# Dowling05.wxmx Multivariable Functions in Economics Part I

TABLE OF CONTENTS



```
load(draw)$  set_draw_defaults(line_width=2, grid = [2,2], point_type = filled_circle,
        head_type = 'nofilled, head_angle = 20, head_length = 0.5,
          background_color = light_gray,  draw_realpart=false)$
 fpprintprec : 5$  ratprint : false$
```

```
load ("Econ1.mac");
```

(%o5) *c:/work5/Econ1.mac*

## 1 Preface

Dowling05.wxmx is one of a number of wxMaxima files available in the section
Economic Analysis with Maxima
on my CSULB webpage. Primary topics are optimization with and without constraints.

We use Maxima to solve a few of the problems (and draw some of the plots) in Ch. 5 of the supplemental text: Introduction to Mathematical Economics, 3rd ed, (Schaum's Outline Series), by Edward T. Dowling (1992, 2001), McGraw-Hill. This modestly priced text is a bargain with many worked out examples. You should compare the examples worked out "by hand" in this text with what we do using Maxima. Section numbers [x.y] refer to sections in Dowling's text.

We have slightly changed some of the symbols used in particular problems.

Ted Woollett
https://web.csulb.edu/~woollett/
April 27, 2022

## 2 Optimization with No Constraints

## 2.1 Conditions for a Relative Maximum or Minimum

Given a function F(x, y), we must first be considering a critical point (x0, y0) at which both Fx = ∂F/∂x = 0 and Fy = ∂F/∂y = 0.

With Fxx = ∂²F/∂x², Fyy = ∂²F/∂y², and Fxy = ∂²^F/(∂x∂y),
we have a RELATIVE MINIMUM provided
Fxx > 0 and Fyy > 0 (convex) and
Fxx*Fyy > (Fxy)^2 (not a saddle point)
at the considered critical point,

we have a RELATIVE MAXIMUM provided
Fxx < 0 and Fyy < 0 (concave) and
Fxx*Fyy > (Fxy)^2 (not a saddle point)
at the considered critical point.

## 2.2 Saddle Points and Inflection Points

If Fxx and Fyy have different signs (one is positive and one is negative, then the critical point is a saddle point (convex in one direction, concave in the orthogonal direction).

If Fxx and Fyy have the same sign (each both positive or each both negative) but Fxx*Fyy < (Fxy)^2, then the function is at an inflection point.

These requirements are incorporated into the function extremum2d.

## 2.3 Analyze (expr, critPts) , plotCP (expr, critPt)

Analyze (expr, critPts), defined in Econ1.mac, is a critical point analysis function which uses Maxima's hessian matrix and determinant functions to perform the required check of the second derivatives at the critical point.

The first argument expr should depend on two (2) variables only, and these do NOT have to be (x,y).

Acceptable forms for critPts (assuming expr depends on x and y):

One critical point: [x = 3, y = 7], or [[x = 3, y = 7]],
Two critical points: [[x = 3, y = 7], [x = -3, y = 7]],
Three critical points: [[x = 3, y = 7], [x = -3, y = 7], [x = 0, y = 0]],
etc.

Analyze prints out information about the nature of the critical point(s) and also the values of the second derivatives of expr, for example [Fxx, Fyy, Fxy] evaluated at the critical point(s).

plotCP(expr, critPoint), also defined in Econ1.mac, shows the surface z(x,y) near the given critical point. critPoint refers to one critical point (only) and has the form
[x = 1, y = 2]
for example, assuming expr depends on the parameters x and y.

## 2.4 Prob. 5.10, jacobian ([expr], [x, y])[1], hessian(expr,[x,y])

(a). Find the critical point(s) for z = 3 x^2 - x y + 2 y^2 - 4 x - 7 y + 12
and determine their nature.

Our solution illustrates the use of Maxima's jacobian function to return a list of the first derivatives of an expression. The list gradz is such a list of first derivatives, here with respect to x and y. The searched for critical point is a point at which both first derivatives are zero, so we use the Maxima function solve, as we have repeatedly used in past chapters. z is a Maxima expression (not a Maxima function) which depends on the parameters x and y, and no other symbols. Hence we could have used just
solns : solve (gradz)
with the same result.

```
z : 3*x^2 - x*y + 2*y^2 - 4*x - 7*y + 12;
gradz : jacobian ([z], [x,y])[1];
solns : solve (gradz, [x,y]);
```

(z) $2\,y^2-x\,y-7\,y+3\,x^2-4\,x+12$
(gradz) $[-y+6\,x-4,4\,y-x-7]$
(solns) $[[x=1,y=2]]$

solve has found one critical point (1,2).

```
→ critPt : soln[1];
(critPt) [x=1,y=2]

→ at (z, critPt);
(%o10) 3

→ Analyze (z, critPt);
1 cp = [x=1,y=2] [relative minimum,value = 3.0]
       secondDeriv = [6,4,-1]
(%o11) done
```

Analyze has found one relative minimum in z located at x = 1, y = 2, at which point z = 3. Analyze finds both zxx and zyy positive, indicating convex shape when passing through a minimum, and we have 24 > 1 (see ingredients next step) so not a saddle point.

We check the second derivatives zxx, zyy, zxy "by hand" using diff.

```
→ secondDeriv : [diff (z,x,2), diff (z,y,2), diff (z,x,1,y,1)];
(secondDeriv) [6,4,-1]
```

In this example, the list of second derivatives is totally numerical, so there is no need here for the normal second step of evaluating the 2nd derivatives at the critical point(s) found.

The hessian matrix, used in the next cell, has zxx and zyy on the diagonal and zxy = zyx on the off-diagonal; this is a faster way to check the second derivatives, and is used in Extr called by Analyze.

```
→ hessian (z, [x,y]);
```

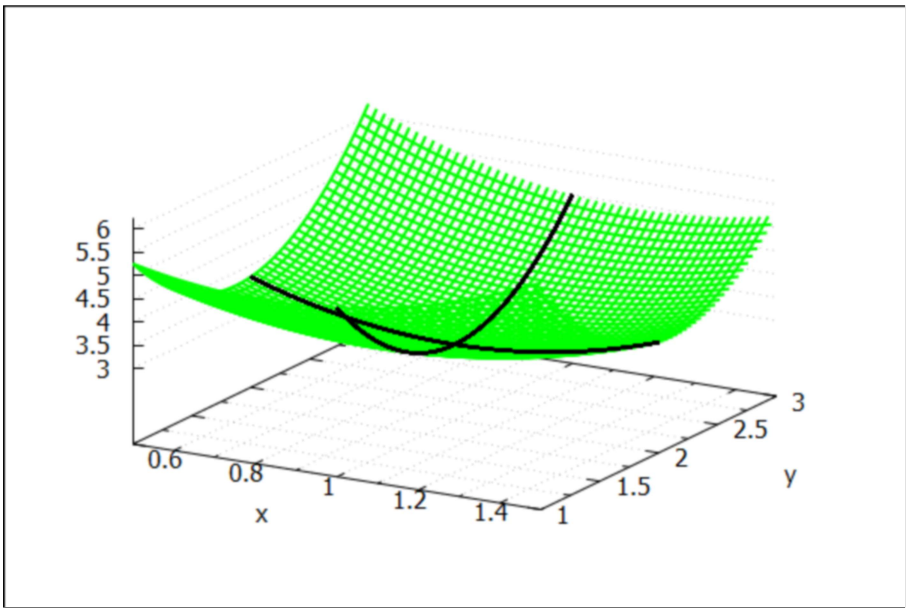
(%o13)
$$\begin{pmatrix} 6 & -1 \\ -1 & 4 \end{pmatrix}$$

plotCP(expr, critPoint), also defined in Econ1.mac, shows the surface z(x,y) near the given critical point.

➔ plotCP (z, critPt);

*surface of* $2\,y^2-x\,y-7\,y+3\,x^2-4\,x+12$
*near critical point =* $[x=1,y=2]$

(%t14)




(%o14)

If you right-click the image and choose "popout interactively", you get a separate Gnuplot window. Expand that window to fullscreen, and drag the image by holding down the left mouse button. This let's you view the surface near the critical point from different angles.

You must close the separate Gnuplot window before you can continue with your calculations.

==========

(b) Same for the expression
f = 60 x + 34 y - 4 x y - 6 x^2 - 3 y^2 + 5.

➔
```
f : 60*x + 34*y - 4*x*y - 6*x^2 - 3*y^2 + 5;
gradf : jacobian ([f], [x,y])[1];
solns : solve (gradf, [x,y]);
```

(f) $-3\,y^2-4\,x\,y+34\,y-6\,x^2+60\,x+5$
(gradf) $[-4\,y-12\,x+60,-6\,y-4\,x+34]$
(solns) $[[x=4,y=3]]$

➔
```
critPt : solns[1];
at (f, critPt);
```

(critPt) $[x=4,y=3]$
(%o19) 176

```
→ Analyze (f, critPt);
```

1 *cp* = $[x=4, y=3]$ [*relative maximum, value = 176.0*]
*secondDeriv* = $[-12, -6, -4]$

(%o20) *done*

```
→ hessian (f, [x,y]);
```

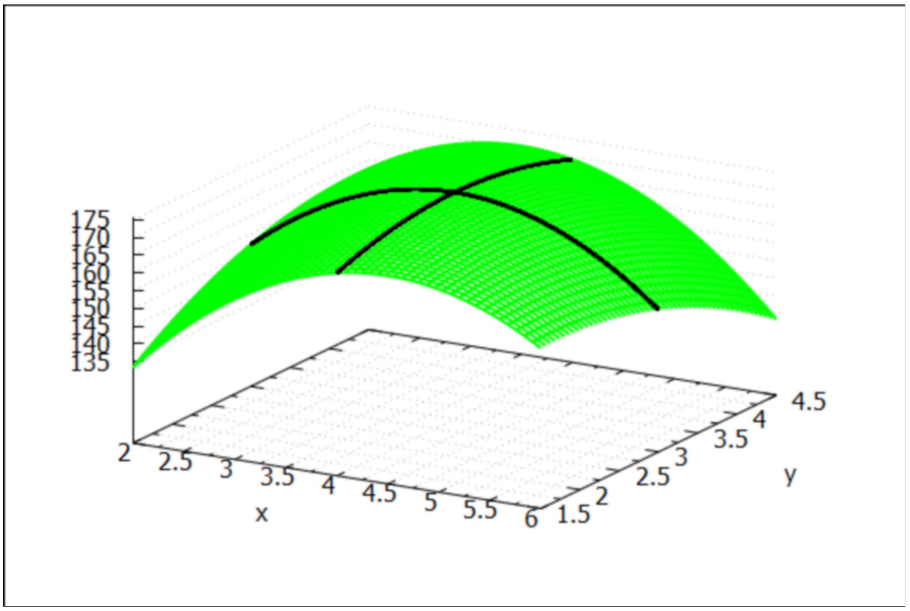
(%o21) $\begin{pmatrix} -12 & -4 \\ -4 & -6 \end{pmatrix}$

fxx and fyy are both negative, indicating concave behavior when passing through a local maximum, and 72 > 16, so not a saddle pt.

```
→ plotCP (f, critPt);
```

*surface of* $-3y^2-4xy+34y-6x^2+60x+5$
*near critical point* = $[x=4, y=3]$

(%t22)




(%o22)

==========

(c) Same for: z = 48 y - 3 x^2 - 6 x y - 2 y^2 + 72 x

```
→ z : 48*y - 3*x^2 - 6*x*y - 2*y^2 + 72*x;
  gradz : jacobian ([z], [x,y])[1];
  solns : solve (gradz, [x,y]);
```

(z) $-2y^2-6xy+48y-3x^2+72x$
(gradz) $[-6y-6x+72, -4y-6x+48]$
(solns) $[[x=0, y=12]]$

```
critPt : solns[1];
at (z, critPt);
```

(critPt) $[x=0,y=12]$

(%o27) 288

```
Analyze (z, critPt);
```

1 *cp* = $[x=0,y=12]$ $[\textit{inflection point}, \textit{value} = 288.0]$
*secondDeriv* = $[-6,-4,-6]$

(%o28) *done*

```
hessian (z, [x,y]);
```

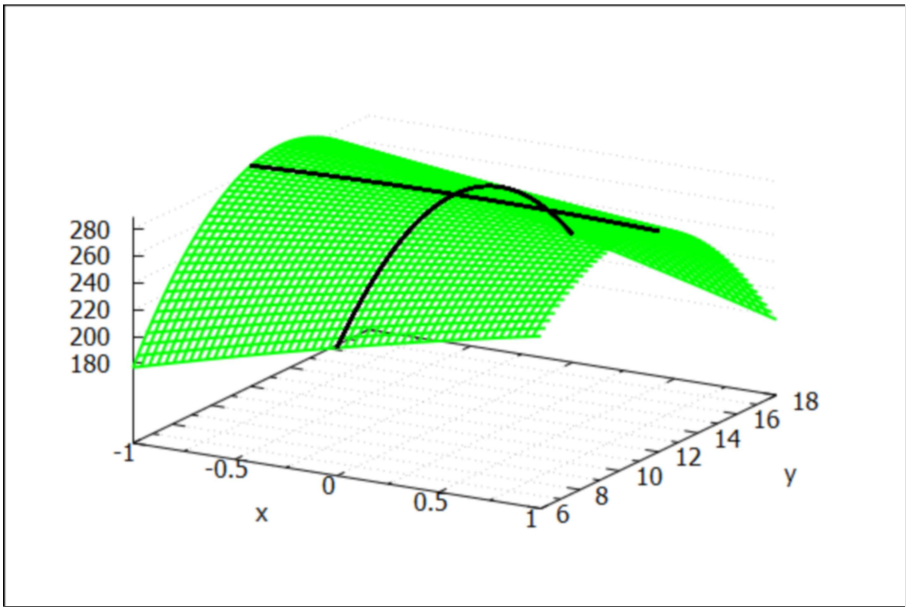
(%o29) $\begin{pmatrix} -6 & -6 \\ -6 & -4 \end{pmatrix}$

zxx and zyy are both negative, indicating possible local maximum with concave shape, but 24 < 36, and since zxx and zyy have the same sign, we have a local inflection point.

```
plotCP (z, critPt);
```

*surface of* $-2\,y^2-6\,x\,y+48\,y-3\,x^2+72\,x$
*near critical point* = $[x=0,y=12]$

(%t30)




(%o30)

==========

(d) Same for f = 5 x^2 - 3 y^2 - 30 x + 7 y + 4 x y

```
f : 5*x^2 - 3*y^2 - 30*x + 7*y + 4*x*y;
gradf : jacobian ([f], [x,y])[1];
solns : solve (gradf, [x,y]);
```

(f) $-3\,y^2+4\,x\,y+7\,y+5\,x^2-30\,x$

(gradf) $[4\,y+10\,x-30,-6\,y+4\,x+7]$

(solns) $[[x=2,y=\frac{5}{2}]]$

```
critPt : solns[1];
at (f, critPt);
```

(critPt) $[x=2,y=\frac{5}{2}]$

(%o35) $-\frac{85}{4}$

```
float(%);
```

(%o36) $-21.25$

```
Analyze (f, critPt);
```

1 *cp* = $[x=2,y=\frac{5}{2}]$ [*saddle point*, *value* = −21.25]

*secondDeriv* = [10, −6, 4]

(%o37) *done*

```
hessian (f, [x,y]);
```

(%o38) $\begin{pmatrix} 10 & 4 \\ 4 & -6 \end{pmatrix}$

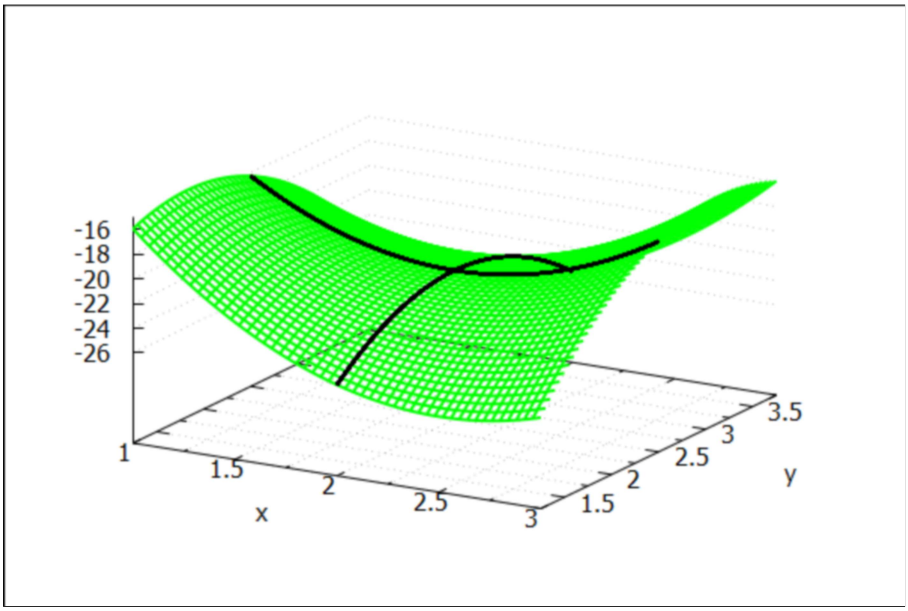
fxx*fyy = -60 < fxy^2 = 16. fxx and fyy opposite signs => saddle point.

```
plotCP (f, critPt);
```

*surface of* $-3\,y^2+4\,x\,y+7\,y+5\,x^2-30\,x$

*near critical point =* $[x=2, y=\frac{5}{2}]$

(%t39)




(%o39)

## 2.5 Prob. 5.11

(a) Analyze the critical points for
z = 3 x^3 - 5 y^2 - 225 x + 70 y + 23

```
z : 3*x^3 - 5*y^2 - 225*x + 70*y + 23;
gradz : jacobian ([z], [x,y])[1];
solns : solve (gradz, [x,y]);
```

(z) $-5\,y^2+70\,y+3\,x^3-225\,x+23$

(gradz) $[9\,x^2-225, 70-10\,y]$

(solns) $[[x=-5, y=7], [x=5, y=7]]$

In this example, solve has found two critical points. Let cp1 refer to (-5,7) and let cp2 refer to (5,7).

```
[cp1, cp2] : solns;
```

(%o43) $[[x=-5, y=7], [x=5, y=7]]$

```
cp1;
```

(%o44) $[x=-5, y=7]$

Values of the Maxima expression z at each of the critical points:

```
→ [at (z, cp1), at (z, cp2)];
(%o45) [1018,-482]
```

The second argument to Analyze can be a list of multiple critical points like solns.

```
→ Analyze (z, solns);
1 cp = [x=-5,y=7] [relative maximum,value = 1018.0]
       secondDeriv = [-90,-10,0]
2 cp = [x=5,y=7] [saddle point,value = -482.0]
       secondDeriv = [90,-10,0]
(%o46) done
```

```
→ H : hessian (z, [x,y]);
```

$$(H)\quad \begin{pmatrix} 18\,x & 0 \\ 0 & -10 \end{pmatrix}$$

This is a case in which the second derivatives are not purely numerical, and we need to evaluate the matrix H at each of the critical points found by solve.

```
→ H1 : at (H, cp1);
```

$$(H1)\quad \begin{pmatrix} -90 & 0 \\ 0 & -10 \end{pmatrix}$$

```
→ H2 : at (H, cp2);
```

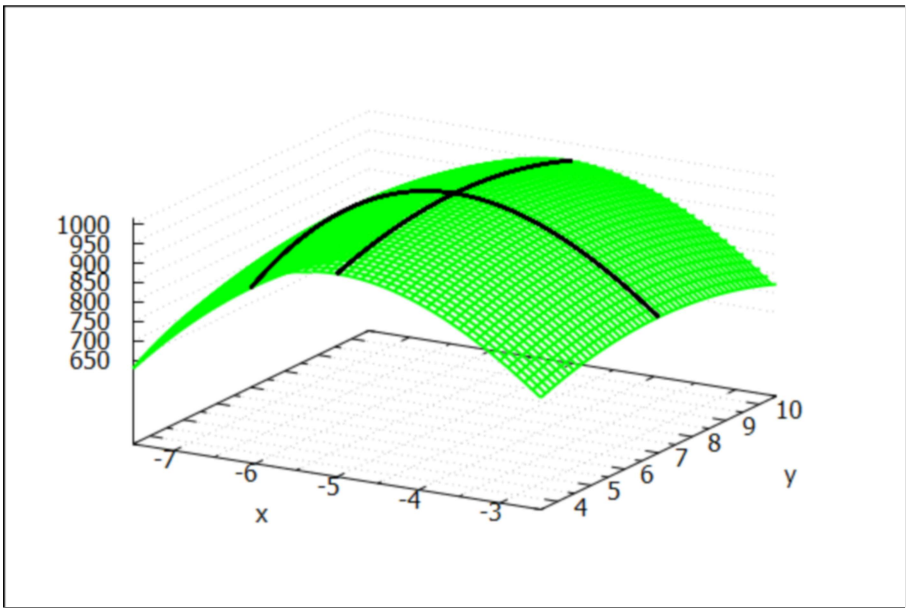
$$(H2)\quad \begin{pmatrix} 90 & 0 \\ 0 & -10 \end{pmatrix}$$

cp1 is a relative maximum.

➔ plotCP (z, cp1);

*surface of* $-5\,y^2+70\,y+3\,x^3-225\,x+23$
*near critical point =* $[x=-5, y=7]$

(%t50)



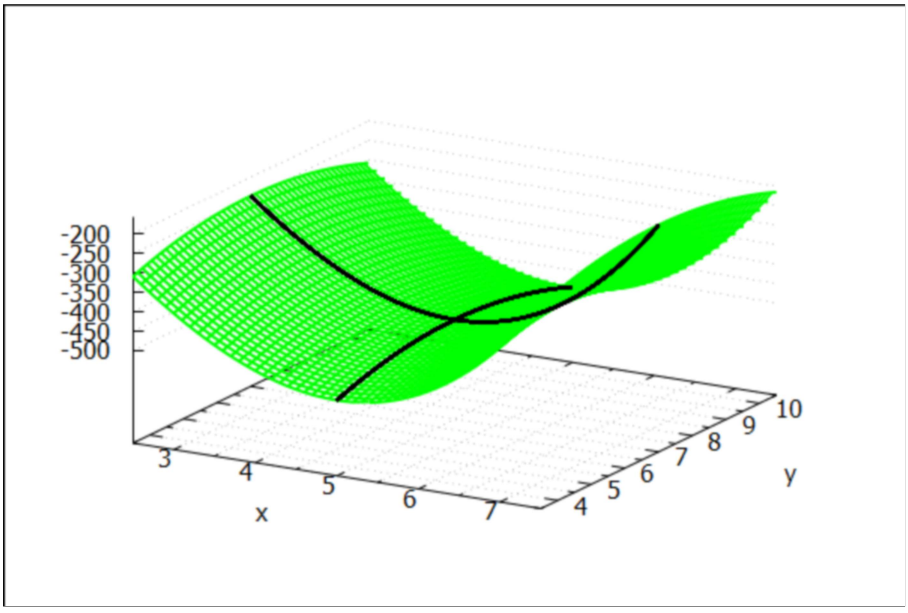

(%o50)

cp2 is a saddle point.

➔ plotCP (z, cp2);

*surface of* $-5\,y^2+70\,y+3\,x^3-225\,x+23$
*near critical point =* $[x=5, y=7]$

(%t51)




(%o51)

---

(b) Same for
f = 3 x^2 + 1.5 y^2 - 18 x y + 17.

```
f : 3*x^3 + 1.5*y^2 - 18*x*y + 17;
gradf : jacobian ([f], [x,y])[1];
solns : solve (gradf, [x,y]);
```

(f) $1.5\,y^2-18\,x\,y+3\,x^3+17$

(gradf) $[9\,x^2-18\,y,3.0\,y-18\,x]$

(solns) $[[x=12,y=72],[x=0,y=0]]$

```
Analyze (f, solns);
```

1 *cp* = $[x=12,y=72]$ [*relative minimum*, *value* = −2575.0]
*secondDeriv* = $[216,3.0,-18]$
2 *cp* = $[x=0,y=0]$ [*inflection point*, *value* = 17.0]
*secondDeriv* = $[0,3.0,-18]$

(%o55) *done*

Check 2nd derivatives and requirements for minimum for cp1 and inflection pt. for cp2.

H is the matrix of second derivatives of f, and depends on the parameter x.

```
H : hessian (f, [x,y]);
```

(H) $\begin{pmatrix} 18\,x & -18 \\ -18 & 3.0 \end{pmatrix}$

Let H1 be H evaluated at the first critical point solns[1].

```
H1 : at (H, solns[1]);
```

(H1) $\begin{pmatrix} 216 & -18 \\ -18 & 3.0 \end{pmatrix}$

Check of fxx*fyy and fxy^2:

```
[216*3, 18^2];
```

(%o58) $[648,324]$

We see that fxx*fyy > fxy^2, so not an inflection point.

Let H2 be H evaluated at the second critical point solns[2], in which x = 0, y = 0.

```
H2 : at (H, soln[2]);
```

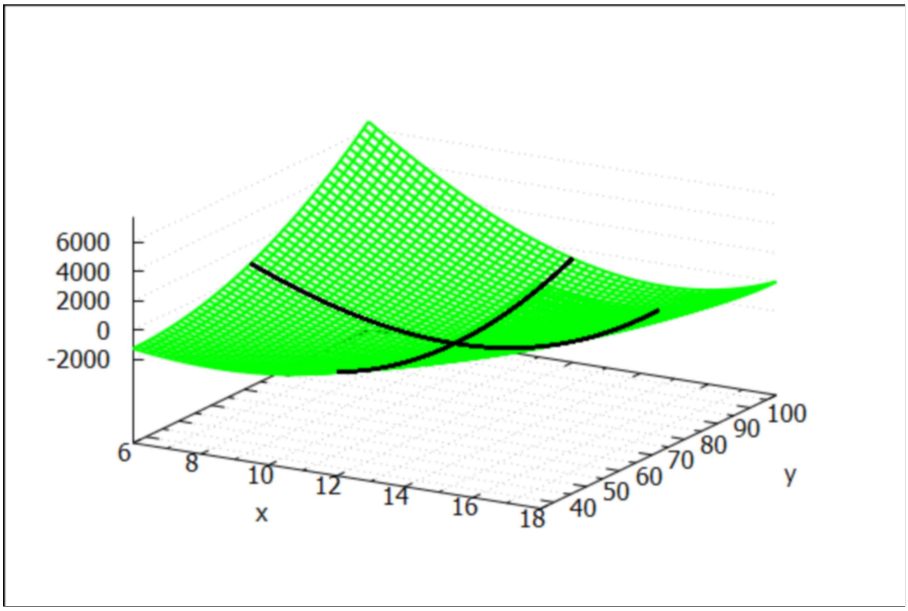
(H2)
$$\begin{pmatrix} 0 & -18 \\ -18 & 3.0 \end{pmatrix}$$

fxx*fyy = 0*3 = 0 < fxy^2 = 18^2, Analyze calls it an inflection point. We can check that with the second plot below.

```
plotCP (f, soln[1]);
```

*surface of* $1.5\,y^2-18\,x\,y+3\,x^3+17$
*near critical point =* $[x=12, y=72]$

(%t60)



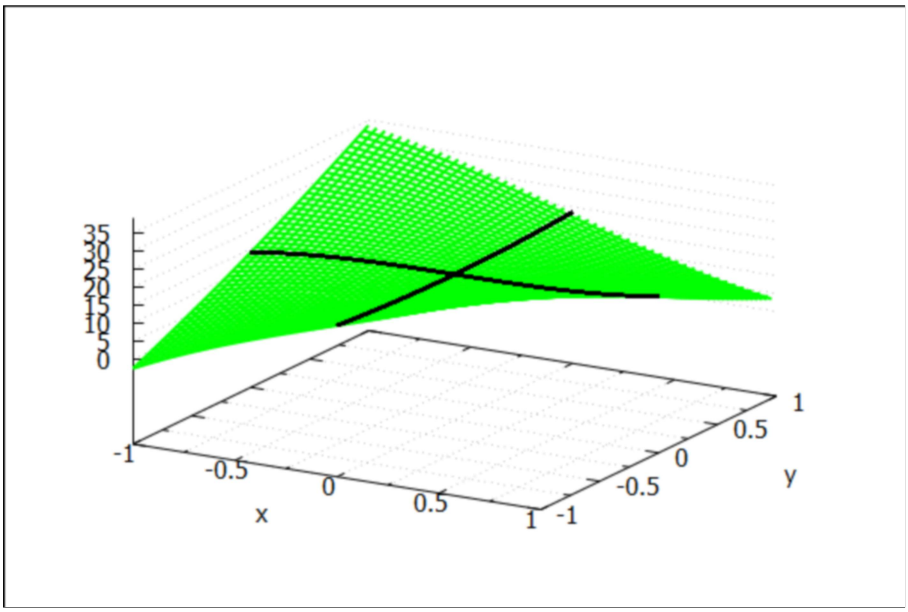

(%o60)

Here is the inflection point case:

```
plotCP (f, soln[2]);
```

*surface of* $1.5\,y^2-18\,x\,y+3\,x^3+17$
*near critical point =* $[x=0,y=0]$

(%t61)




(%o61)

---

(c) Same for
f = 3 x^3 - 9 x y + 3 y^3.

```
f : 3*x^3 - 9*x*y + 3*y^3;
gradf : jacobian ([f], [x,y])[1];
soln : solve (gradf, [x,y]);
```

(f) $3\,y^3-9\,x\,y+3\,x^3$

(gradf) $[9\,x^2-9\,y,9\,y^2-9\,x]$

(soln) $[[x=-\frac{\sqrt{3}\,\%i+1}{2},y=\frac{\sqrt{3}\,\%i-1}{2}],[x=\frac{\sqrt{3}\,\%i-1}{2},y=-\frac{\sqrt{3}\,\%i+1}{2}],[x=1,y=1],[x=0,y=0]]$

Ignore complex roots; for Economics problems, we are generally interested in real and non-negative values of the search variables. We can use Maxima's rest (alist, 2) function to return a list without the first two sublists in alist.

```
critPts : rest (soln, 2);
```

(critPts) $[[x=1,y=1],[x=0,y=0]]$

```
Analyze (f, critPts);
```

1 *cp* = [*x* = 1, *y* = 1] [*relative minimum*, *value* = −3.0]
*secondDeriv* = [18, 18, −9]
2 *cp* = [*x* = 0, *y* = 0] [*inflection point*, *value* = 0.0]
*secondDeriv* = [0, 0, −9]

(%o66) *done*

```
H : hessian (f, [x,y]);
```

$$(H)\quad \begin{pmatrix} 18\,x & -9 \\ -9 & 18\,y \end{pmatrix}$$

```
H1 : at (H, critPts[1]);
```

$$(H1)\quad \begin{pmatrix} 18 & -9 \\ -9 & 18 \end{pmatrix}$$

```
H2 : at (H, critPts[2]);
```

$$(H2)\quad \begin{pmatrix} 0 & -9 \\ -9 & 0 \end{pmatrix}$$

---

(d) Same for
f = x^3 - 6 x^2 + 2 y^3 + 9*y^2 - 63 x - 60 y.

```
f : x^3 - 6*x^2 + 2*y^3 + 9*y^2 - 63*x - 60*y;
gradf : jacobian ([f], [x,y])[1];
solns : solve (gradf, [x,y]);
```

(f) $2\,y^3+9\,y^2-60\,y+x^3-6\,x^2-63\,x$

(gradf) $[3\,x^2-12\,x-63, 6\,y^2+18\,y-60]$

(solns) $[[x=-3, y=-5], [x=7, y=-5], [x=-3, y=2], [x=7, y=2]]$

If this were an Economics problem, we would ignore the solutions which involve negative values for the search variables. However, we will keep them all for this example, as Dowling does.

```
Analyze (f, solns);
```

```
1  cp =  [x=-3,y=-5]  [relative maximum,value = 383.0]
         secondDeriv =  [-30,-42,0]
2  cp =  [x=7,y=-5]  [saddle point,value = -117.0]
         secondDeriv =  [30,-42,0]
3  cp =  [x=-3,y=2]  [saddle point,value = 40.0]
         secondDeriv =  [-30,42,0]
4  cp =  [x=7,y=2]  [relative minimum,value = -460.0]
         secondDeriv =  [30,42,0]
```

(%o73) *done*

```
H : hessian (f, [x,y]);
```

(H)
$$\begin{pmatrix} 6x-12 & 0 \\ 0 & 12y+18 \end{pmatrix}$$

We use a small "do loop" to print out the values of H at the four critical points.

```
for j thru 4 do print (j,"   ", at (H, solns[j]))$
```

1
$$\begin{pmatrix} -30 & 0 \\ 0 & -42 \end{pmatrix}$$

2
$$\begin{pmatrix} 30 & 0 \\ 0 & -42 \end{pmatrix}$$

3
$$\begin{pmatrix} -30 & 0 \\ 0 & 42 \end{pmatrix}$$

4
$$\begin{pmatrix} 30 & 0 \\ 0 & 42 \end{pmatrix}$$

# 3 Numerical Method: mnewton

Solving systems of nonlinear equations is much more difficult than solving one nonlinear equation. A wider variety of behavior is possible: determining the existence and number of solutions or even a good starting guess is more complicated. There is no method which can guarantee convergence to the desired solution. The computing labor increases rapidly with the number of dimensions of the problem.

When solve cannot cope with the problem of finding solutions of a set of equations, we can try Maxima's mnewton function.

In numerical analysis, Newton's method (also known as the Newton Raphson method or the Newton Fourier method) is perhaps the best known method for finding successively better approximations to the zeros (or roots) of a real valued function. Newton's method can often converge remarkably quickly, especially if the iteration begins sufficiently near the desired root. Just how near sufficiently near needs to be and just how remarkably quickly depends on the problem, as is discussed in detail below.

Unfortunately, far from the desired root, Newton's method can easily lead an unwary user astray, and astray with little warning. Such users are advised to heed the advice of Press, et. al. (1992), who suggest embedding Newton's method in a routine that also detects possible convergence failures.

Newton's method can also be used to find a minimum or maximum of such a function, by finding a zero in the function's first derivative.

Consider the function
f = x^2 + 2 y^2 - 4 (x + y - 1) / 3.
This is a function which solve CAN find roots. We can compare the mnewton method to what solve has found symbolically.

```
f : x^2 + 2*y^2 - 4*(x + y - 1)/3;
gradf : jacobian ([f], [x,y])[1];
solns : solve (gradf, [x, y]);
```

(f) $$2\,y^2-\frac{4\,(y+x-1)}{3}+x^2$$

(gradf) $$\left[2\,x-\frac{4}{3},4\,y-\frac{4}{3}\right]$$

(solns) $$\left[\left[x=\frac{2}{3},y=\frac{1}{3}\right]\right]$$

To use the mnewton method, we first need to load in the software file mnewton.mac.

```
load ("mnewton.mac");
```

(%o79) *C:/maxima-5.43.2/share/maxima/5.43.2/share/mnewton/mnewton.mac*

Recall gradf defined above is just a list of the first derivatives fx and fy.
mnewton ([fx, fy], [x, y], [xguess, yguess]) then tries to locate a point (x0,y0)
such that fx and fy are simultaneously very small ("close to zero"), and that point (x0,y0)
is then an approximate numerical value of a critical point of f.

```
mnewton (gradf, [x, y], [1, 1] );
```

(%o80) $$\left[\left[x=0.66667,y=0.33333\right]\right]$$

```
(%i81) newtonepsilon;
```

(%o81) $1.0\ 10^{-8}$

## 3.1 Manual Description of mnewton (gradF, varL, GuessL)

From the Maxima manual:

mnewton (FuncList,VarList,GuessList)

Multiple nonlinear functions solution using the Newton method. FuncList is the list of functions to solve, VarList is the list of variable names, and GuessList is the list of initial approximations.

[Note: if FuncList has the form [expr1,expr2], then mnewton is trying to solve for variables such that expr1 = 0 and simultaneously expr2 = 0.]

The solution is returned in the same format that solve() returns. If the solution is not found, [] is returned.

This function is controlled by global variables newtonepsilon and newtonmaxiter.

Given an expression f depending on two variables, call them x and y here, the 'FuncList' is a list containing the first derivative of f wrt x, and the first derivative of f wrt y. mnewton then is looking for a value of (x,y) for which these first derivatives are equal to zero, i.e, a "critical point."

## 3.2 Solving for Roots Numerically: an Example

If solve cannot reduce the system of equations to a polynomial in one variable, we get an algsys error message. To see this error return, try this pair of inputs in a cell:

```
gradf : [x^2 + y^2 -2,  exp(x - 1) + y^3 - 2];
solns : solve (gradf, [x, y]);
```

We pursue this root finding problem with mnewton. Define mn (x0,y0) as a Maxima function which calls mnewton with the initial guess list [x0, y0].

```
(%i82) gradf : [x^2 + y^2 -2,  exp(x - 1) + y^3 - 2];
       mn(x0,y0) := mnewton(gradf, [x, y], [x0, y0] )$
```

(gradf) $[y^2+x^2-2, y^3+\%e^{x-1}-2]$

```
(%i84) mn(1.1,0.9);
```

(%o84) $[[x=1.0, y=1.0]]$

Check that x = 1, y = 1 is a solution of the set of equations in gradf, such that gradf[1] = 0 and gradf[2] = 0 at this critical point.

```
→ at (gradf, [x = 1, y = 1]);
(%o85) [0,0]
```

We can explore the sensitivity of the solution found to the initial guess by making a list gL of alternative starting points [x0,y0], and trying them out.

```
→ gL : makelist([1.1 + i/10, 0.9 - i/10], i, 0, 5) ;
(gL) [[1.1,0.9],[1.2,0.8],[1.3,0.7],[1.4,0.6],[1.5,0.5],[1.6,0.4]]

→ gL[1];
(%o87) [1.1,0.9]

→ apply ('mn, gL[1]);
(%o88) [[x=1.0,y=1.0]]

→ length (gL);
(%o89) 6

→ for j thru length (gL) do
    print ( j,"  ", gL[j], "  ", apply ('mn, gL[j]));
1   [1.1,0.9]   [[x=1.0,y=1.0]]
2   [1.2,0.8]   [[x=1.0,y=1.0]]
3   [1.3,0.7]   [[x=1.0,y=1.0]]
4   [1.4,0.6]   [[x=1.0,y=1.0]]
5   [1.5,0.5]   [[x=-0.71375,y=1.2209]]
6   [1.6,0.4]   [[x=1.0,y=1.0]]
(%o90) done
```

# 4 Optimization with Equality Constraints

Differential calculus is also used to maximize or minimize a function subject to constraint. Given a function f(x,y) subject to a constraint g(x, y) = k (a constant), a new function F can be formed by (1) writing the constraint as a quantity equal to zero, (2) multiplying it by λ (the "Lagrange multiplier"), and (3) adding the product to the original function:

$$F(x, y, \lambda) = f(x, y) + \lambda\,[k - g(x, y)] \qquad \text{[Eq. 5.6]}$$

Here F(x, y, λ ) is the Lagrangian function, f(x, y) is the original or "objective function," and (k - g(x, y)) is the "constraint." Since the constraint is always set equal to zero, the product λ [k - g(x, y)] also equals zero, and the addition of the term does not change the value of the objective function. Critical values x0 , y0 , and λ0 , at which the Lagrangian function is optimized, are found by taking the partial derivatives of F with respect to all three independent variables, setting them equal to zero, and solving simultaneously three equations (the "first order conditions")

Fx (x,y,λ) = 0, Fy (x, y,λ) = 0, Fλ (x, y,λ) = 0, ( Fz means partial derivative wrt z here)

Second-order conditions differ from those of unconstrained optimization and are treated in Ch. 12, Sec. 12.5. See Example 9; Problems 5.12 to 5.14; Ch. 6, Sections 6.6, 6.9, and 6.10; and Problems 6.28 to 6.39 and 6.41 to 6.44.

For constraints involving inequalities, see Chapter 13 for concave programming.

## 4.1 Example 9, optimum (expr, varL, constraints)

Optimize z = 4 x^2 + 3 x y + 6 y^2 such that x + y = 56.

Instead of using F for the Lagrangian function in this example, we use L.

We first form the Lagrangian function L, using λ to represent the Lagrangian multiplier, calculate a list of the first derivatives of L wrt (x, y, λ) using the Maxima function jacobian. We then ask solve to find the critical points, each such point having a definite value of x,y and λ.

```
→ z : 4*x^2 + 3*x*y + 6*y^2;
  g : x + y;
  L : z + λ*(56 - g);
  gradL : jacobian ([L], [x, y, λ])[1];
  solns : solve (gradL, [x, y, λ]);
```

(z) $6\,y^2+3\,x\,y+4\,x^2$

(g) $y+x$

(L) $(-y-x+56)\,\lambda+6\,y^2+3\,x\,y+4\,x^2$

(gradL) $[-\lambda+3\,y+8\,x,-\lambda+12\,y+3\,x,-y-x+56]$

(solns) $[[x=36,y=20,\lambda=348]]$

solve has found one critical point solution.

```
→ soln : solns[1];
```

(soln) $[x=36,y=20,\lambda=348]$

Let cp1 be the list [x = 36, y = 20]. We can use Maxima to extract these elements here by using rest (alist, -1) which returns all of the elements of alist except the last one.

```
→ cp1 : rest (soln, -1);
```

(cp1) $[x=36,y=20]$

What is the value of the expression z at this critical point?

```
→ at (z, cp1);
```

(%o98) 9744

In addition to the step-by-step method above, you can use our Maxima function optimum.

The Maxima function optimum (expr, vL, constraint(s)) is defined in Econ1.mac and can solve problems involving one or two equality constraints. The first argument expr can be a function of more than two variables, and vL is a list of the variables used in expr and g.

The nature of any critical points found by solve is analyzed (inside optimum) by forming the "bordered Hessian matrix" evaluated at each critical point and then looking at the sign pattern of the "leading principal minors". Internally, the Maxima function optimum uses lam[1], lam[2], for any needed Lagrange multipliers. The solution is printed out in two rows, with the second row being forced into floating point form. "objsub" refers to the value of the objective function f(x,y) at the critical point.

See Dowling11-12.wxmx and Dowlingfit11-12.pdf for further examples.

```
→ optimum (z, [x, y], 56 - g);
```

$$lagrangian = 6\,y^2+(3\,x-lam_1)\,y+4\,x^2-lam_1\,x+56\,lam_1$$

$$soln = [x=36,y=20,lam_1=348] \quad objsub = 9744$$

$$soln = [x=36.0,y=20.0,lam_1=348.0] \quad objsub = 9744.0$$

*relative minimum*

$$LPM's = [LPM3=-14.0]$$

(%o99) *done*

optimum always produces a global list you can access via the symbol cp:

```
→ cp;
```

(%o100) $[[x=36,y=20,lam_1=348]]$

```
→ at (z, cp[1] );
```

(%o101) 9744

Here is a plot showing the constraint line x + y = 56, and curves of z = constant.
This plot illustrates a graphical method of locating the largest value of z consistent with an equality constraint x + y = 56.
z is the Maxima expression depending on the parameters x and y defined above.

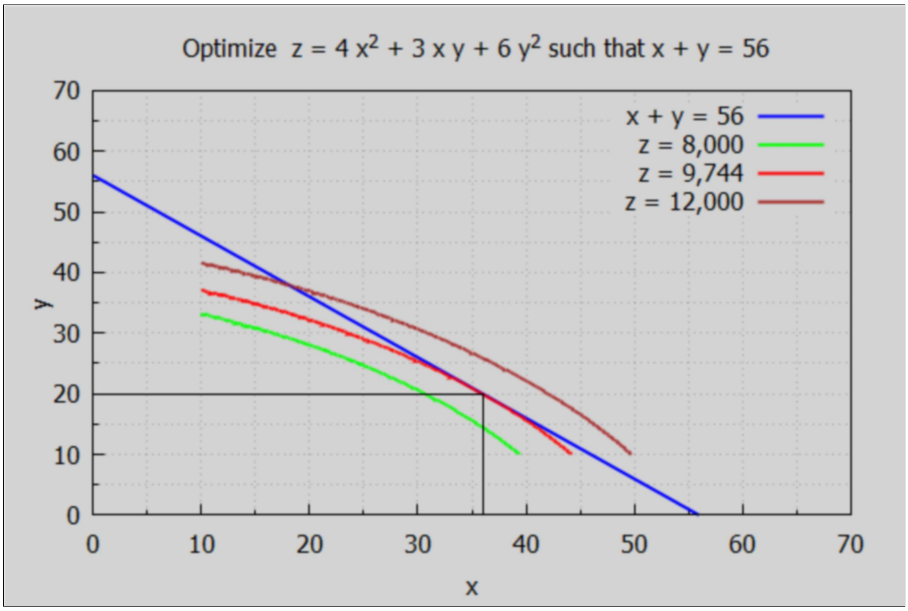
```
→ z;
```

(%o102) $6\,y^2+3\,x\,y+4\,x^2$

The draw2d function implicit is being used here with the syntax
implicit (f(x, y), x, x1, x2, y, y1, y2)
which tries to make a plot of all points of the x-y plane for which f(x,y) = 0,
within the region: x1 <= x <= x2 and y1 <= y <= y2.

```
wxdraw2d (xlabel = "x", ylabel = "y", xrange = [0,70], yrange = [0,70],
   title = " Optimize  z = 4 x^2 + 3 x y + 6 y^2 such that x + y = 56",
    key = "x + y = 56", explicit (56 - x, x, 0, 70),
   color = green, key = "z = 8,000", implicit (z = 8e3, x, 10, 50, y, 10, 50),
   color = red, key = "z = 9,744", implicit (z = 9744, x, 10, 50, y, 10, 50),
   color = brown, key = "z = 12,000", implicit (z = 1.2e4, x, 10, 50, y, 10, 50),
   color = black, line_width = 1, key = "", explicit (20,x, 0, 36),
   parametric (36, yy, yy, 0, 20) )$
```

(%t103)




From Econ1.mac:

---

optimum (func, varL, constraints)

handles one or more equality constraints

syntax

optimum (f,varL) case no equality constraints, calls CPtest

optimum (f,varL, fc) case one equality constraint fc = 0, calls BHtest

optimum (f, var, [fc])

optimum (f, varL, [fc1, fc2]) case, both fc1 = 0 and fc2 = 0.

etc.

defines global cp

screens solutions to only deal with real non-negative values

of the search variables in varL.

---

optimumAll (func, varL, constraints)

accepts all solutions returned by solve.

defines global cp

calls CPtest or BHtest

---

## 4.2 Significance of the Langrange Multiplier [5.6]

Let's require x + y = 57 (instead of 56). What happens to the critical point values, and what happens to the value of the lagrangian multiplier. Using optimum, for convenience,

```
optimum (z, [x, y], 57 - g);
```

$$lagrangian = 6\,y^2 + (3\,x - lam_1)\,y + 4\,x^2 - lam_1\,x + 57\,lam_1$$

$$soln = \left[x = \frac{513}{14}, y = \frac{285}{14}, lam_1 = \frac{4959}{14}\right] \quad objsub = \frac{282663}{28}$$

$$soln = [x = 36.643, y = 20.357, lam_1 = 354.21] \quad objsub = 1.0095\,10^4$$

*relative minimum*

$$LPM's = [LPM3 = -14.0]$$

(%o104) *done*

By increasing the targeted value of x+y from 56 to 57, we get a new solution for which both x and y are somewhat increased, and the value of the objective function z has increased from 9744 to 10,095.1

```
10095 - 9744;
```

(%o105) 351

Δz = 351 ~ 348, so the value of λ, the lagrangian multiplier, in our original calculation (348) predicts the approximate increase in z due to a one (1) unit increase in the targeted sum (x + y), here from 56 to 57.

The Lagrange multiplier λ approximates the marginal impact on the objective function caused by a small change in the constant of the constraint. With λ = 348 in Example 9, for instance, a 1-unit increase (decrease) in the constant k of the constraint would cause Fs (evaluated at the critical point) to increase (decrease) by approximately 348 units, as is demonstrated in Example 10. Lagrange multipliers are often referred to as "shadow prices." In utility maximization subject to a budget constraint, for example, λ will estimate the marginal utility of an extra dollar of income. See Problem 6.36.

Note: Since in (Eq 5.6) at the beginning of our Sec. "Constrained Optimization with Lagrange Multipliers", λ [k - g(x, y)] = λ [g(x, y) - k] = 0, either form can be added to or subtracted from the objective function without changing the critical values of x and y. Only the sign of λ will be affected. For the interpretation of λ given here, however, the precise form used in Equation (5.6) should be adhered to. That form is:

$$F(x, y, \lambda) = f(x, y) + \lambda\,[k - g(x, y)] \qquad [Eq.\ 5.6]$$

See Problems 5.12 to 5.14.

## 4.3 Prob 5.12 a)

5.12. (1) Use Lagrange multipliers to optimize the following functions subject to the given constraint, and
(2) estimate the effect on the value of the objective function from a 1-unit change in the constant of the constraint.

a.) z = 4 x^2 - 2 x y + 6 y^2 subject to the constraint x + y = 72.

```
z : 4*x^2 - 2*x*y + 6*y^2;
g : x + y;
L : z + λ*(72 - g);
gradL : jacobian ([L], [x, y, λ])[1];
solns : solve (gradL, [x, y, λ]);
```

(z) $6\,y^2-2\,x\,y+4\,x^2$

(g) $y+x$

(L) $(-y-x+72)\,\lambda+6\,y^2-2\,x\,y+4\,x^2$

(gradL) $[-\lambda-2\,y+8\,x,-\lambda+12\,y-2\,x,-y-x+72]$

(solns) $[[x=42,y=30,\lambda=276]]$

```
optimum (z, [x, y], 72 - g);
```

$lagrangian = 6\,y^2+(-2\,x-lam_1)\,y+4\,x^2-lam_1\,x+72\,lam_1$

$soln = [x=42,y=30,lam_1=276]$ $objsub = 9936$

$soln = [x=42.0,y=30.0,lam_1=276.0]$ $objsub = 9936.0$

*relative minimum*

$LPM's = [LPM3=-24.0]$

(%o111) *done*

Now increase the targeted value of x+y from 72 to 73 (one unit).

```
optimum (z, [x, y], 73 - g);
```

$lagrangian = 6\,y^2+(-2\,x-lam_1)\,y+4\,x^2-lam_1\,x+73\,lam_1$

$soln = [x=\frac{511}{12},y=\frac{365}{12},lam_1=\frac{1679}{6}]$ $objsub = \frac{122567}{12}$

$soln = [x=42.583,y=30.417,lam_1=279.83]$ $objsub = 1.0214\,10^4$

*relative minimum*

$LPM's = [LPM3=-24.0]$

(%o112) *done*

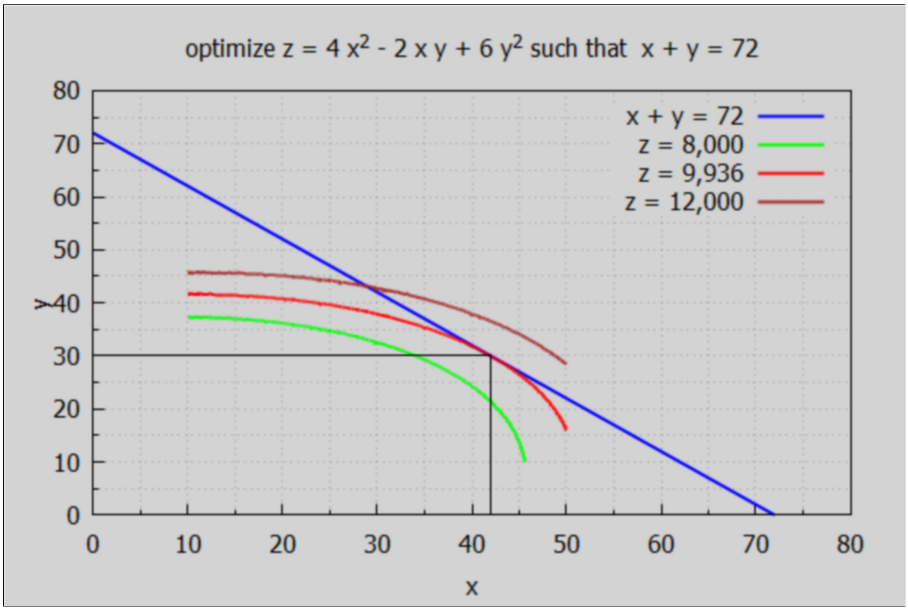
Raising the targeted sum x + y from 72 to 73 produces an increase in z of 278, which is close to the critical point value of λ in our original calculation (278 ~ 276).

Here is a plot showing the constraint line x + y = 72 and some curves of z = constant.

```
(%i112) z;
```

(%o113) $6\,y^2-2\,x\,y+4\,x^2$

```
(%i114) wxdraw2d (xlabel = "x", ylabel = "y", xrange = [0,80], yrange = [0,80],
    title = "optimize z = 4 x^2 - 2 x y + 6 y^2 such that  x + y = 72",
    key = "x + y = 72", explicit (72 - x, x, 0, 72),
    color = green, key = "z = 8,000", implicit (z = 8e3, x, 10, 50, y, 10, 50),
    color = red, key = "z = 9,936",
    implicit (z = 9936, x, 10, 50, y, 10, 50), color = brown, key = "z = 12,000",
    implicit (z = 1.2e4, x, 10, 50, y, 10, 50), color = black, line_width = 1, key = "",
    explicit (30,x, 0, 42), parametric (42, yy, yy, 0, 30) )$
```




(%t114)

# 5 Total and Partial Differentials [5.8]

del(x) represents the differential of the variable x in Maxima.

diff returns an expression containing del if an independent variable is not specified.

```
(%i115) z : x^4 + 8*x*y + 3*y^3;
```

(z) $3\,y^3+8\,x\,y+x^4$

Given the Maxima expression z depending on variables x and y, diff(z) returns the total differential of z

dz = (∂z/∂x) dx + (∂z/∂y) dy

Here is an example of using diff(z) without specifying the variable. The Maxima function diff (expr) looks at the free variables (parameters) in expr and mutiplies the partial derivatives diff (z, x), diff(z,y),...etc by the corresponding del(x), del(y),...

```
→ diff (z);
```
(%o116) $(9\,y^2+8\,x)\,\mathrm{del}(y)+(8\,y+4\,x^3)\,\mathrm{del}(x)$

```
→ grind(%)$
(9*y^2+8*x)*del(y)+(8*y+4*x^3)*del(x)$
```

Using a dot (.) to "multiply" two lists in Maxima, we could create the same result using:

```
→ [diff(z,x), diff(z,y)] . [del(x), del(y)];
```
(%o118) $(9\,y^2+8\,x)\,\mathrm{del}(y)+(8\,y+4\,x^3)\,\mathrm{del}(x)$

If we declare y to be a constant, using the syntax:

```
→ declare (y, constant);
```
(%o119) *done*

then diff(z) only returns  diff (z, x)*del(x).

```
→ diff(z);
```
(%o120) $(4\,x^3+8\,y)\,\mathrm{del}(x)$

Use remove (y, constant) to undo that declaration.

```
→ remove (y, constant);
```
(%o121) *done*

```
→ diff(z);
```
(%o122) $(9\,y^2+8\,x)\,\mathrm{del}(y)+(8\,y+4\,x^3)\,\mathrm{del}(x)$

A partial differential holding y constant can be formed using:

```
→  dzx : diff(z,x) * del(x);
```
(dzx) $(8\,y+4\,x^3)\,\mathrm{del}(x)$

```
→  at (dzx, [x = 1, y = 2] ), del(x) = 1/10;
```
(%o124) 2

A partial differential measures the change in the dependent variable of a multivariate function resulting from a small change in one of the independent variables and assumes the other independent variables are constant.
See Problems 5.16 and 5.17 and 6.45 to 6.52.

# 6 Total Derivatives [5.9]

Given a case where z = f(x, y) and y = g(x), that is, when x and y are not independent, a change in x will affect z directly through the function f and indirectly through the function g.

To measure the effect of a change in x on z when x and y are not independent, the total derivative must be found. The total derivative measures the direct effect of x on z, which is ∂z/∂x, plus the indirect effect of x on z through y which is ∂z/∂y * dy/dx.

In brief the total derivative in this case is
dz/dx = ∂z/∂x + ∂z/∂y * dy/dx.

## 6.1 Example 14

z = 6*x^3 + 7*y, and y = g(x) = 4*x^2 + 3*x + 8. Find the total derivative of z wrt x

```
→  z : 6*x^3 + 7*y;
   g : 4*x^2 + 3*x + 8;
```
(z) $7\,y+6\,x^3$

(g) $4\,x^2+3\,x+8$

```
→  dzdx : diff(z,x) + diff(z,y)*diff(g,x), expand;
```
(dzdx) $18\,x^2+56\,x+21$

since the result so far no longer depends on y, we don't need to go another step to replace all y's by g's and expand again.

## 6.2 Another Example

Change the definition of z so the second term is 7*y^3, and leave y = same g(x).

```
→  z : 6*x^3 + 7*y^3;
   g : 4*x^2 + 3*x + 8;
```

(z) $7\,y^3+6\,x^3$

(g) $4\,x^2+3\,x+8$

```
→  dzdx : diff(z,x) + diff(z,y)*diff(g,x), expand;
```

(dzdx) $168\,x\,y^2+63\,y^2+18\,x^2$

To get the total derivative dzdx as a function of only x, we now need to replace y with g and expand.

```
→  dzdx : at (dzdx, y = g), expand;
```

(dzdx) $2688\,x^5+5040\,x^4+13776\,x^3+12681\,x^2+13776\,x+4032$

## 6.3 Example 15

Let z = 8*x^2 + 3*y^2, x = f(t) = 4*t, y = g(t) = 5*t. Find the total derivative dz/dt.

```
→  f : 4*t;
   g : 5*t;
   z : 8*x^2 + 3*y^2;
```

(f) $4\,t$

(g) $5\,t$

(z) $3\,y^2+8\,x^2$

```
→  dzdt : diff(z,x)*diff(f,t) + diff(z,y)*diff (g,t), expand;
```

(dzdt) $30\,y+64\,x$

```
→  dzdt : at (dzdt, [x = f, y = g] ), expand;
```

(dzdt) $406\,t$

# *7 Implicit and Inverse Function Rules [5.10]*

Functions of the form y = f(x) express y explicitly in terms of x and are called explicit functions. Expressions of the form f(x, y) = 0 do not express y explicitly in terms of x and are called implicit functions.

If an implicit function f(x, y) = 0 exists and fy # 0 at the point around which the implicit function is defined, the total differential is simply fx dx + fy dy = 0. Recalling that a derivative is a ratio of differentials, we can then rearrange the terms to get the "implicit function rule":

If f(x,y) = 0, then dy/dx = - fx/fy = - ∂f/∂x / ∂f/∂y.

Given a function p = F(q), an inverse function q = G(p) exists if each value of q yields one and only one value of p. Assuming the inverse function exists,

dp = Fq * dq and dq = Gp * dp where Fq = partial derivative of F(q) wrt q and Gp = partial derivative of G(p) wrt p. Then dp = Fq * Gp * dp and dividing by Gp*dp, Fq = 1 / Gp, or we get the "inverse function rule":

∂p/∂q = 1/ ∂q/∂p

## 7.1 Example 16b

Given the implicit function f(x,y) = 3 x^4 - 7 y^5 - 86 = 0, find dy/dx using the implicit function rule.

```
→ f : 3*x^4 - 7*y^5 - 86;
```

(f) $-7\,y^5+3\,x^4-86$

```
→ dydx : -diff (f,x) / diff (f,y);
```

(dydx) $\frac{12\,x^3}{35\,y^4}$

## 7.2 Example 17

Given Q = 25 + 3 P^3, find dP/dQ.

```
→ Q : 25 + 3*P^3;
  dPdQ : 1/diff(Q, P);
```

(Q) $3\,P^3+25$

(dPdQ) $\frac{1}{9\,P^2}$

This answer assumes P # 0!