess for the minimum, **nonlcon** is an M-file function call for the nonlinear constraints and defines **C(x)** and **Ceq(x)** . Note that if any argument is not present, it should be set to a null set []. If **lb** and **ub** are not known, they should be set to null sets as well. **Lambda** is the Lagrange multipliers.

Example 1: **Minimize** $\mathbf{f(x) = -x_1x_2x_3}$

s.t. $\mathbf{0 \le x_1 + 2x_2 + 2x_3 \le 72}$

$\mathbf{x_0} = [10 \;\; 10 \;\; 10]^T$

Rewrite the linear inequality constraints as:

$\mathbf{-x_1 - 2x_2 - 2x_3 \le 0}$ **and** $\mathbf{x_1 + 2x_2 + 2x_3 \le 72}$

Write an M-file for **f(x)**:

```
function f = myfun(x)
f = –x(1)*x(2)*x(3);
```

The MATLAB input is:

```
>> A = [–1 –2 –2; 1 2 2];
>> b = [0; 72];
>> x0 = [10; 10; 10];
>> [x, fval] = fmincon(@myfun, x0, A, b)

x =

   24.0000    12.0000    12.0000

fval =

  –3.456e+03
```

Example 2: **Minimize** $\mathbf{f(x) = e^{x_1}[4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1]}$

s.t. $\mathbf{1.5 + x_1x_2 - x_1 - x_2 \leq 0}$

$\mathbf{-x_1x_2 \leq 10}$

$\mathbf{x_0} = [-1\ -1]^T$

Write an M-file for the objective function:

```
function f = objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

Write an M-file for the nonlinear constraints:

```
function [c, ceq] = confun(x)
c = [1.5+x(1)*x(2) –x(1) –x(2); –x(1)*x(2) –10];
ceq = [];
```

The MATLAB input is:

```
>> x0 = [–1; –1];
>> [x, fval] = fmincon(@objfun, x0, [], [], [], [], [], [], @confun)

x =
```

```
    −9.5474    1.0474

fval =

    0.0236
```