

Dowling13Bfit.wmx

Optimization with Inequality Constraints

TABLE OF CONTENTS

PREFACE	1
KKT CONDITIONS	2
PROB 13.35 MAX PROFIT, 1 CONSTRAINT	2
Graphical Method.....	4
KKT Method	6
Grid Search Method	8
PROB 13.36, CONTINUATION	9
PROB 13.37 MIN COST, 1 CONSTRAINT	12
PROB 13.38, CONTINUATION	14
CHRIS JONES ONE CONSTRAINT EXAMPLE	16
PROB 13.39 MAX UTILITY, 2 CONSTRAINTS	18
Grid Search Method with Two Constraints	19
KKT Method with Two Constraints	20
Chaing-Wainwright Ex. 1 Max Utility, 2 Constraints	20
Chaing-Wainwright Ex. 2 Min Cost, 2 Constraints	23
Baldani, Bradfield, Turner Example: Max $f(x,y)$ w/ 2 Constraints	25

```
(%i5) load(draw)$ set_draw_defaults(line_width=2, draw_realpart=false)$
      fpprintprec:5$ ratprint:false$ kill(all)$
```

```
(%i1) load ("Econ1.mac");
```

```
(%o1) c:/work5/Econ1.mac
```

1 Preface

Dowling13B.wmxm uses Maxima to work some of the problems in the last section of Ch. 13 of Introduction to Mathematical Economics (3rd ed), by Edward T. Dowling, (Schaum's Outline Series), McGraw-Hill, 2012. This text is a bargain, with many complete problems worked out in detail. You should compare Dowling's solutions, worked out "by hand", with what we do using Maxima here.

Typical problems considered here are to maximize utility or profit while satisfying one or two constraints, which have the form $g_1(x,y) \geq 0$, $g_2(x,y) \geq 0$.

Also considered is the problem of minimizing cost while satisfying one or two constraints, which have the form $g_1(x,y) \leq 0$, $g_2(x,y) \leq 0$.

Three methods are used for each maximization or minimization problem considered.

- 1). Purely graphical using a plot with lines and/or curves which represent the objective function $f(x,y) = \text{constant}$, and the lines and/or curves which represent the constraints.
- 2). A grid search (which ignores points which do not satisfy the constraints) of some (x,y) region, and
- 3). The KKT algorithm.

Solutions are checked by forming the Lagrangian function and calculating its first derivatives.

Two code files, Econ1.mac and KKT.mac are available in the same section (of Economic Analysis with Maxima), which define Maxima functions used in this worksheet.

Use load ("Econ1.mac") and load ("KKT.mac").

Edwin L. (Ted) Woollett
<http://web.csulb.edu/~woollett/>
July, 2021

2 *KKT(f, min/max, g), KKT(f, min/max, g1, g2)*

Maxima code KKT (Karush-Kuhn-Tucker). Use load ("KKT.mac") to load our definitions of the Maxima function KKT.

SYNTAX:

KKT (f(x,y), max, g(x,y)) with $g(x,y) \geq 0$

KKT (f(x,y), max, g1(x,y), g2(x,y)) with $g1(x,y) \geq 0$ and $g2(x,y) \geq 0$

KKT (f(x,y), min, g(x,y)) with $g(x,y) \leq 0$

KKT (f(x,y), min, g1(x,y), g2(x,y)) with $g1(x,y) \leq 0$ and $g2(x,y) \leq 0$

(%i2) load ("KKT.mac");

(%o2) c:/work5/KKT.mac

3 *KKT Method Summary*

References:

1. Alpha C. Chiang and Kevin Wainwright, Fundamental Methods of Mathematical Economics, 4th ed, McGraw-Hill, 2005 (Sec. 13.1).

2. https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions

For one constraint, say KKT(f, max, g), our code uses the first derivatives of the Lagrangian function $L = f + \lambda * g$. Our code is limited to two variables which must be called (x,y). Sticking to that limitation we then require that:

a solution to the maximization problem with one constraint satisfy

1. $\partial L / \partial x \leq 0$, $x \geq 0$, $x * \partial L / \partial x = 0$
2. $\partial L / \partial y \leq 0$, $y \geq 0$, $y * \partial L / \partial y = 0$
3. $\partial L / \partial \lambda \geq 0$, $\lambda \geq 0$, $\lambda * \partial L / \partial \lambda = 0$

a solution to the minimization problem with one constraint satisfy

1. $\partial L / \partial x \geq 0$, $x \geq 0$, $x * \partial L / \partial x = 0$
2. $\partial L / \partial y \geq 0$, $y \geq 0$, $y * \partial L / \partial y = 0$
3. $\partial L / \partial \lambda \leq 0$, $\lambda \geq 0$, $\lambda * \partial L / \partial \lambda = 0$.

Note that in each case above, $\partial L / \partial \lambda = g(x,y)$, so the conditions on $\partial L / \partial \lambda$ are the same as the conditions on g.

For two constraints, say KKT(f , max, g_1 , g_2), our code uses the first derivatives of the Lagrangian function $L = f + \lambda_1 g_1 + \lambda_2 g_2$. Our code is limited to two variables which must be called (x,y) . Sticking to that limitation we then require that:

a solution to the maximization problem with two constraints satisfy

1. $\partial L / \partial x \leq 0$, $x \geq 0$, $x^* \partial L / \partial x = 0$
2. $\partial L / \partial y \leq 0$, $y \geq 0$, $y^* \partial L / \partial y = 0$
3. $\partial L / \partial \lambda_1 \geq 0$, $\lambda_1 \geq 0$, $\lambda_1^* \partial L / \partial \lambda_1 = 0$
4. $\partial L / \partial \lambda_2 \geq 0$, $\lambda_2 \geq 0$, $\lambda_2^* \partial L / \partial \lambda_2 = 0$

a solution to the minimization problem with two constraints satisfy

1. $\partial L / \partial x \geq 0$, $x \geq 0$, $x^* \partial L / \partial x = 0$
2. $\partial L / \partial y \geq 0$, $y \geq 0$, $y^* \partial L / \partial y = 0$
3. $\partial L / \partial \lambda_1 \leq 0$, $\lambda_1 \geq 0$, $\lambda_1^* \partial L / \partial \lambda_1 = 0$.
4. $\partial L / \partial \lambda_2 \leq 0$, $\lambda_2 \geq 0$, $\lambda_2^* \partial L / \partial \lambda_2 = 0$.

Note that in each case above, $\partial L / \partial \lambda_1 = g_1(x,y)$, so the conditions on $\partial L / \partial \lambda_1$ are the same as the conditions on g_1 (and likewise for $g_2(x,y)$ and $\partial L / \partial \lambda_2$).

The extension to more than two variables and more than two constraints is then straightforward.

4 Dowling Prob. 13.35, Maximize Profit, 1 constraint

Maximize profits $\pi = 64x - 2x^2 + 96y - 4y^2 - 13$ subject to the production constraint $x + y \leq 20$.

If we define $g(x,y) = 20 - x - y \geq 0$, then $20 \geq x + y \rightarrow x + y \leq 20$.

Let Pr stand for the profit.

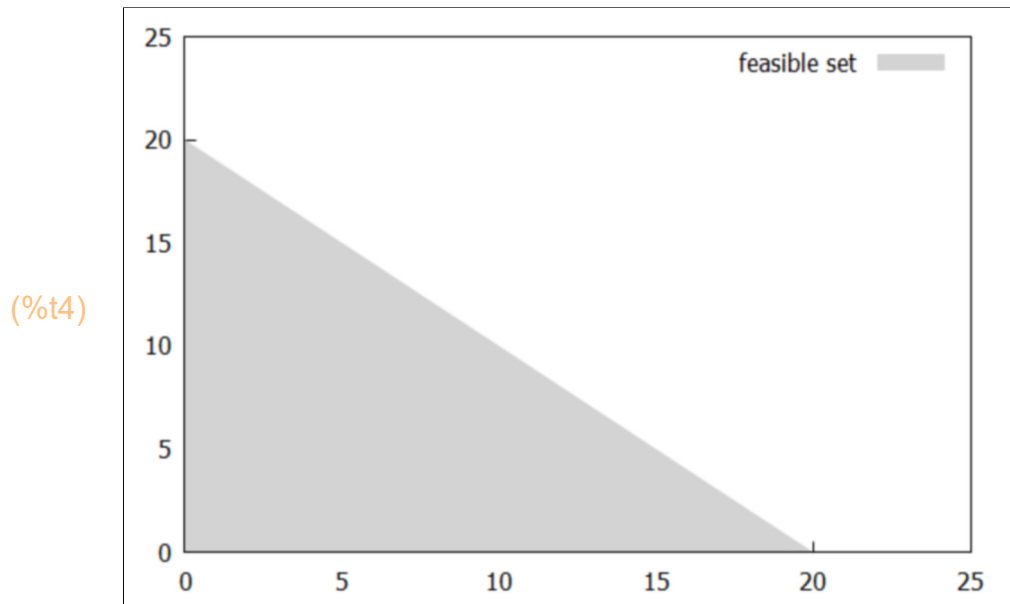
$$(\%i3) \quad \text{Pr} : 64*x - 2*x^2 + 96*y - 4*y^2 - 13;$$

$$(\text{Pr}) \quad -4*y^2 + 96*y - 2*x^2 + 64*x - 13$$

4.1 Graphical Search

Plot the "feasible set" of solution points (x,y) .

```
(%i4) wxdraw2d (yrange = [0,25], xrange = [0,25], key = "feasible set",  
filled_func = true, fill_color = light_gray, explicit (20 - x,x,0,30))$
```



Look at the values of profit Pr at the corners of the feasible set.

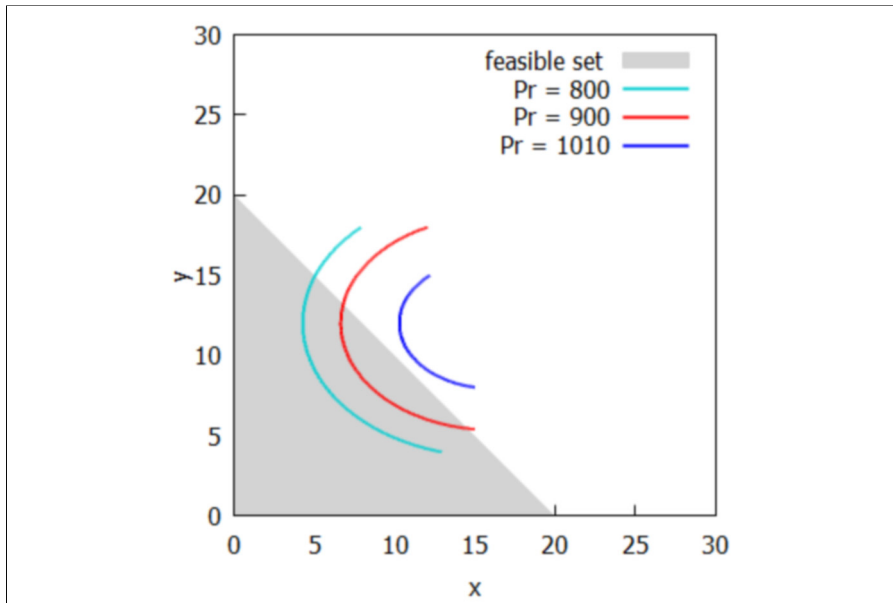
```
(%i5) [at (Pr, [x = 0, y = 20]), at (Pr, [x = 20, y = 0]) ];
```

```
(%o5) [307,467]
```

Plot some $Pr = \text{constant}$ curves using the implicit function inside `wxdraw2d`.

```
(%i6) wxdraw2d (yrange = [0,30], xrange = [0,30], line_width = 2, proportional_axes = 'xy,
xlabel = "x", ylabel = "y", key = "feasible set",
filled_func = true, fill_color = light_gray, explicit (20 - x,x,0,30),
filled_func = false,
color = dark_turquoise, key = "Pr = 800", implicit (Pr = 800, x,3,18,y,4,18),
color = red, key = "Pr = 900", implicit (Pr = 900, x,3,15,y,4,18),
color = blue, key = "Pr = 1010", implicit (Pr = 1010, x,3,15,y,4,15))$
```

(%t6)



The plot implies the optimum (maximum) point (in the feasible set) will be found on the boundary defined by $x + y \leq 20$.

4.2 KKT Search

Try using our function KKT

```
(%i7) Pr;
```

```
(%o7) -4 y2+96 y-2 x2+64 x-13
```

```
(%i8) g : 20 - x - y;
```

```
(g) -y-x+20
```

(%i9) KKT (Pr,max, g);

$$\text{soln} = \left[x = \frac{32}{3}, y = \frac{28}{3}, \text{lam} = \frac{64}{3} \right] \quad g(x,y) \text{ at soln} = 0$$

$$\text{objsub} = \frac{2969}{3}$$

$$\text{soln} = [x = 10.667, y = 9.3333, \text{lam} = 21.333] \quad \text{objsub} = 989.67$$

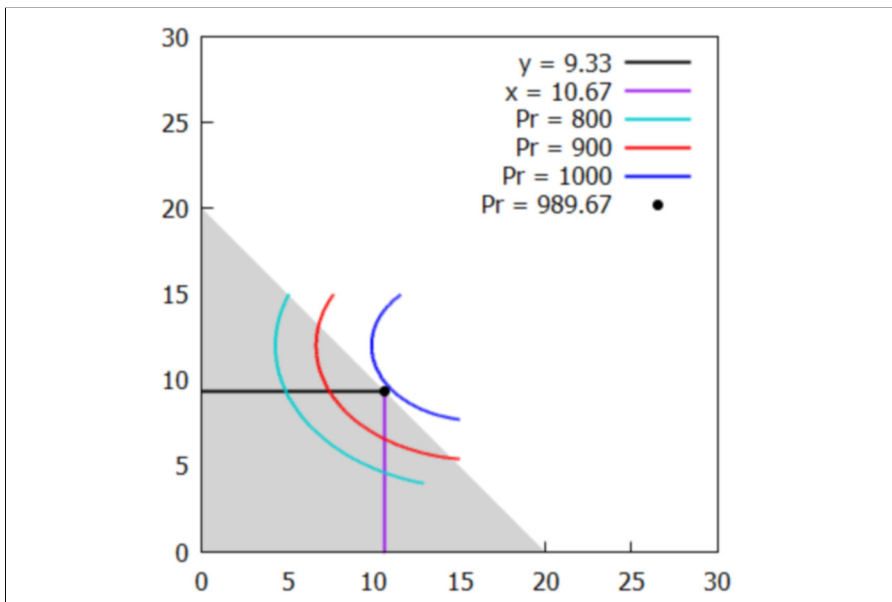
(%o9) done

KKT has found the optimum solution with $\text{lam} > 0$ and hence $g = 0$, the constraint is binding; the maximum occurs on the boundary (of the feasible set) defined by $g \geq 0$, more specifically, the boundary defined by $x + y \leq 20$.

Put the maximum value location on our figure using the function points inside wxdraw2d.

```
(%i10) wxdraw2d (yrange = [0,30], xrange = [0,30], proportional_axes = 'xy,
  line_width = 2, filled_func = true, fill_color = light_gray, explicit (20 - x,x,0,30),
  filled_func = false,
  color = black, key = "y = 9.33", explicit (28/3, x, 0, 32/3),
  color = purple, key = "x = 10.67", parametric (32/3, t, t, 0, 28/3),
  color = dark_turquoise, key = "Pr = 800", implicit (Pr = 800, x,3,15,y,4,15),
  color = red, key = "Pr = 900", implicit (Pr = 900, x,3,15,y,4,15),
  color = blue, key = "Pr = 1000", implicit (Pr = 1000, x,3,15,y,4,15),
  color = black,
  point_type = filled_circle, key = "Pr = 989.67", points ([ [32/3, 28/3] ]) )$
```

(%t10)



To check the solution found, form the Lagrangian function L , and find the relevant derivatives.

```
(%i12) L : Pr + lam*g;
      [Lx : diff(L,x), Ly : diff(L,y), Llam : diff (L, lam) ];
(L)    -4 y^2+96 y+lam (-y-x+20) -2 x^2+64 x-13
(%o12) [-4 x-lam+64, -8 y-lam+96, -y-x+20]
```

Then find their values at the solution found by KKT.

```
(%i13) subst ([x=32/3,y=28/3,lam=64/3], [Lx, Ly, Llam,g]);
(%o13) [0,0,0,0]
```

Since the optimum x , y , and λ values are all greater than zero, part of the KKT conditions for an optimum are that the corresponding derivatives of the Lagrangian function should be zero. So we can find the same solution using solve:

```
(%i14) solve ([Lx,Ly,Llam], [x, y, lam] );
(%o14) [[x= $\frac{32}{3}$ ,y= $\frac{28}{3}$ ,lam= $\frac{64}{3}$ ]]
```

4.3 gridSearch (func, xR, yR, maxmin, constraint)

In our software file Econ1.mac is a straightforward grid search function we call gridSearch, with the following four acceptable invocations: (You could use $f(u,v)$, $g(u,v)$, for example instead of $f(x,y)$, $g(x,y)$, or any other variable names. The first symbol (x) in the second argument (list) is always taken as one of the two variable names, and the first symbol (y) in the third argument (list) is always taken as the second variable name.

```
gridSearch (f(x,y), [x, x1, x2, dx], [y, y1, y2, dy], max, g(x,y))
  returns [fmax, [xmax, ymax] ] , and only looks at points for which  $g(x,y) \geq 0$ .
```

```
gridSearch (f(x,y), [x, x1, x2, dx], [y, y1, y2, dy], min, g(x,y))
  returns [fmin, [xmin, ymin] ] , and only looks at points for which  $g(x,y) \leq 0$ .
```

```
gridSearch (f(x,y), [x, x1, x2, dx], [y, y1, y2, dy], max, g1(x,y), g2(x,y) )
  returns [fmax, [xmax, ymax] ] , and only looks at points for which both
   $g1(x,y) \geq 0$  and  $g2(x,y) \geq 0$ .
```

```
gridSearch (f(x,y), [x, x1, x2, dx], [y, y1, y2, dy], min, g1(x,y), g2(x,y) )
  returns [fmin, [xmin, ymin] ] , and only looks at points for which
  both  $g1(x,y) \leq 0$  and  $g2(x,y) \leq 0$ .
```

```
(%i15) gridSearch (Pr, [x,10,11,0.01], [y,9,10,0.01], max, g);
(%o15) [989.67, [x=10.67,y=9.33]]
```


which is the same answer as we have been finding in other ways.

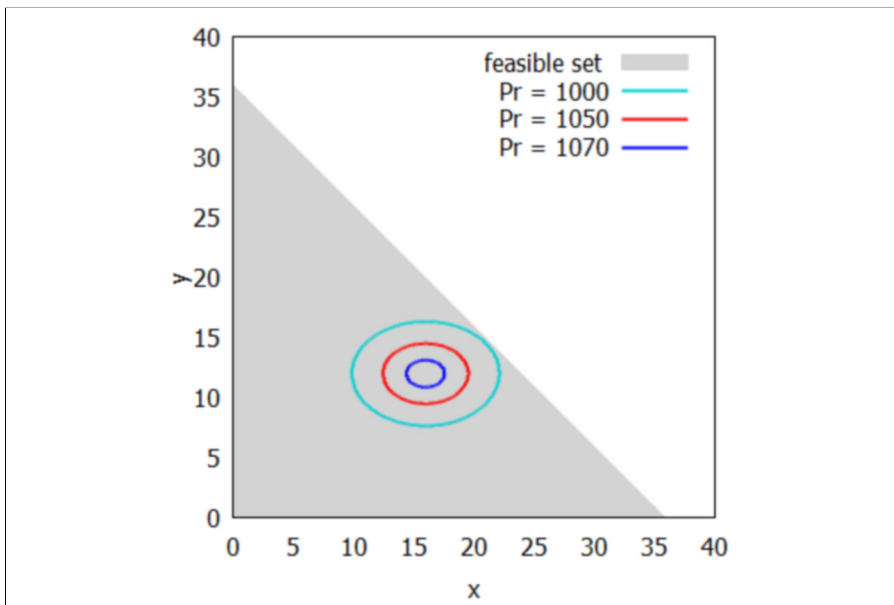
5 Dowling Prob. 13.36, Continuation Problem

Maximize the production function in Prob. 13.35 with the new production constraint $x + y \leq 36$, enlarging the feasible set. Let $g(x,y) = 36 - x - y \geq 0$. Then $36 \geq x + y$.

Let's use the same `wxdraw2d` code we used with Prob. 13.35, and change the boundary to $x+y = 36$. Add some curves of constant Pr using `implicit`. This takes some experimentation with the values of the constant and the range of x and y used inside the `implicit` function.

```
(%i16) wxdraw2d (yrange = [0,40], xrange = [0,40], xlabel = "x", ylabel = "y",
  proportional_axes = 'xy, line_width = 2, key = "feasible set",
  filled_func = true, fill_color = light_gray, explicit (36 - x,x,0,40),
  filled_func = false,
  color = dark_turquoise, key = "Pr = 1000", implicit (Pr = 1000, x,3,30,y,4,25),
  color = red, key = "Pr = 1050", implicit (Pr = 1050, x,3,30,y,4,20),
  color = blue, key = "Pr = 1070", implicit (Pr = 1070, x,3,30,y,4,20))$
```

(%t16)



If you create a separate gnuplot window by using `draw2d` instead of `wxdraw2d`:

```
(%i17) draw2d (yrange = [0,40], xrange = [0,40], xlabel = "x", ylabel = "y",
  proportional_axes = 'xy, line_width = 2, key = "feasible set",
  filled_func = true, fill_color = light_gray, explicit (36 - x,x,0,40),
  filled_func = false,
  color = dark_turquoise, key = "Pr = 1000", implicit (Pr = 1000, x,3,30,y,4,25),
  color = red, key = "Pr = 1050", implicit (Pr = 1050, x,3,30,y,4,20),
  color = blue, key = "Pr = 1070", implicit (Pr = 1070, x,3,30,y,4,20));
```

```
(%o17) [gr2d (explicit, implicit, implicit, implicit)]
```

you can expand the plot to full screen and then watch the point coordinates change as you move the cursor around. Placing the cursor at the center of the blue oval, you have almost found by inspection that $x \sim 16$, $y \sim 12$. You have to close the separate gnuplot window to continue your work in wxMaxima.

It is clear from this plot that the optimum (maximum) point will be "interior" to the feasible set and the constraint is "not binding".

After defining a new expression for $g \geq 0$, we can call `gridSearch1` for an approximate solution.

```
(%i18) g : 36 - x - y;
```

```
(g) -y - x + 36
```

```
(%i19) gridSearch (Pr, [x, 15,17,0.1], [y, 11,13,0.1], max, g);
```

```
(%o19) [1075.0, [x=16.0, y=12.0]]
```

Next let's try out KKT,

```
(%i20) KKT (Pr, max, g);
```

```
constraint is non-binding
```

```
soln = [lam=0, x=16, y=12] g(x,y) at soln = 8
```

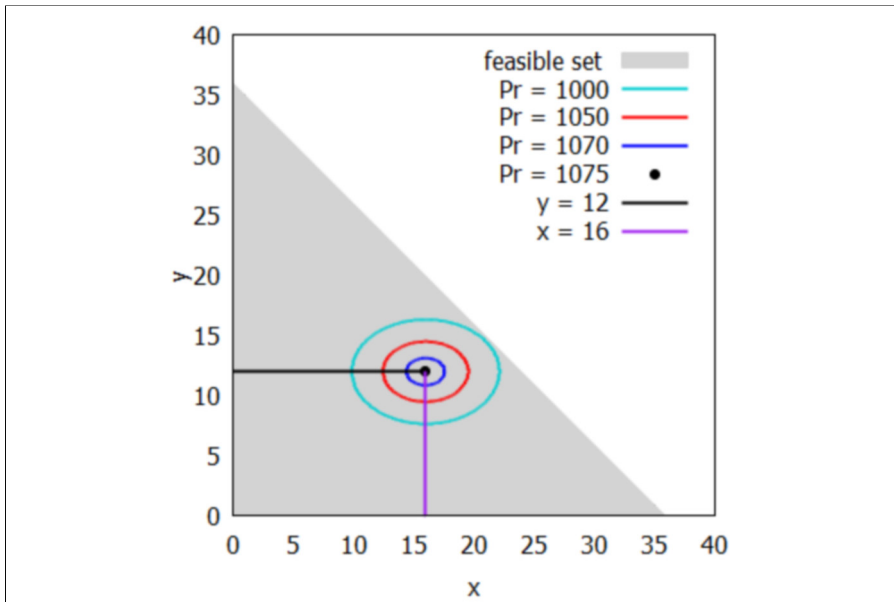
```
objsub = 1075
```

```
soln = [lam=0.0, x=16.0, y=12.0] objsub = 1075.0
```

```
(%o20) done
```

```
(%i21) wxdraw2d (yrange = [0,40], xrange = [0,40], xlabel = "x", ylabel = "y",
    proportional_axes = 'xy, line_width = 2, key = "feasible set ",
    filled_func = true, fill_color = light_gray, explicit (36 - x,x,0,40),
    filled_func = false,
    color = dark_turquoise, key = "Pr = 1000", implicit (Pr = 1000, x,3,30,y,4,25),
    color = red, key = "Pr = 1050", implicit (Pr = 1050, x,3,30,y,4,20),
    color = blue, key = "Pr = 1070", implicit (Pr = 1070, x,3,30,y,4,20),
    color = black,
    point_type = filled_circle, key = "Pr = 1075", points ([ [16, 12] ]),
    key = "y = 12", explicit (12, x, 0, 16),
    color = purple, key = "x = 16", parametric (16, t, t, 0, 12) )$
```

(%t21)



Check solution using first derivatives of the Lagrangian function.

```
(%i24) L : Pr + lam*g;
[Lx : diff(L,x), Ly : diff(L,y), Llam : diff (L, lam)];
subst ([x = 16,y = 12, lam = 0], [Pr,x + y,Lx, Ly, Llam,g]);
```

```
(L) -4 y^2+96 y+lam (-y-x+36) -2 x^2+64 x-13
```

```
(%o23) [-4 x-lam+64, -8 y-lam+96, -y-x+36]
```

```
(%o24) [1075,28,0,0,8,8]
```

With $x + y = 28 < 36$, the production constraint is met with a profit = 1075 and the solution is optimal since none of the Kuhn-Tucker conditions are violated, and no other possible scenario satisfies all the Kuhn-Tucker conditions. (The program KKT systematically explores all possible scenarios.)

The solution returned by solve, IF we assume $Lx = 0$, $Ly = 0$, $Llam = 0$, is:

```
(%i25) soln : solve ([Lx,Ly,Llam], [x,y,lam])[1];
```

```
(soln) [x= $\frac{64}{3}$ , y= $\frac{44}{3}$ , lam= $-\frac{64}{3}$ ]
```

and is SUB-OPTIMAL since $\lambda = -64/3 < 0$

```
(%i26) subst (soln,[Pr,x + y,Lx, Ly, Llam,g]);
```

```
(%o26) [ $\frac{2969}{3}$ , 36, 0, 0, 0, 0]
```

```
(%i27) float(%);
```

```
(%o27) [989.67, 36.0, 0.0, 0.0, 0.0, 0.0]
```

This sub-optimal solution requires a greater production ($x + y = 36$) with less profit (989.67) than the optimal solution.

6 Dowling Prob. 13.37, Minimize Cost, one constraint

Dowling, PROB 13.37:

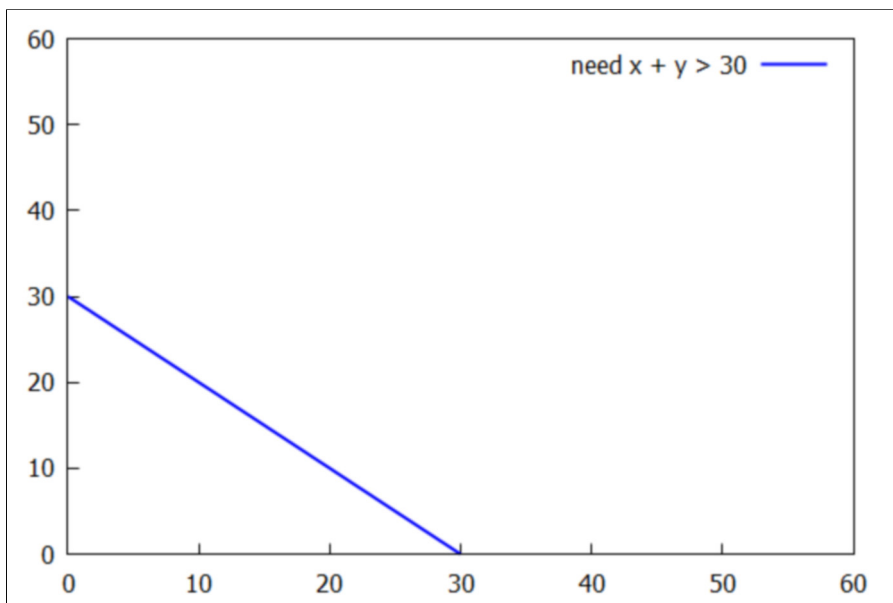
Minimize the cost $c = 5x^2 - 80x + y^2 - 32y$ subject to $x + y \geq 30$.

We then need $g(x,y) = 30 - x - y \leq 0$ so that $30 \leq x + y$.

Note carefully: when calling `KKT(f, min, g)`, we need $g(x,y) \leq 0$.

```
(%i28) wxdraw2d (yrange = [0,60], xrange = [0,60],
key = "need x + y > 30",explicit (30 - x,x,0,30))$
```

```
(%t28)
```

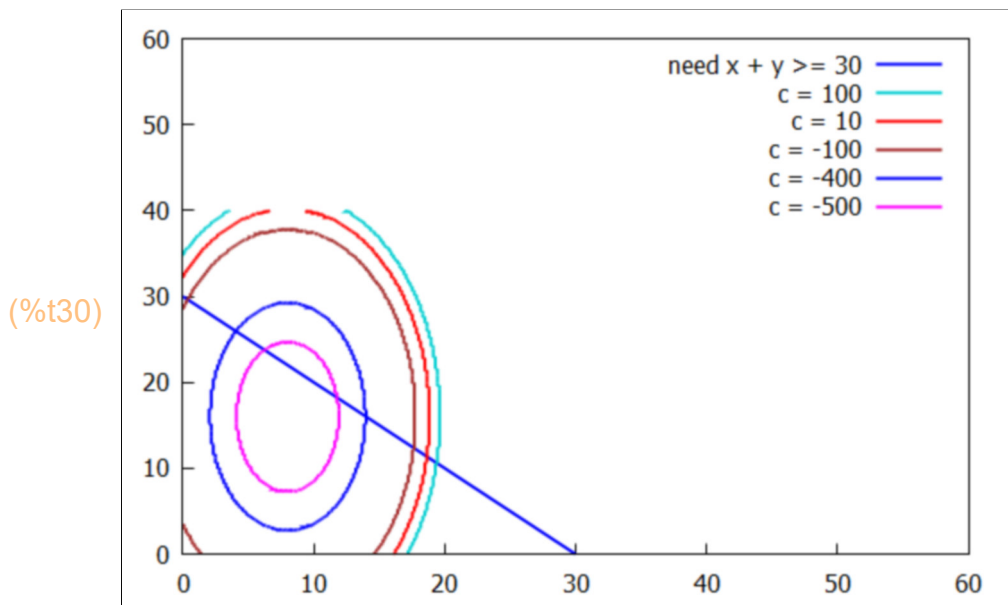


Add some curves $c = \text{const}$ to this plot.

```
(%i29) c : 5*x^2 - 80*x + y^2 - 32*y;
```

```
(c) y^2 - 32 y + 5 x^2 - 80 x
```

```
(%i30) wxdraw2d (yrange = [0,60], xrange = [0,60],
  key = "need x + y >= 30", explicit (30 - x, x, 0, 30),
  color = dark_turquoise, key = "c = 100", implicit (c = 100, x, 0, 40, y, 0, 40),
  color = red, key = "c = 10", implicit (c = 10, x, 0, 40, y, 0, 40),
  color = brown, key = "c = -100", implicit (c = -100, x, 0, 40, y, 0, 40),
  color = blue, key = "c = -400", implicit (c = -400, x, 0, 40, y, 0, 40),
  color = magenta, key = "c = -500", implicit (c = -500, x, 0, 40, y, 0, 40))$
```



Remember we are seeking a solution for which the $\text{cost}(x,y)$ is minimum subject to $x + y \geq 30$,

From the plot we can find a minimum for which $x + y = 30$, and $c < -500$. In principle, we could lock-in on this minimum by adding a few more cost curves.

If you remove the `wx` from `wxdraw` for the above figure and interactively read the coordinates as you move the cursor in the separate `gnuplot` window (see above), you can see that roughly, $x \sim 8.8$, $y \sim 21.6$.

```
(%i32) g : 30 - x - y;
```

```
gridSearch (c, [x, 8, 10, 0.1], [y, 20, 22, 0.1], min, g);
```

```
(g) -y - x + 30
```

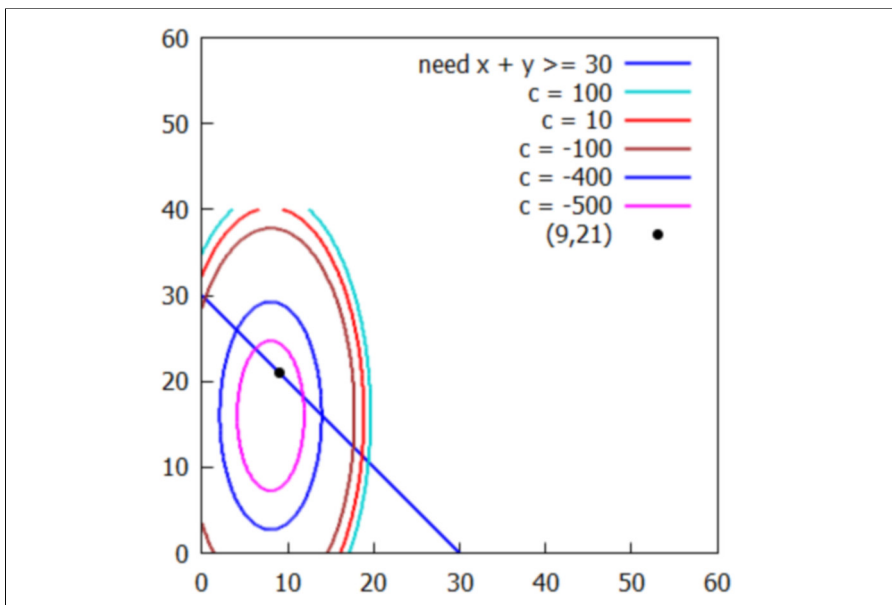
```
(%o32) [-546.0, [x=9.0, y=21.0]]
```

Next try KKT.

```
(%i33) KKT( c, min, g );
soln = [x=9,y=21,lam=10]  g(x,y) at soln = 0
objsub = -546
soln = [x=9.0,y=21.0,lam=10.0]  objsub = -546.0
(%o33) done
```

```
(%i34) wxdraw2d (yrange = [0,60], xrange = [0,60], proportional_axes = 'xy,
  key = "need x + y >= 30",explicit (30 - x,x,0,30),
  color = dark_turquoise, key = "c = 100", implicit (c = 100, x,0,40,y,0,40),
  color = red, key = "c = 10", implicit (c = 10, x,0,40,y,0,40),
  color = brown, key = "c = -100", implicit (c = -100, x,0,40,y,0,40),
  color = blue, key = "c = -400", implicit (c = -400, x,0,40,y,0,40),
  color = magenta, key = "c = -500", implicit (c = -500, x,0,40,y,0,40),
  color = black, point_type = filled_circle,
  key = "(9,21)", points( [[9, 21] ]))$
```

(%t34)



Solution analysis:

```
(%i37) L : c + lam*g;
[Lx, Ly, Llam] : [diff(L,x), diff(L,y), diff(L,lam)];
subst ( [x = 9, y = 21, lam = 10], [c, x + y, Lx, Ly, g] );
(L)  y2 - 32 y + lam (-y - x + 30) + 5 x2 - 80 x
(%o36) [10 x - lam - 80, 2 y - lam - 32, -y - x + 30]
(%o37) [-546, 30, 0, 0, 0]
```

We see this solution satisfies $g \leq 0$, since $g = 0$.

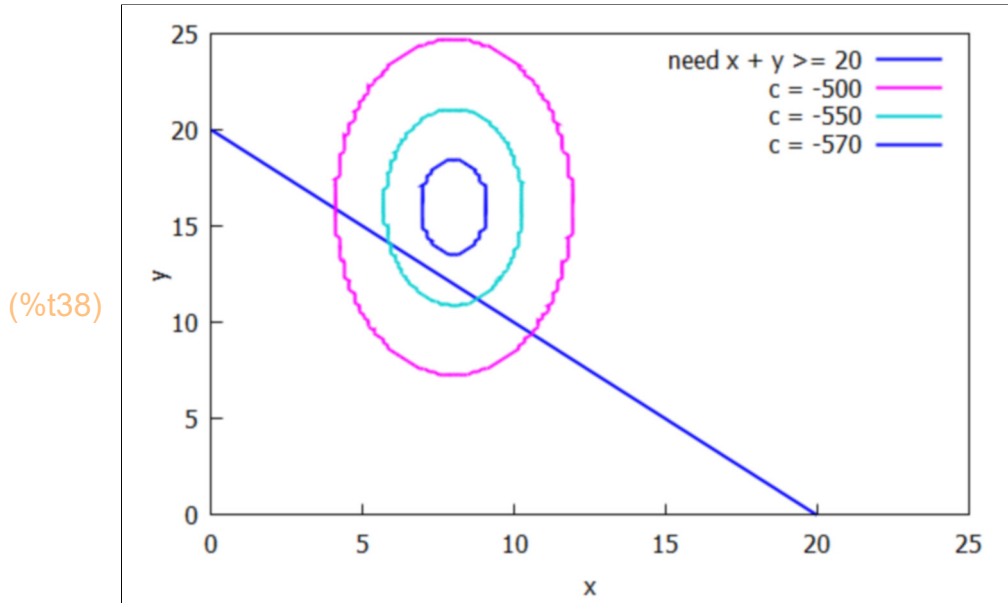
7 Dowling Prob. 13.38, Continuation Problem

Dowling Prob 13.38

Minimize the same cost function $c = 5x^2 - 80x + y^2 - 32y$ as in Prob 13.37 subject to the new constraint $x + y \geq 20$, enlarging the feasible set.

Now: $g(x,y) = 20 - x - y \leq 0 \Rightarrow 20 \leq x + y$.

```
(%i38) wxdraw2d (yrange = [0,25], xrange = [0,25], xlabel = "x",ylabel = "y",
  key = "need x + y >= 20",explicit (20 - x,x,0,20),
  color = magenta, key = "c = -500", implicit (c = -500, x,0,40,y,0,40),
  color = dark_turquoise, key = "c = -550", implicit (c = -550, x,0,30,y,0,30),
  color = blue, key = "c = -570", implicit (c = -570, x,0,40,y,0,40))$
```



We see from the plot that we can find a minimum < -570 for which $x + y > 20$.

Using the draw2d version in a separate gnuplot window and watching the coordinates of the cursor, we see roughly $x \sim 8$, $y \sim 16$.

Try using `gridSearch (... , min, g)` with the new $g = 20 - x - y \leq 0$

```
(%i40) g : 20 - x - y;
  gridSearch (c, [x, 7, 9, 0.1], [y, 15,17, 0.1], min, g);
```

```
(g) -y - x + 20
```

```
(%o40) [-576.0, [x=8.0, y=16.0]]
```

Next try KKT:

```
(%i41) KKT( c, min, g );
      constraint is non-binding
      soln = [ lam=0,x=8,y=16]  g(x,y) at soln = -4
      objsub = -576
      soln = [ lam=0.0,x=8.0,y=16.0]  objsub = -576.0
(%o41) done
```

With $\text{lam} = 0$, the constraint is nonbinding. The firm can minimize costs (using the optimal solution: $x = 8$ and $y = 16$) while exceeding its production quota of 20 (by 4 units).

Details of solution:

```
(%i44) L : c + lam*g;
      [Lx, Ly, Llam] : [diff(L,x), diff(L,y), diff(L,lam)];
      subst ( [x = 8, y = 16,lam = 0], [c, x + y, Lx, Ly, g] );
(L)    y^2 - 32 y + lam (-y - x + 20) + 5 x^2 - 80 x
(%o43) [ 10 x - lam - 80, 2 y - lam - 32, -y - x + 20 ]
(%o44) [-576, 24, 0, 0, -4]
```

We see that $x + y = 24 > 20$.

8 Chris Jones KKT One Constraint Example

Chris Jones,
Math. Dept, Univ. of Leeds
<http://www1.maths.leeds.ac.uk/~cajones/pubs.html>

Math 2640, Introduction to Optimization
<http://www1.maths.leeds.ac.uk/~cajones/math2640/MATH2640.html>

Sample KKT problem, pdf page 4:
<http://www1.maths.leeds.ac.uk/~cajones/math2640/notes4.pdf>

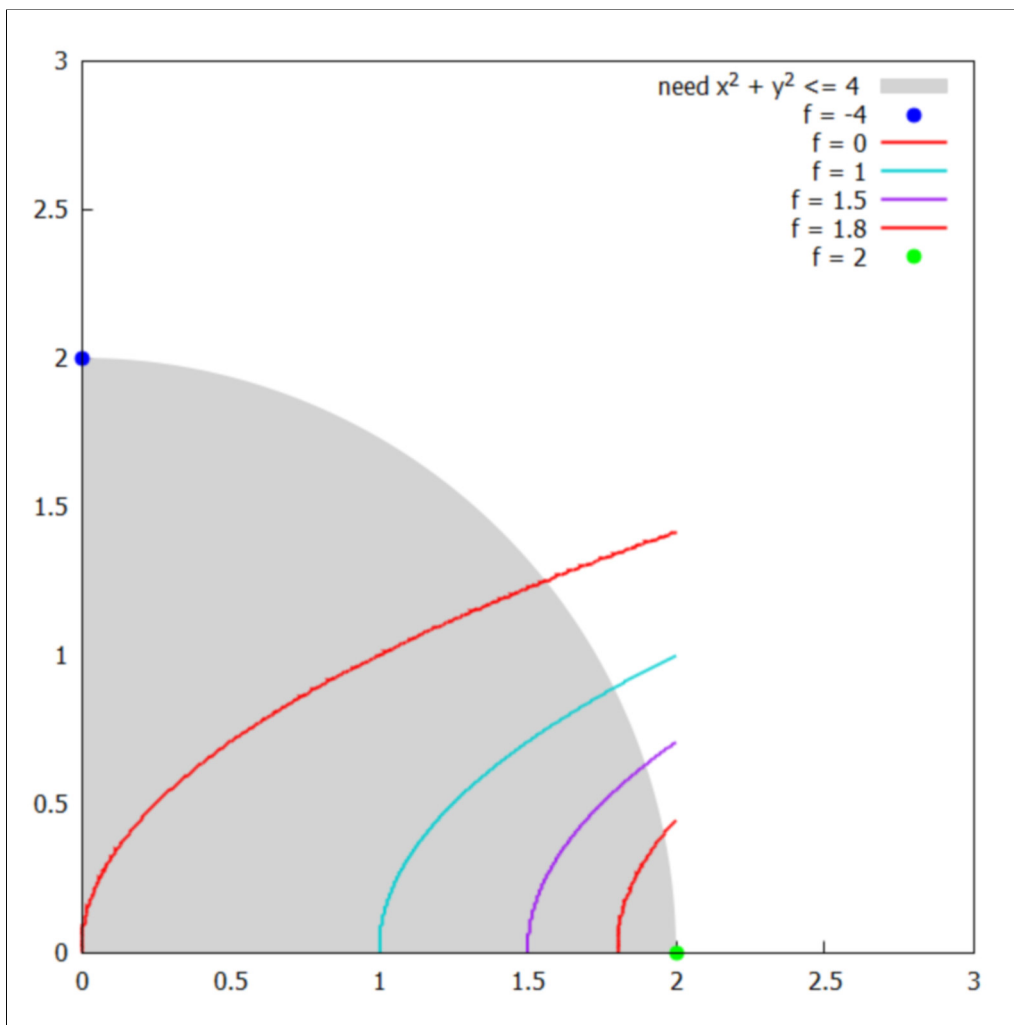
Maximize $f = x - y^2$, s.t. $x^2 + y^2 \leq 4$. i.e., s.t. $0 \leq 4 - x^2 - y^2$.
So define $g(x,y) = 4 - x^2 - y^2 \geq 0$.

```
(%i46) f : x - y^2;
      g : 4 - x^2 - y^2;
(f)    x - y^2
(g)    -y^2 - x^2 + 4
```


Make a plot of the feasible set of solution points (x,y) and curve of $f(x,y) = \text{constant}$.

```
(%i47) wxdraw2d (yrange = [0, 3], xrange = [0,3], proportional_axes = 'xy,
  key = " need  $x^2 + y^2 \leq 4$  ", filled_func = true, fill_color = light_gray,
  explicit (sqrt (4 -  $x^2$ ), x, 0, 2), filled_func = false,
  point_size = 1.5, point_type = filled_circle,
  color = blue, key = "f = -4", points ([ [0, 2] ]),
  color = red, key = "f = 0", implicit (f = 0, x,0,2,y,0,2),
  color = dark_turquoise, key = "f = 1", implicit (f = 1, x,0,2,y, 0,1),
  color = purple, key = "f = 1.5", implicit (f = 1.5, x,0,2,y,0,1),
  color = red, key = "f = 1.8", implicit (f = 1.8, x,0,2,y,0,2),
  color = green, key = "f = 2", points ([ [2,0] ])), wxplot_size = [680,680];
```

(%t47)



(%o47)

We see from the plot that the maximum of f that satisfies $x^2 + y^2 \leq 4$, $x \geq 0$, $y \geq 0$, is the corner solution $x = 2$, $y = 0$.

```
(%i48) gridSearch (f, [x,1.7, 2.3, 0.01], [y, -0.3, 0.3, 0.01], max, g);
```

```
(%o48) [2.0,[x=2.0,y=0.0]]
```

```
(%i49) KKT (f, max, g);
```

```
soln = [x=2,lam= $\frac{1}{4}$ ,y=0] g(x,y) at soln = 0
```

```
objsub = 2
```

```
soln = [x=2.0,lam=0.25,y=0.0] objsub = 2.0
```

```
(%o49) done
```

Check the first derivatives of the Lagrangian:

```
(%i52) L : f + lam*g;
```

```
[Lx, Ly, Llam] : [diff(L,x), diff(L,y), diff(L,lam)];
```

```
subst ([x = 2, y = 0, lam = 1/4], [f, x^2 + y^2, Lx, Ly, g] );
```

```
(L) -y^2+lam (-y^2-x^2+4) +x
```

```
(%o51) [1-2 lam x, -2 lam y-2 y, -y^2-x^2+4]
```

```
(%o52) [2,4,0,0,0]
```

The optimal solution returned from KKT lies on the constraining border (the constraint is binding) since $g = 0$.

9 Dowling Prob. 13.39; Maximize Utility, 2 Constraints

Maximize the utility function $u = x*y$ subject to the two constraints:

1. budget constraint: $3x + 4y \leq 144 \rightarrow 0 \leq 144 - 3x - 4y, \rightarrow 144 - 3x - 4y \geq 0$,

2. dietary constraint: $5x + 2y \leq 120 \rightarrow 120 - 5x - 2y \geq 0$,

as well as $x \geq 0, y \geq 0$.

To use $KKT(u, \max, g1, g2)$ we need $g1 \geq 0, g2 \geq 0$.

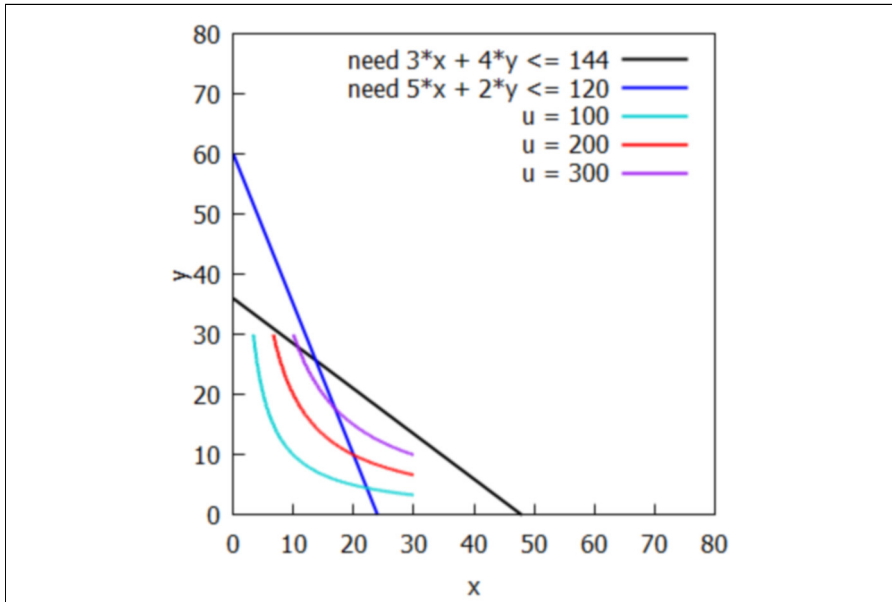
Make a plot of the two constraint borders, and show $u = \text{constant}$ curves (indifference curves).

```
(%i53) u : x*y;
```

```
(u) x y
```

```
(%i54) wxdraw2d (yrange = [0,80], xrange = [0,80], line_width = 2, proportional_axes = 'xy,
    xlabel = "x", ylabel = "y", /* key = "feasible set",
    filled_func = true, fill_color = light_gray, */
    color = black, key = "need 3*x + 4*y <= 144", explicit (36 - 3*x/4, x, 0, 48),
    color = blue, key = "need 5*x + 2*y <= 120", explicit (60 - 5*x/2, x, 0, 24),
    /* filled_func = false, */
    color = dark_turquoise, key = "u = 100", implicit (u = 100, x,0,30,y,0,30),
    color = red, key = "u = 200", implicit (u = 200, x,0,30,y,0,30),
    color = purple, key = "u = 300", implicit (u = 300, x,0,30,y,0,30))$
```

(%t54)



We see from the plot that the optimum (maximum) value of $u = x*y$ is found near the point where the constraint lines cross.

```
(%i55) soln : solve ([3*x + 4*y = 144, 5*x + 2*y = 120], [x, y ])[1];
```

```
(soln) [x = 96/7, y = 180/7]
```

```
(%i56) float(soln);
```

```
(%o56) [x = 13.714, y = 25.714]
```

```
(%i57) float (at (u, soln));
```

```
(%o57) 352.65
```

9.1 gridSearch (f, xR, yR, maxmin, g1, g2)

Use gridSearch (f(x,y), [x, x1, x2, dx], [y, y1, y2, dy], maxmin, g1, g2)
 in which if maxmin = max, then $g1(x,y) \geq 0$ and $g2(x,y) \geq 0$
 and if maxmin = min, then $g1(x,y) \leq 0$ and $g2(x,y) \leq 0$.

```
(%i59) g1 : 144 - 3*x - 4*y;
      g2 : 120 - 5*x - 2*y;
(g1)   -4 y - 3 x + 144
(g2)   -2 y - 5 x + 120

(%i60) gridSearch (u, [x, 13.5, 13.9, 0.01], [y, 25.4, 30, 0.01], max, g1, g2);
(%o60) [352.48, [x=13.71, y=25.71]]
```

We expect KKT to find this optimal solution with $g1 \geq 0$, $g2 \geq 0$.

```
(%i61) KKT (u, max, g1, g2);
      soln = [x = 96/7, y = 180/7, lam1 = 60/49, lam2 = 216/49]  g1 = 0  g2 = 0
      objsub = 17280/49
      soln = [x = 13.714, y = 25.714, lam1 = 1.2245, lam2 = 4.4082]  objsub =
      352.65
(%o61) done
```

Both constraints are binding ($g1 = 0$, $g2 = 0$) implies the optimum solution lies at the intersection of these boundary curves.

```
(%i64) L : u + lam1*g1 + lam2*g2;
      [Lx, Ly, L1, L2] : [diff(L,x), diff(L,y), diff(L,lam1), diff(L, lam2)];
      subst ([x = 96/7, y = 180/7, lam1 = 60/49, lam2 = 216/49], [u, Lx, Ly, L1, L2, g1, g2]);

(L)   x y + lam2 (-2 y - 5 x + 120) + lam1 (-4 y - 3 x + 144)
(%o63) [y - 5 lam2 - 3 lam1, x - 2 lam2 - 4 lam1, -4 y - 3 x + 144, -2 y - 5 x + 120
      ]
(%o64) [17280/49, 0, 0, 0, 0, 0, 0]

(%i65) float(%);
(%o65) [352.65, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

10 Chiang/Wainwright Ex. 1, Maximize Utility, 2 Constraints

Chiang & Wainwright, p. 406-408

Maximize the utility $u = x \cdot y$ subject to the budget constraint $x + y \leq 100$ and a "ration constraint" $x \leq 40$.

Let $g1 = 100 - x - y \geq 0$, and $g2 = 40 - x \geq 0$.

In this problem with a fixed upper limit on good x , it is useful to make a plot showing the region of feasible solution points (x,y) , the kink in the border of the feasible set, and a sample indifference curve $u(x,y) = u_0$.

```
(%i71) g1 : 100 - x - y;
      g2 : 40 - x;
      node1 : [x=0,y=100];
      node2 : solve ([g1,g2], [x,y])[1];
      node3 : [x = 40, y = 0];
      y1 : rhs (solve(g1,y)[1]);

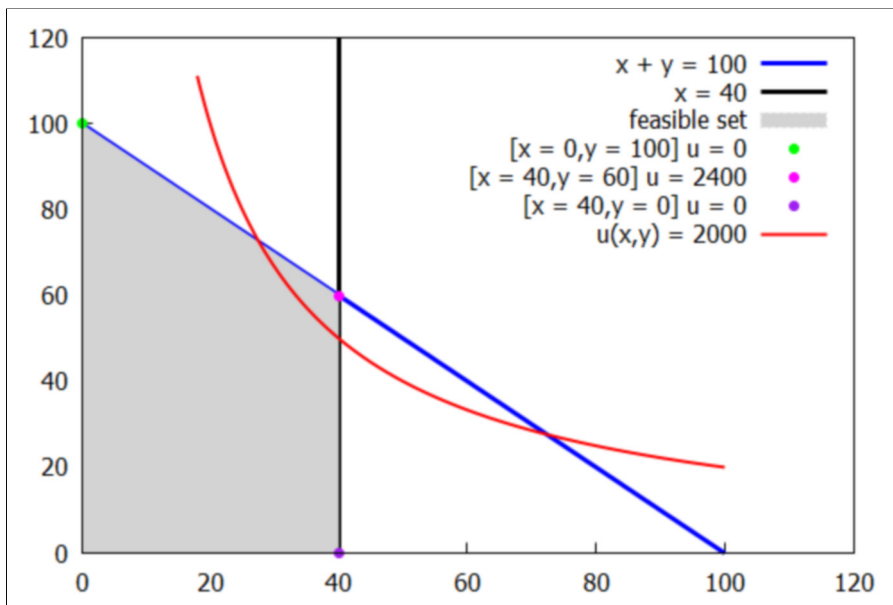
(g1)  -y - x + 100
(g2)  40 - x
(node1) [x=0,y=100]
(node2) [x=40,y=60]
(node3) [x=40,y=0]
(y1)  100 - x

(%i72) u(x,y) := x*y;
(%o72) u(x,y) := x y

(%i73) [u1,u2,u3] : map (lambda ([xx],at( u(x,y), xx)), [node1, node2, node3] );
(%o73) [0,2400,0]
```

```
(%i74) wxdraw2d(line_width = 3, xrange = [0,120],yrange = [0,120], key = "x + y = 100",
/* two bounding curves */
explicit (y1, x, 0, 100), color = black, key = "x = 40",
parametric (40, t, t, 0, 120),
filled_func = true, fill_color = light_gray, key = "feasible set",line_width = 1,
explicit (y1,x,0,40),
/* three nodes */
point_type = filled_circle, color = green,key = sconcat (node1," u = ",u1),
points ([ map('rhs, node1) ]),
key = sconcat (node2," u = ",u2), color = magenta,points ([ map('rhs, node2) ]),
key = sconcat (node3," u = ",u3), color = purple,points ([ map('rhs, node3) ]),
/* one u = constant curve */
color = red,key = "u(x,y) = 2000",filled_func = false, line_width = 2,
explicit (2000/x, x, 18, 100))$
```

(%t74)



```
(%i75) gridSearch (u,[x, 39,41,0.01],[y,59,61,0.01], max, g1, g2);
```

```
(%o75) [2400.0,[x=40.0,y=60.0]]
```

```
(%i76) KKT (u, max, g1, g2);
```

```
soln = [x=40,y=60,lam1=40,lam2=20] g1= 0 g2 = 0
```

```
objsub = 2400
```

```
soln = [x=40.0,y=60.0,lam1=40.0,lam2=20.0] objsub = 2400.0
```

```
(%o76) done
```

```
(%i77) soln : [x=40,y=60,lam1=40,lam2=20];
```

```
(soln) [x=40,y=60,lam1=40,lam2=20]
```

```
(%i80) L : u + lam1*g1 + lam2*g2;
      [Lx, Ly, L1, L2] : [diff (L, x), diff (L, y), diff (L, lam1), diff (L, lam2)];
      subst (soln, [ u,x+y, x, Lx, Ly,L1, L2, g1, g2]);

(L)   x y + lam1 (-y - x + 100) + lam2 (40 - x)
(%o79) [y - lam2 - lam1, x - lam1, -y - x + 100, 40 - x]
(%o80) [2400, 100, 40, 0, 0, 0, 0, 0, 0]
```

The optimal solution has $x = 40$, $x + y = 100$, $g1 = g2 = 0$, $u = 2400$.

11 Chiang/Wainwright Ex. 2, Minimize Cost, 2 Constraints

p. 410 - 411.

Minimize $C = (x - 4)^2 + (y - 4)^2$ subject to

1. $2x + 3y \geq 6$, $0 \geq 6 - 2x - 3y$
 $g1(x,y) = 6 - 2x - 3y \leq 0$
2. $-3x - 2y \geq -12$, $0 \geq -12 + 3x + 2y$,
 $g2(x,y) = -12 + 3x + 2y \leq 0$

Constraint #2 is equivalent to $3x + 2y \leq 12$, since if we multiply thru by (-1) we must reverse the direction of the inequality. (If $-5 > -12$ then $5 < 12$)

The cost function is a circle of radius \sqrt{C} with center at $(x=4, y=4)$. With a protractor, ruler and a piece of graph paper, draw the lines $2x + 3y = 6$, $3x + 2y = 12$, and circles of various radii centered at $(4,4)$; the cost is the radius squared. You will quickly see that the minimum cost corresponds to a point on the line $3x + 2y = 12$, with a cost of about 5.

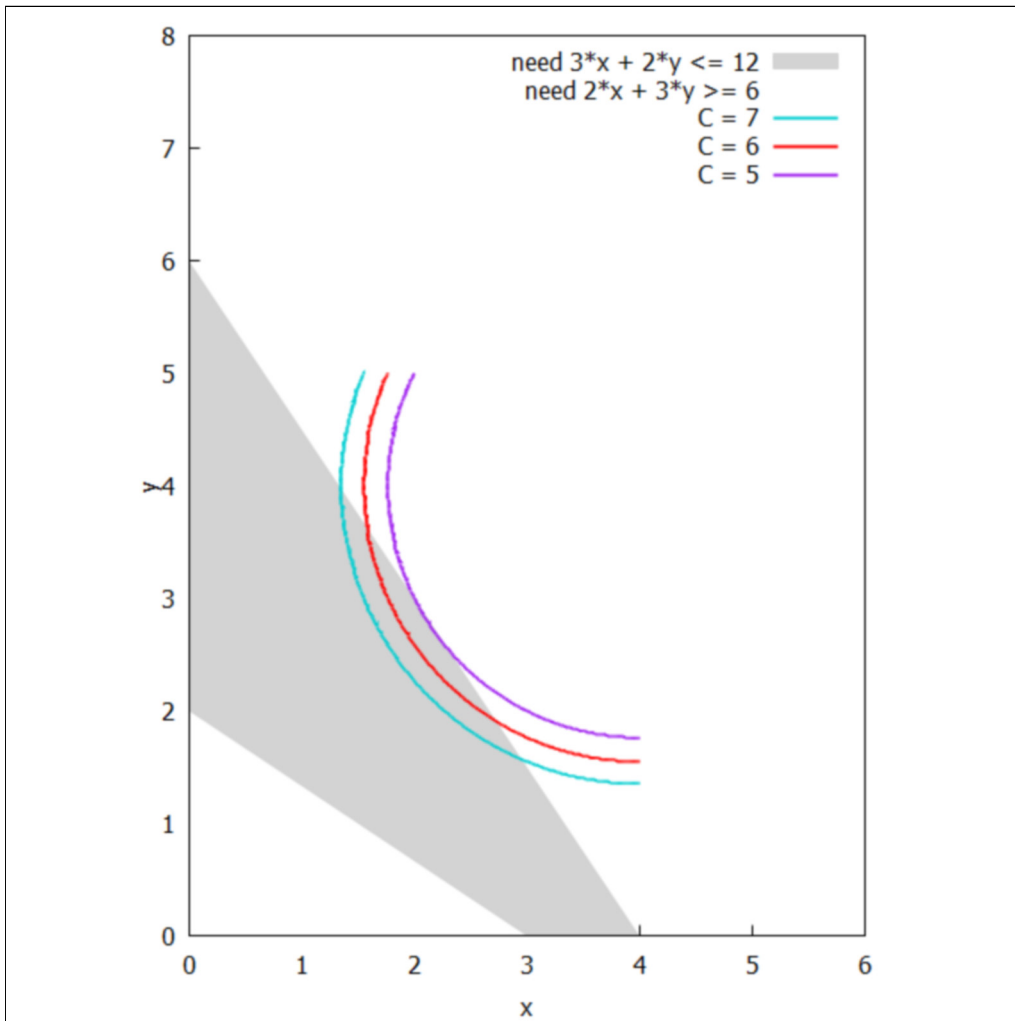
```
(%i83) C : (x - 4)^2 + (y - 4)^2;
      g1 : 6 - 2*x - 3*y;
      g2 : -12 + 3*x + 2*y;

(C)   (y-4)^2 + (x-4)^2
(g1)  -3 y - 2 x + 6
(g2)  2 y + 3 x - 12

(%i84) gridSearch (C,[x,2,2.4,0.01],[y,2.6,3,0.01],min, g1, g2);
(%o84) [4.9237, [x=2.14, y=2.79]]
```

```
(%i85) wxdraw2d (yrange = [0,8], xrange = [0,6], xlabel = "x", ylabel = "y",
    proportional_axes = 'xy, filled_func = true, fill_color = light_gray,
    color = green, key = "need 3*x + 2*y <= 12", explicit (6 - 3*x/2,x,0,4),
    fill_color = white, key = "need 2*x + 3*y >= 6", explicit (2 - 2*x/3,x,0,3),
    color = dark_turquoise, key = "C = 7", implicit (C = 7, x,1,4,y,1,5),
    color = red, key = "C = 6", implicit (C = 6, x,1,4,y,1,5),
    color = purple, key = "C = 5", implicit (C = 5, x,1,4,y,1,5)),wxplot_size = [680,680];
```

(%t85)



(%o85)

```
(%i86) KKT (C, min, g1, g2);
```

$$\text{soln} = [lam1=0, x=\frac{28}{13}, y=\frac{36}{13}, lam2=\frac{16}{13}] \quad g1 = -\frac{86}{13} \quad g2 = 0$$

$$\text{objsub} = \frac{64}{13}$$

$$\text{soln} = [lam1=0.0, x=2.1538, y=2.7692, lam2=1.2308] \quad \text{objsub} = 4.9231$$

```
(%o86) done
```



```
(%i90) soln : [x=28/13, y=36/13, lam1=0, lam2=16/13];
L : C + lam1*g1 + lam2*g2;
[Lx : diff (L, x), Ly : diff (L,y), L1 : diff (L, lam1), L2 : diff (L, lam2) ];
float (subst (soln, [ C, Lx, Ly,L1, L2, g1, g2]));

(soln) [x = 28/13, y = 36/13, lam1 = 0, lam2 = 16/13]

(L) lam2 (2 y + 3 x - 12) + lam1 (-3 y - 2 x + 6) + (y - 4)^2 + (x - 4)^2
(%o89) [2 (x - 4) + 3 lam2 - 2 lam1, 2 (y - 4) + 2 lam2 - 3 lam1, -3 y - 2 x + 6, 2 y
+ 3 x - 12]
(%o90) [4.9231, 0.0, 0.0, -6.6154, 0.0, -6.6154, 0.0]
```

The first constraint was $g_1 \leq 0$, or $2x + 3y \geq 6$. With $g_1 = -6.6$, this is not binding:

```
(%i91) at (2*x + 3*y, soln), numer;
(%o91) 12.615
```

The second constraint was $g_2 \leq 0$ or $-3x - 2y \geq -12$, and with $g_2 = 0$, is a binding constraint.

```
(%i92) at (-3*x - 2*y, soln), numer;
(%o92) -12.0
```

12 Maximize, 2 Constraints, Baldani, Bradfield, Turner

This example is from Baldani, Bradfield, and Turner, *Mathematical Economics*, 2nd. ed, P. 291 - 293.

Maximize $f(x,y) = x + y$ subject to $x \geq 0$, $y \geq 0$, $x + 2y \leq 5$, $2x + y \leq 6$.

Define $g_1(x,y)$ and $g_2(x,y)$ so $g_1 \geq 0$, $g_2 \geq 0$.

```
(%i95) f : x + y;
g1 : 5 - x - 2*y;
g2 : 6 - 2*x - y;

(f) y + x
(g1) -2 y - x + 5
(g2) -y - 2 x + 6
```

Where do the lines $g_1 = 0$, $g_2 = 0$ cross?

```
(%i96) cross : solve ([g1,g2],[x,y])[1];
```

```
(cross) [x= $\frac{7}{3}$ ,y= $\frac{4}{3}$ ]
```

```
(%i97) float(cross);
```

```
(%o97) [x=2.3333,y=1.3333]
```

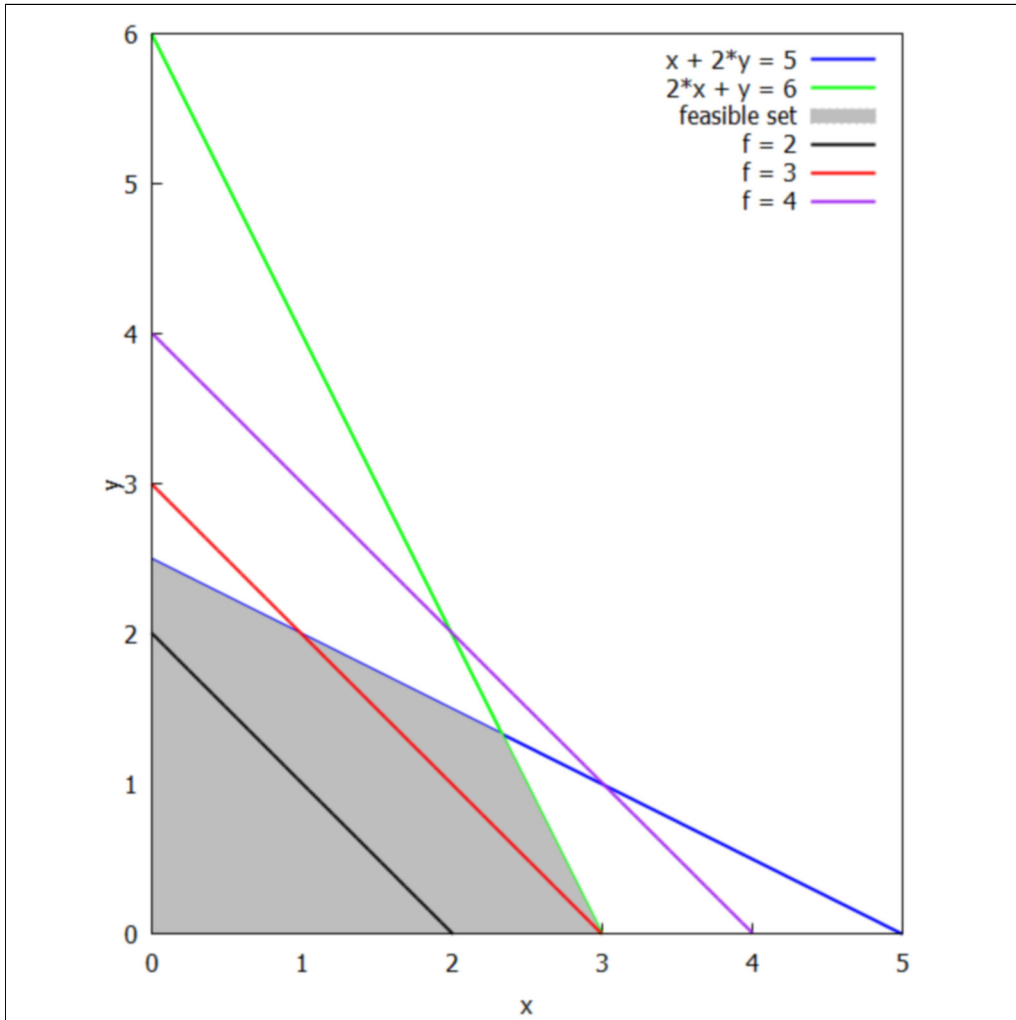
```
(%i98) float (at(f, cross));
```

```
(%o98) 3.6667
```

Graphical exploration; show feasible (x,y) set in light gray, and some curves $f = \text{constant}$.

```
(%i99) wxdraw2d(line_width = 2, xrange = [0,5], yrange = [0,6], xlabel = "x", ylabel = "y",
    proportional_axes = 'xy, key = "x + 2*y = 5", explicit (5/2 - x/2,x, 0, 5),
    color = green, key = "2*x + y = 6",
    explicit (6 - 2*x, x, 0, 3), key = "feasible set",
    filled_func = true, fill_color = gray,
    explicit (if x < 7/3 then 5/2 - x/2 else 6 - 2*x, x, 0, 3),
    color = black, key = "f = 2", implicit (f = 2, x,0,4,y,0,4),
    color = red, key = "f = 3", implicit (f = 3, x,0,4,y,0,4),
    color = purple, key = "f = 4", implicit (f = 4, x,0,4,y,0,4)),wxplot_size = [680,680];
```

(%t99)



(%o99)

```
(%i100) gridSearch (f,[x,2.1,2.4,0.01],[y,1.1,1.5,0.01], max, g1, g2 );
(%o100) [3.66,[x=2.34,y=1.32]]
```

```
(%i101) KKT (f, max, g1, g2);
```

$$\text{soln} = \left[x = \frac{7}{3}, y = \frac{4}{3}, \text{lam1} = \frac{1}{3}, \text{lam2} = \frac{1}{3} \right] \quad g1 = 0 \quad g2 = 0$$

$$\text{objsub} = \frac{11}{3}$$

$$\text{soln} = \left[x = 2.3333, y = 1.3333, \text{lam1} = 0.33333, \text{lam2} = 0.33333 \right]$$

$$\text{objsub} = 3.6667$$

```
(%o101) done
```