

Optimization, Part B

TABLE OF CONTENTS

PREFACE	1
REFERENCES	2
OPTIMIZATION CONSTRAINED BY EQUALITIES	2
INPUT- OUTPUT ANALYSIS	40
EIGENVALUE TESTS FOR SIGN DEFINITENESS	56
BINDING AND NONBINDING CONSTRAINTS IN ECON.	63

```
(%i4) 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$
```

```
(%i5) load ("Econ1.mac");
```

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

1 Preface

Dowling12B.wmxm uses Maxima to review the analysis of optimization of functions of economic variables which are constrained by equalities, following the second part of Ch. 12 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.

A code file Econ1.mac as available in the same section (of Economic Analysis with Maxima), which defines many Maxima functions used in this worksheet.

Use load ("Econ1.mac");

This worksheet is one of a number of wxMaxima files available in the section
Economic Analysis with Maxima
on my CSULB webpage.

Edwin L. (Ted) Woollett
<https://home.csulb.edu/~woollett/>
 June 13, 2022

Part A includes unconstrained optimization examples. In Part B the use of the bordered Hessian matrix for second order conditions in optimization with equality constraints is discussed, and also the use of eigenvalues determination of sign definiteness. Also discussed in some detail in Part B is the subject of Input-Output Analysis in an open economy.

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

2 *References*

Notes on a High School course on optimization methods with a large variety of approaches to the problem.

<https://web.stanford.edu/group/sisl/k12/optimization/#!index.md>

Course notes, problem sets, and Exams for Math 2640, Introduction to Optimization, Leeds Univ., Chris Jones, 2005

<http://www1.maths.leeds.ac.uk/~cajones/math2640/MATH2640.html>

Fundamental Methods of Mathematical Economics, Alpha C. Chiang and Keven Wainwright, 4th ed., 2005, McGraw-Hill

Wainwright's course notes at British Columbia Institute of Technology, Burnaby, British Columbia, Canada

<http://faculty.bcitbusiness.ca/kevinw/chiang/ChapterLectureNotes.htm>

Wainwright's 2007 Econ 331 course at Simon Fraser Univ.

<http://www.sfu.ca/~wainwrig/Econ331/331.htm>

Martin J. Gander, Math. Dept., Univ. of Geneva, Ch. 4, Optimization

<https://www.unige.ch/~gander/teaching/polycopie.pdf>

<https://www.unige.ch/~gander/>

Math 2070, Univ. of Sydney Optimization notes

<https://www.maths.usyd.edu.au/u/UG/IM/MATH2070/r/NLOptWC.pdf>

Wolfram summary of Mathematics methods for optimization.

<https://reference.wolfram.com/language/tutorial/ConstrainedOptimizationIntroduction.html>

3 *Optimization Constrained by Equalities [12.5]*

3.1 optimum (f, varL, h), optimum (f, varL, [h1, h2]), etc.

For one equality constraint $h = 0$, use optimum (f, varL, h).

If f depends on the economic variables x and y, varL = [x, y],

If f depends on the economic variables x, y, and z, varL = [x, y, z], etc.

For two equality constraints $h1 = 0$ and $h2 = 0$, use optimum (f, varL, [h1, h2]).

And so on for more equality constraints.

3.2 Bordered Hessian: Bhessian (L, gList, varList), BHtest (BH, m, n)

Assume the objective function f is a function of n variables (x_1, x_2, \dots, x_n), and we have m constraints of form $g(x_1, x_2, \dots, x_n) = c$, where c is a constant.

The bordered Hessian matrix BH can be calculated with the syntax

BH : Bhessian (L, gList, varList)

in which L is the Lagrangian, glist is (in general) a list of expressions like [g1, g2] related to the equality constraints in force, and varList is a list of the search variables [x,y,...].

For only one equality constraint, such as $g = c$, where g is a Maxima expression depending on search variables (x,y), c is a constant, and the objective function f depends also on (x,y), the Lagrangian function is

L : $f + \lambda^*(c - g)$ OR (for example) L : $f + \text{lam}^*(c - g)$,
and the bordered Hessian matrix is

BH : Bhessian (L, g, [x,y]) OR BH : Bhessian (L, [g], [x,y]).

To insert the Greek letter λ at any point, press Esc, then the letter l, and then press Enter. You can also use some ordinary name, such as lam, or lam1, for the Lagrangian multiplier.

If the optimization is subject to two equality constraints, for example $g_1 = c_1$, $g_2 = c_2$, we need two Lagrange multipliers to define L

L : $f + \lambda_1^*(c_1 - g_1) + \lambda_2^*(c_2 - g_2)$, OR (e.g.) L : $f + \text{lam1}^*(c_1 - g_1) + \text{lam2}^*(c_2 - g_2)$,
BH : Bhessian (L, [g1, g2], [x,y]).

For more constraints and/or more search variables, just enlarge the lists.

The Maxima function Bhessian is defined in Econ1.mac, and the above syntax produces a $(m + n) \times (m + n)$ square matrix which will involve the first derivatives of the m constraint functions g_1, g_2, \dots , and the second derivatives of the Lagrangian function L with respect to (wrt) the n search variables in varList. In the most general case this bordered Hessian matrix will be a function of all n search variables and all m lagrange multipliers λ_k .

In a general case, assume n search variables x_1, x_2, \dots, x_n , and m equality constraints $g_1 = c_1, g_2 = c_2, \dots, g_m = c_m$.

To test for a maximum or minimum, the Hessian matrix BH should first be evaluated at a critical point, which is a solution of the $m + n$ equations $\text{grad}(L) = 0$, where $\text{grad}(L)$ is the list of the first derivatives of the Lagrangian L with respect to (wrt) the $m+n$ "variables" $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m$.

Let BH_k stand for the "kth bordered leading principal minor" of a bordered Hessian matrix in which the smallest value of $k = m+1$, and the largest $k = n$.

You can calculate BH_k using our Maxima function $BLPM$,

$BH_k : BLPM (BH, m, k)$

which returns $LPM (BH, m + k)$, with $k = m + 1$ thru $k = n$. There are $(n - m)$ leading principal minors of a bordered Hessian matrix: beginning with $LPM (BH, 2m + 1)$ and ending with $LPM (BH, m + n)$, using our function LPM , which we discussed in Dowling12A.wmxm, Sec. "Leading Principal Minors of a Matrix LPM ".

But it is easiest instead to follow the path:

1. Calculate the Lagrange expression L .
2. Calculate the bordered Hessian matrix $BH : B_{\text{hessian}} (L, gList, varList)$.
3. Evaluate BH at a candidate critical point at which the first derivatives of L wrt the search variables and the Lagrangian multipliers are all simultaneously equal to zero.
- 4 Call $BH_{\text{test}} (BH, m, n)$, in which $m =$ number of equality constraints, $n =$ number of search variables, and BH corresponds to the desired critical point.

$BH_{\text{test}} (BH, m, n)$ returns a list of the $n - m$ leading principal minors of the bordered Hessian matrix BH , beginning with $LPM (BH, 2m+1)$ and ending with $LPM (BH, m + n) = \det BH$, and if the relevant criteria set out below are met, prints out the nature of the critical point used to generate BH .

Let $\text{sgn}(x) = \text{sign}(x)$. For example, $\text{sgn}(-2) = -1$, $\text{sgn}(6) = +1$.

The number of relevant leading principal minors to check is $n - m$, in which $n = \text{number of variables}$, $m = \text{number of equality constraints}$. $k_{\min} = 2^*m+1$, $k_{\max} = m+n$. In ascending order, these are the determinants (leading principal minors)

$\text{LPM}_k == \text{LPM}(\text{BH}, k)$ with $k = k_{\min}, k_{\min}+1, \dots, k_{\max}$,
in which $k_{\min} = 2^*m+1$, $k_{\max} = m+n$.

If it happens that $k_{\min} = k_{\max}$ (ie., $n - m = 1$), then there is only one determinant to find: $\text{LPM}(\text{BH}, m+n)$ which is the same as determinant (BH) .

However, $\text{LPM}(\text{BH}, m+n)$ is always the same as determinant (BH) .

IN SUMMARY, for $n = \text{length}(\text{varList})$, $m = \text{length}(\text{gList})$, $k = 2^*m+1, 2^*m+2, \dots, m+n$,

sufficient conditions (but not necessary) for a relative MAXIMUM are:

- (a) $\text{sgn}(\det \text{BH}) == \text{sgn}(\text{LPM}(\text{BH}, m+n)) = (-1)^n$,
- (b) $\text{sgn}(\text{LPM}_k)$ alternates for successive k 's in the range (k_{\min}, k_{\max}) ,

sufficient conditions (but not necessary) for a relative MINIMUM are:

- (a) $\text{sgn}(\det \text{BH}) == \text{sgn}(\text{LPM}(\text{BH}, m+n)) = (-1)^m$,
- (b) $\text{sgn}(\text{LPM}_k)$ remains the same as $(-1)^m$ for all k (no sign oscillation).

3.3 Chiang-Wainwright, Ch.12, Example 1

Find the extremum of $f = x*y$ subject to $x + y = 6$. For this example $m = 1$ and $n = 2$. Hence $k_{\min} = 2^*m + 1 = 3$, and $k_{\max} = m + n = 3$, so there is only one leading principal minor, $\text{LPM}_3 = \text{LPM}(\text{BH}, 3) == \text{determinant}(\text{BH})$, of the bordered Hessian matrix BH to check.

For this problem, the sign oscillation criteria play no role.

$(-1)^m = (-1)^1 = -1$, so we have a local minimum if $\text{LPM}_3 < 0$.

$(-1)^n = (-1)^2 = +1$, so we have a local maximum if $\text{LPM}_3 > 0$.

```
(%i7) f : x*y;
      g : x + y;
(f)   x y
(g)   y+x
```

3.3.1 Graphical Exploration

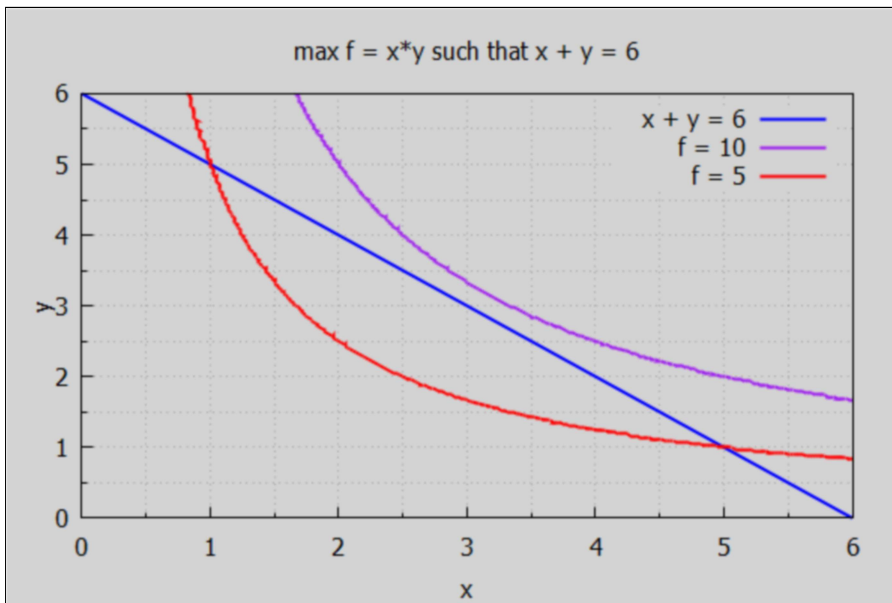
In the x - y plane, a solution (critical) point (x,y) must lie on the line $x + y = 6$. For what value of the objective function $f = x*y$ can this occur? We first draw the line $x+y = 6$ ($y = 6 - x$) in blue, and then plot the locus of the points $f = x*y$ for both $f = 5$ and $f = 10$ for orientation. For the latter two curves, we use draw2d's function

```
implicit (F(x,y) = C, x, x1, x2, y, y1, y2),
```

which tries to draw all points (x,y) for which $F(x,y) = C$ in the ranges provided.

```
(%i10) xmax : 6$ ymax : 6$
wxdraw2d ( xlabel = "x", ylabel = "y",
title = "max f = x*y such that x + y = 6",
key = " x + y = 6", explicit (6 - x, x, 0, xmax),
color = purple, key = " f = 10", implicit ( f = 10, x, 0, xmax, y, 0, ymax),
color = red, key = " f = 5", implicit ( f = 5, x, 0, xmax, y, 0, ymax))$
```

(%t10)



This graphical exploration suggests perhaps $f = 9$, with $x = 3$, $y = 3$ roughly. We could add another implicit curve corresponding to $f = 9$ to tie this down. Instead we use our friend `optimum (func, varList, constraint)`.

3.3.2 optimum (f, [x,y], 6 - g)

```
(%i11) optimum (f, [x,y], 6 - g);
      lagrangian = (x - lam1) y - lam1 x + 6 lam1
      solve returns [[x=3,y=3,lam1=3]]
      optimum only evaluates real non-negative solutions
      soln = [x=3,y=3,lam1=3]  objsub = 9
      soln = [x=3.0,y=3.0,lam1=3.0]  objsub = 9.0
      relative maximum
      LPM's = [LPM3=2.0]
(%o11) done
```

Since $LPM3 = 2 > 0$ we have a relative maximum.

optimum creates a global list cp.

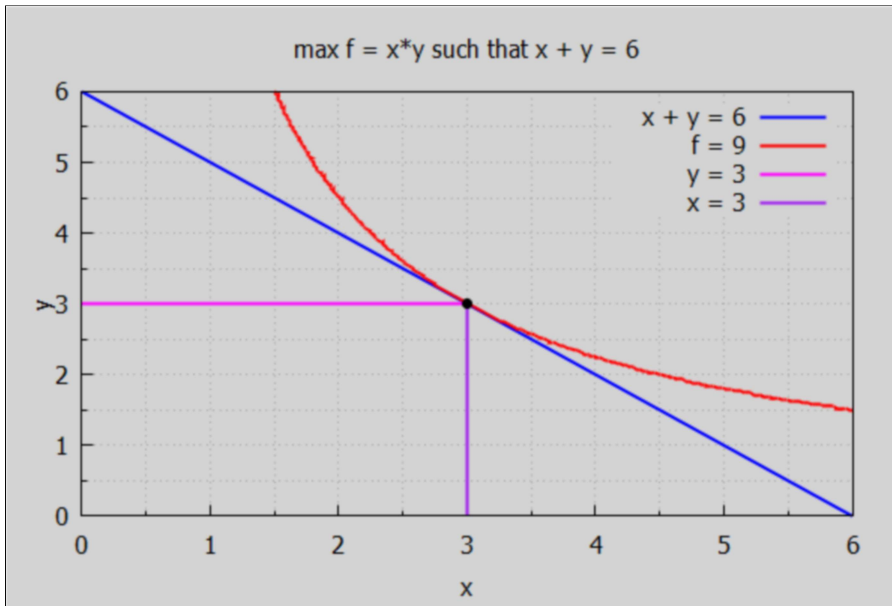
```
(%i12) cp;
(%o12) [[x=3,y=3,lam1=3]]

(%i13) cp : cp[1];
(cp)   [x=3,y=3,lam1=3]

(%i14) at (f, cp);
(%o14) 9
```

Show the curve $f = 9$ together with the constraint line $x + y = 6$ (i.e., $y = 6 - x$).

```
(%i17) xmax : 6$ ymax : 6$
wxdraw2d ( xlabel = "x", ylabel = "y",
title = "max f = x*y such that x + y = 6",
key = " x + y = 6", explicit (6 - x, x, 0, xmax),
color = red, key = " f = 9", implicit ( f = 9, x, 0, xmax, y, 0, ymax),
color = magenta, key = "y = 3", explicit (3, x, 0, 3),
color = purple, key = "x = 3", parametric (3, yy, yy, 0, 3),
color = black, key = "", points ([[3,3]]))$
```



(%t17)

3.3.3 Do it "by hand"...

Form the Lagrangian function

$$L = f + \lambda (6 - g) = f + \lambda (6 - x - y),$$

and look for the simultaneous solutions to the three first derivatives (wrt x , y , and λ) set equal to zero (i.e., the critical points). The jacobian function offers a fast route to a list of these first derivatives, although you have to use `stuff[1]` to get the list we want.

```
(%i18) [f, g];
```

```
(%o18) [x y, y+x]
```

```
(%i21) L : f + λ*(6 - g);
```

```
gradL : jacobian ([L], [x, y, λ]) [1];
```

```
solns : solve (gradL, [x, y, λ]);
```

```
(L) (-y-x+6) λ+x y
```

```
(gradL) [y-λ, x-λ, -y-x+6]
```

```
(solns) [[x=3, y=3, λ=3]]
```

```
(%i22) soln : solns[1];
```

```
(soln) [x=3, y=3, λ=3]
```


The same solution as found by optimum.

```
(%i23) at (f, soln);
```

```
(%o23) 9
```

The bordered Hessian matrix BH for one equality constraint $g(x_1, x_2, \dots, x_n)$ can be calculated using Bhessian (Lagrange-func, g, [x1,x2,...,xn]). Bhessian is defined in Econ1.mac.

```
(%i24) BH : Bhessian (L, g, [x,y]);
```

```
(BH) 
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

```

For this example, BH (Bhessian) is already purely numerical, so we can skip the step of evaluating it at the critical values of x , y , and λ . The second derivatives of the Lagrangian function L (wrt x and y) appear in the 2×2 block in the lower right corner. The first derivatives of g wrt x and y appear on the borders.

With $m = 1$ and $n = 2$, $(-1)^m = -1$ and $(-1)^n = +1$. $2^*m + 1 = 3$, $m + n = 3$, so the single leading principal minor we need to check is $LPM3 == LPM(BH, 3)$.

```
(%i25) LPM (BH, 3);
```

```
(%o25) 2
```

Since $LPM3 = 2$ is positive and has the same sign as $(-1)^n$, the critical point found ($x = 3$, $y = 3$) is a relative maximum.

Our Maxima function BHtest (BH, m, n) returns a list of all relevant leading principal minors of the bordered Hessian matrix BH, and if the relevant criteria set out above are met, will print out the nature of the critical point used to generate BH.

```
(%i26) BHtest (BH, 1, 2);
```

```
relative maximum
```

```
(%o26) [LPM3=2]
```

$LPM(BH,3)$ is defined as the determinant of the submatrix of BH formed by deleting all but the third row and all but the third column, but since BH is a 3×3 matrix this submatrix is just BH. So we could have just calculated the determinant directly.

```
(%i27) determinant (BH);
```

```
(%o27) 2
```

Dowling makes use of the language: "kth bordered leading principal minor" , which can be calculated using our Maxima function BLPM (BH, m, k), where m = number of constraints, and k takes values from $k_{min} = m + 1$ up to $k_{max} = n$.

For our case, $m = 1$, $k_{min} = 2$, $k_{max} = n = 2$, so there is just the determinant BLPM (BH, 1, 2) --> LPM (BH, 3).

In this paragraph, let Hb stand for "H bar", and Hb2 for the "second bordered leading principal minor". Dowling uses the overbar symbol (over H) for his bordered Hessian matrix, but in the absence of an overbar symbol in wxMaxima I will use Hb. Then Dowling's notation is |Hbk| for the kth bordered leading principal minor. BLPM is defined in Econ1.mac.

```
(%i28) Hb2 : BLPM (BH, 1, 2);
```

```
(Hb2) 2
```

3.4 Ch. 12, Example 5

Optimize the function $f = 4x^2 + 3xy + 6y^2$ subject to the constraint $x + y = 56$.

```
(%i30) g : x + y;
      f : 4*x^2 + 3*x*y + 6*y^2;
```

```
(g) y+x
```

```
(f) 6 y^2+3 x y+4 x^2
```

Here $m = 1$, $n = 2$, $m+n = 3$, $2*m+1 = 3$, check the sign of the leading principal minor LPM3.

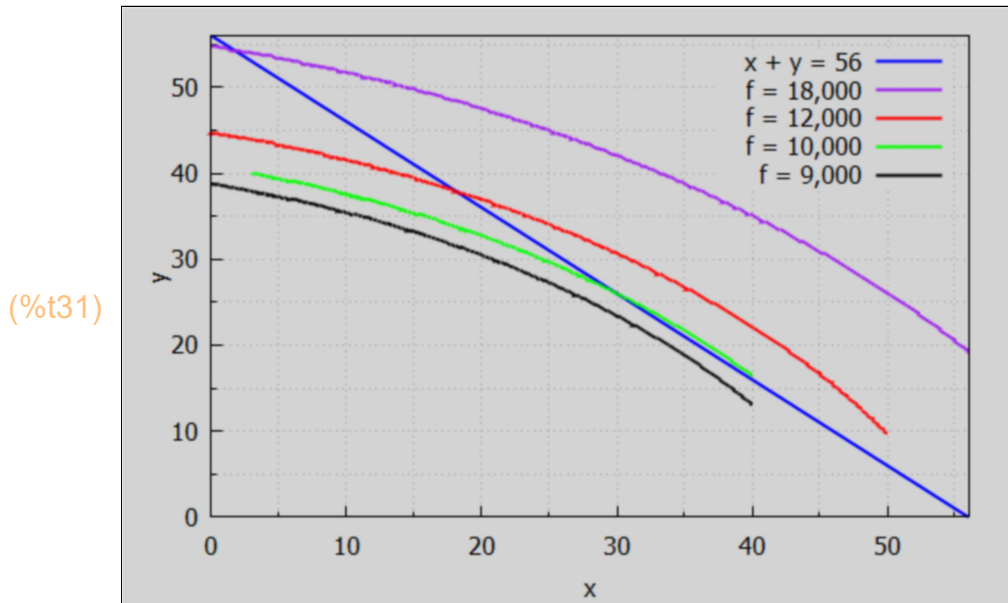
$(-1)^m = (-1)^1 = -1$, so local minimum if $LPM3 < 0$.

$(-1)^n = (-1)^2 = +1$, so local maximum if $LPM3 > 0$.

3.4.1 Graphical Exploration

Given that x and y have to satisfy $x + y = 56$ (and live on the blue line below) we are looking for the point that will yield either a minimum or a maximum value for f . We plot some $f = \text{constant}$ curves using draw2d's implicit function.

```
(%i31) wxdraw2d ( xlabel = "x", ylabel = "y",
  key = " x + y = 56", explicit (56 - x, x, 0, 56),
  color = purple, key = " f = 18,000", implicit ( f = 18000, x, 0, 56,y,0,56),
  color = red, key = " f = 12,000", implicit ( f = 12000, x, 0, 50,y,0,50),
  color = green, key = " f = 10,000", implicit ( f = 10000, x, 0, 40,y,0,40),
  color = black, key = " f = 9,000", implicit ( f = 9000, x, 0, 40,y,0,40))$
```



3.4.2 optimum (f, [x,y], 56 - g)

```
(%i32) optimum (f, [x,y], 56 - g);
```

lagrangian = $6y^2 + (3x - lam_1)y + 4x^2 - lam_1x + 56lam_1$

solve returns $[[x=36, y=20, lam_1=348]]$

optimum only evaluates real non-negative solutions

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

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

relative minimum

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

```
(%o32) done
```

Since $LPM3 < 0$ we have a relative minimum at $(x = 36, y = 20)$.

optimum creates the global list cp.

```
(%i33) cp;
```

```
(%o33)  $[[x=36, y=20, lam_1=348]]$ 
```

```
(%i34) soln : cp[1];
(soln) [x=36,y=20,lam1=348]
```

```
(%i35) soln : at (soln, lam[1] = λ);
(soln) [x=36,y=20,λ=348]
```

```
(%i36) at (f, soln);
(%o36) 9744
```

3.4.3 Do it "by hand"....

```
(%i37) [f, g];
(%o37) [6 y2+3 x y+4 x2, y+x]
```

```
(%i40) L : f + λ*(56 - g);
gradL : jacobian ([L], [x, y, λ])[1];
solns : solve (gradL, [x, y, λ]);
(L) (-y-x+56) λ+6 y2+3 x y+4 x2
(gradL) [-λ+3 y+8 x, -λ+12 y+3 x, -y-x+56]
(solns) [[x=36,y=20,λ=348]]
```

```
(%i41) soln : solns[1];
(soln) [x=36,y=20,λ=348]
```

BH is the bordered Hessian matrix appropriate to this problem.

```
(%i42) BH : Bhessian (L, g, [x,y]);
(BH) 
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 8 & 3 \\ 1 & 3 & 12 \end{pmatrix}$$

```

Since BH is already purely numerical, we again skip the step of evaluating BH at the critical point revealed by the first order conditions (FOC).

This is another case of $m = 1$ equality constraints and $n = 2$ variables. So there is only one leading principal minor to check, $LPM3 == LPM(BH,3)$.

```
(%i43) BHtest (BH, 1, 2);
relative minimum
(%o43) [LPM3=-14]
```

(%i44) LPM (BH,3);

(%o44) -14

(%i45) determinant (BH);

(%o45) -14

Since $m = 1$, $(-1)^m = -1$, and $LPM(BH, 3) < 0$, the sufficient conditions for a relative minimum have been met.

3.5 Prob. 12.28, Optimization of $f(x, y, z)$ with one constraint

Use the bordered Hessian to check the second order conditions in Prob. 5.12(c), where $f = 4x + y + z^2$ was optimized subject to $x + y + z = 56$. This is a $m = 1$, $n = 3$ problem.

$k_{min} = 2m + 1 = 3$, $k_{max} = m + n = 4$, check $LPM(BH,3)$ and $LPM(BH,4)$.

$(-1)^m = -1$; local minimum if $LPM4 < 0$, $LPM3 < 0$.

$(-1)^n = -1$; local maximum if $LPM4 < 0$, $LPM3 > 0$.

A "graphical exploration" is difficult for $n = 3$ variables. So proceed to the bordered Hessian test.

```

(%i48) f : 4*x*y*z^2;
      g : x + y + z;
      optimum (f, [x,y,z], 56 - g);

(f) 4 x y z^2
(g) z+y+x

lagrangian = 4 x y z^2 - lam_1 z - lam_1 y - lam_1 x + 56 lam_1
solve returns [[x=%r1,y=56-%r1,z=0,lam_1=0],[x=14,y=14,z=28,lam_1
=43904],[x=0,y=0,z=56,lam_1=0],[x=28,y=28,z=0,lam_1=0]]
optimum only evaluates real non-negative solutions
-----
i = 1 soln = [x=%r1,y=56-%r1,z=0,lam_1=0] objsub = 0
soln = [x=%r1,y=56.0-1.0 %r1,z=0.0,lam_1=0.0] objsub = 0.0
indefinite
LPM's = [LPM3=0.0,LPM4=0.0]
-----
i = 2 soln = [x=14,y=14,z=28,lam_1=43904] objsub = 614656
soln = [x=14.0,y=14.0,z=28.0,lam_1=4.3904 10^4] objsub = 6.1466 10^5
relative maximum
LPM's = [LPM3=6272.0,LPM4=-1.9669 10^7]
-----
i = 3 soln = [x=0,y=0,z=56,lam_1=0] objsub = 0
soln = [x=0.0,y=0.0,z=56.0,lam_1=0.0] objsub = 0.0
indefinite
LPM's = [LPM3=2.5088 10^4,LPM4=1.5735 10^8]
-----
i = 4 soln = [x=28,y=28,z=0,lam_1=0] objsub = 0
soln = [x=28.0,y=28.0,z=0.0,lam_1=0.0] objsub = 0.0
indefinite
LPM's = [LPM3=0.0,LPM4=0.0]

(%o48) done

(%i49) cp;
(%o49) [[x=%r1,y=56-%r1,z=0,lam_1=0],[x=14,y=14,z=28,lam_1=
43904],[x=0,y=0,z=56,lam_1=0],[x=28,y=28,z=0,lam_1=0]]

```

The only physical solution which has x, y, z, f all positive is the second solution in this list.

```
(%i50) soln : cp[2];
```

```
(soln) [x=14,y=14,z=28,lam1=43904]
```

```
(%i51) soln : at (soln, lam[1] = λ);
```

```
(soln) [x=14,y=14,z=28,λ=43904]
```

```
(%i52) at (f, soln);
```

```
(%o52) 614656
```

LPM4 < 0, and LPM3 > 0, so sufficient conditions for a relative maximum.

"By hand" solution steps: start with FOC (first order conditions).

```
(%i53) [f, g];
```

```
(%o53) [4 x y z2, z+y+x]
```

```
(%i56) L : f + λ*(56 - g);
```

```
gradL : jacobian ([L], [x, y, z, λ])[1];
```

```
solns : solve (gradL, [x, y, z, λ]);
```

```
(L) (-z-y-x+56) λ+4 x y z2
```

```
(gradL) [4 y z2-λ, 4 x z2-λ, 8 x y z-λ, -z-y-x+56]
```

```
(solns) [[x=%r2,y=56-%r2,z=0,λ=0],[x=14,y=14,z=28,λ=43904],[x=0,y=0,z=56,λ=0],[x=28,y=28,z=0,λ=0]]
```

The solutions for which $f = 0$ and $\lambda = 0$ do not involve the enforcement of the constraint, so the only physical solution is the second list.

```
(%i57) soln : solns[2];
```

```
(soln) [x=14,y=14,z=28,λ=43904]
```

```
(%i58) cp : rest (soln, -1);
```

```
(cp) [x=14,y=14,z=28]
```

The value of the objective function f at the given critical point is the same as the value of the Lagrangian function L at the critical point, since the expression which multiplies λ is zero at the critical point.

```
(%i59) at (f, cp);
```

```
(%o59) 614656
```

```
(%i60) at (L, cp);
```

```
(%o60) 614656
```

Second order conditions (SOC).

Construct the bordered Hessian matrix using Bhessian

```
(%i61) BH : Bhessian(L, g, [x, y, z]);
```

$$(BH) \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 4z^2 & 8yz \\ 1 & 4z^2 & 0 & 8xz \\ 1 & 8yz & 8xz & 8xy \end{pmatrix}$$

Convert to a purely numerical matrix by evaluating BH at the critical point. We can use either soln or cp for this, since the symbol λ does not appear in this matrix.

```
(%i62) BH : at (BH, cp);
```

$$(BH) \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 3136 & 3136 \\ 1 & 3136 & 0 & 3136 \\ 1 & 3136 & 3136 & 1568 \end{pmatrix}$$

Call BHtest (BH, m, n) for m = 1 constraint, and n = 3 variables.

```
(%i63) BHtest (BH, 1, 3);
```

```
relative maximum
```

```
(%o63) [LPM3=6272, LPM4=-19668992]
```

With n = 3, $(-1)^n = -1$, $\det BH < 0$, signs alternate, so sufficient condition for a relative maximum.

3.6 Ch. 12, Example 6, Cobb-Douglas Model

Extend the analysis of Example 10 in Dowling Ch. 6, Sec 9, to determine whether the critical point found is a relative maximum or minimum.

The Cobb-Douglas model can be used to relate the quantity q of units which can be produced as a function of the number of units of capital K and the number of units of labor L ,

$$q = A \cdot K^\alpha \cdot L^\beta$$

in which $\alpha > 0$ is the "output elasticity of capital", $0 < \beta < 1$ is the "output elasticity of labor", and $A > 0$ is an "efficiency parameter" measuring the level of technology employed.

With P_K the price per unit of capital K and P_L the price per unit of labor L , the cost of production of q units is

$$\text{cost} = P_K \cdot K + P_L \cdot L.$$

Assume the budget constraint is \$108 and $P_K = 3$ \$/unit of capital, $P_L = 4$ \$/unit of labor, and we want to optimize $q = K^{(0.4)} \cdot L^{(0.5)}$ subject to the constraint that

$$3 \cdot K + 4 \cdot L = 108.$$

This is a $m = 1$, $n = 2$ problem. $2m + 1 = 3$, $m + n = 3$, so only $LPM(BH,3)$ needs to be checked. $(-1)^m = -1$ so we have a local minimum if $LPM3 < 0$.

$(-1)^n = +1$, so we have a local maximum if $LPM3 > 0$.

3.6.1 optimum (q , [K , L], 108 - g)

solve will be more likely to find solutions if we simplify the form of q using ratsimp.

```
(%i64) [K, L];
```

```
(%o64) [K, (-z-y-x+56) lambda+4 x y z^2]
```

```
(%i67) kill(L)$
```

```
g : 3*K + 4*L;
```

```
q : ratsimp (K^(0.4)*L^(0.5));
```

```
(g) 4 L+3 K
```

```
(q) K^(2/5)*sqrt(L)
```

(%i68) optimum (q, [K,L], 108 - g);

lagrangian = $-4 \text{ lam}_1 L + K^{2/5} \sqrt{L} - 3 \text{ lam}_1 K + 108 \text{ lam}_1$

solve returns $[[K=16, L=15, \text{lam}_1 = \frac{8^{1/5}}{4\sqrt{15}}]]$

optimum only evaluates real non-negative solutions

soln = $[K=16, L=15, \text{lam}_1 = \frac{8^{1/5}}{4\sqrt{15}}]$ objsub = $\sqrt{15} 16^{2/5}$

soln = $[K=16.0, L=15.0, \text{lam}_1=0.097839]$ objsub = 11.741

relative maximum

LPM's = $[LPM3=0.52833]$

(%o68) done

(%i69) cp;

(%o69) $[[K=16, L=15, \text{lam}_1 = \frac{8^{1/5}}{4\sqrt{15}}]]$

(%i70) soln : cp[1], numer;

(soln) $[K=16, L=15, \text{lam}_1=0.097839]$

(%i72) Kmax : at (K, soln);

Lmax : at (L, soln);

(Kmax) 16

(Lmax) 15

(%i73) qmax : at (q, soln), numer;

(qmax) 11.741

(%i74) sconcat ("q = ", qmax);

(%o74) q = 11.741

(%i75) grind(%);

"q = 11.741"\$

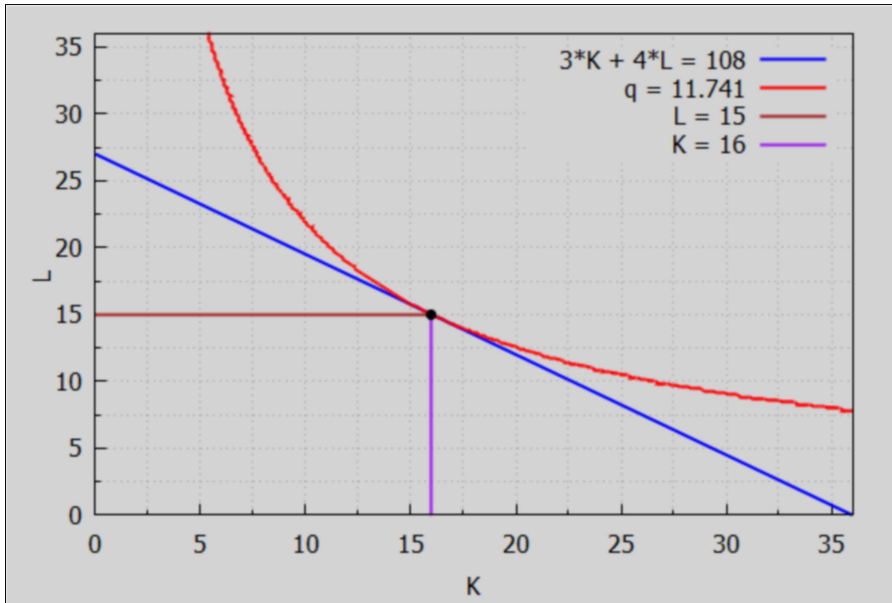
(%o75) done

(%i76) string(qmax);

(%o76) 11.741

```
(%i77) wxdraw2d ( xlabel = "K", ylabel = "L",
  key = " 3*K + 4*L = 108", explicit ((108 - 3*K)/4, K, 0, 36),
  color = red, key = sconcat ( " q = ", qmax), implicit ( q = qmax, K, 0, 36, L, 0, 36),
  color = brown, key = "L = 15", explicit (Lmax, K, 0, Kmax),
  color = purple, key = "K = 16", parametric (Kmax, yy, yy, 0, Lmax),
  color = black, key = "", points ([ [Kmax, Lmax] ]))$
```

(%t77)



3.6.2 Do it "by hand"

We include the budget constraint by defining the "Lagrange function"

$$Q = q + \lambda (108 - 3 K - 4 L),$$

in which λ is the "Lagrange multiplier", and we are constraining the optimization by requiring that $g = 3 K + 4 L = 108$. We then solve for the values of K , L , and λ for which Q is optimized.

We need to use a symbol different from L for the Lagrange function, since we are using L for one of the search variables in the objective function q and the constraint g .

Let Q be the Lagrange function. We previously used L as the symbol for the Lagrangian expression, hence use of `kill(L)` above.

This is a $m = 1$, $n = 2$ problem. $2 m + 1 = 3$, $m + n = 3$, so only $LPM(BH,3)$ needs to be checked. $(-1)^m = -1$ so we have a local minimum if $LPM3 < 0$.

$(-1)^n = +1$, so we have a local maximum if $LPM3 > 0$.

```
(%i78) [q,g];
```

```
(%o78) [K2/5√L, 4 L + 3 K]
```

```
(%i81) Q : q + λ*(108 - g);
      gradQ : jacobian ([Q], [K,L,λ])[1];
      solns : solve ( gradQ, [K, L, λ] );

(Q)   (-4 L - 3 K + 108) λ + K2/5 √L

(gradQ) [  $\frac{2\sqrt{L}}{5K^{3/5}} - 3\lambda, \frac{K^{2/5}}{2\sqrt{L}} - 4\lambda, -4L - 3K + 108$  ]

(solns) [[K=16, L=15, λ= $-\frac{8^{1/5}}{4\sqrt{15}}$ ]]
```

```
(%i82) soln : solns[1];

(soln) [K=16, L=15, λ= $-\frac{8^{1/5}}{4\sqrt{15}}$ ]
```

Let cp be the critical point (K,L) found.

```
(%i83) cp : rest (soln, -1);
(cp)   [K=16, L=15]
```

Both the objective function q and the Lagrangian Q should have the same value at the critical point cp, since the expression multiplying lam is zero at cp.

```
(%i84) subst (cp, [q, Q] ), numer;
(%o84) [11.741, 11.741]
```

Second order conditions (SOC):

```
(%i85) BH : Bhessian (Q, g, [K, L]);

(BH)   
$$\begin{pmatrix} 0 & 3 & 4 \\ 3 & -\frac{6\sqrt{L}}{25K^{8/5}} & \frac{1}{5K^{3/5}\sqrt{L}} \\ 4 & \frac{1}{5K^{3/5}\sqrt{L}} & -\frac{K^{2/5}}{4L^{3/2}} \end{pmatrix}$$

```

Before checking LPM's, evaluate at critical point cp.

```
(%i86) BH : at (BH, cp);
```

$$(BH) \begin{pmatrix} 0 & 3 & 4 \\ 3 & -\frac{6\sqrt{15}}{25 \cdot 16^{8/5}} & \frac{1}{5\sqrt{15} \cdot 16^{3/5}} \\ 4 & \frac{1}{5\sqrt{15} \cdot 16^{3/5}} & -\frac{16^{2/5}}{4 \cdot 15^{3/2}} \end{pmatrix}$$

Let's try out BHtest using BH in the above form.

```
(%i87) BHtest (BH, 1, 2);
```

relative maximum

```
(%o87) [LPM3=4 \left( \frac{3}{5\sqrt{15} \cdot 16^{3/5}} + \frac{24\sqrt{15}}{25 \cdot 16^{8/5}} \right) - 3
```

$$\left(-\frac{3 \cdot 16^{2/5}}{4 \cdot 15^{3/2}} - \frac{4}{5\sqrt{15} \cdot 16^{3/5}} \right)]$$

```
(%i88) float(%);
```

```
(%o88) [LPM3=0.52833]
```

Now we convert BH to floating point numbers and then retry BHtest.

```
(%i89) BH : float (BH);
```

$$(BH) \begin{pmatrix} 0.0 & 3.0 & 4.0 \\ 3.0 & -0.011007 & 0.0097839 \\ 4.0 & 0.0097839 & -0.013045 \end{pmatrix}$$

```
(%i90) BHtest (BH, 1, 2);
```

relative maximum

```
(%o90) [LPM3=0.52833]
```

With $n = 2$, $(-1)^n = +1$, and $\det BH > 0$, so we have met the sufficient conditions for a relative maximum.

3.7 Leydold-Petry Example: Cobb-Douglas

This example is from the pdf: Introduction to Maxima for Economics, Sec. 9.7.

Suppose we have to maximize a Cobb-Douglas production function

$$Y = K \cdot L^2$$

under the constraint

$$g(K, L) = K + L = 3.$$

```
(%i91) [K, L];
```

```
(%o91) [K, L]
```

```
(%i93) Y : K*L^2;
```

```
g : K + L;
```

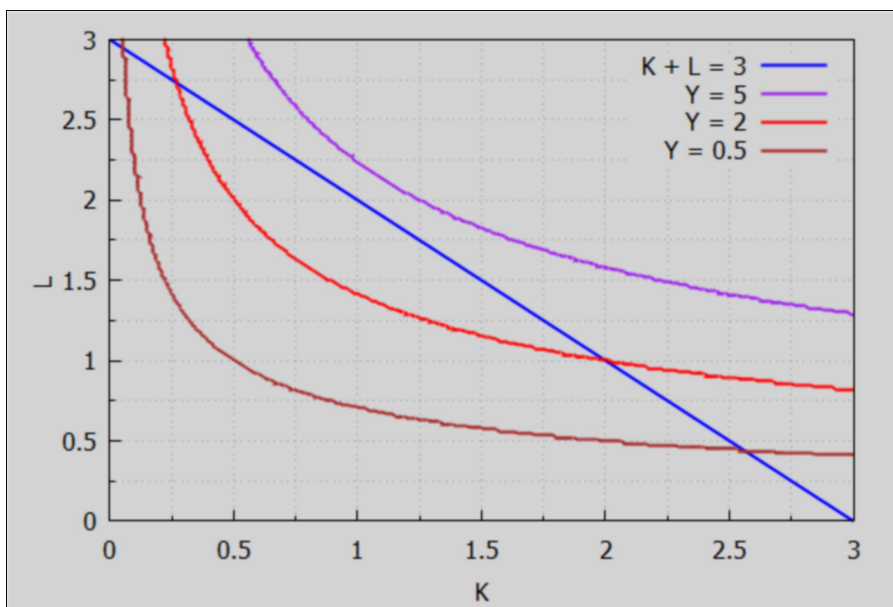
```
(Y) K L^2
```

```
(g) L + K
```

3.7.1 Graphical Exploration

```
(%i94) wxdraw2d ( xlabel = "K", ylabel = "L",
  key = "K + L = 3", explicit (3 - K, K, 0, 3),
  color = purple, key = "Y = 5", implicit ( Y = 5, K, 0, 3, L, 0, 3),
  color = red, key = "Y = 2", implicit ( Y = 2, K, 0, 3, L, 0, 3),
  color = brown, key = "Y = 0.5", implicit ( Y = 0.5, K, 0, 3, L, 0, 3))$
```

```
(%t94)
```



3.7.2 optimum (Y, [K, L], 3 - g)

```
(%i95) optimum (Y, [K, L], 3 - g);
      lagrangian = K L^2 - lam_1 L - lam_1 K + 3 lam_1
      solve returns [[K=1,L=2,lam_1=4],[K=3,L=0,lam_1=0]]
      optimum only evaluates real non-negative solutions
      -----
      i = 1 soln = [K=1,L=2,lam_1=4] objsub = 4
      soln = [K=1.0,L=2.0,lam_1=4.0] objsub = 4.0
      relative maximum
      LPM's = [LPM3=6.0]
      -----
      i = 2 soln = [K=3,L=0,lam_1=0] objsub = 0
      soln = [K=3.0,L=0.0,lam_1=0.0] objsub = 0.0
      relative minimum
      LPM's = [LPM3=-6.0]
(%o95) done
```

optimum has found a maximum at $K = 1$, $L = 2$, with $Y = 4$, which agrees with our graphical exploration, and has also found a local minimum at $K = 3$, $L = 0$, $Y = 0$, but we are only interested in maximizing the production function Y at some $L > 0$ and $Y > 0$ value.

3.7.3 Do it "by hand"...

Let F be the Lagrangian function.

```
(%i98) F : Y + lambda*(3 - g);
      gradF : jacobian ([F], [K,L,lambda])[1];
      solns : solve ( gradF, [K, L, lambda] );
(F)      (-L-K+3) lambda + K L^2
(gradF) [L^2 - lambda, 2 K L - lambda, -L - K + 3]
(solns) [[K=1,L=2,lambda=4],[K=3,L=0,lambda=0]]

(%i99) soln : solns[1];
(soln) [K=1,L=2,lambda=4]

(%i100) at (Y, soln);
(%o100) 4
```

Construct the bordered Hessian matrix using the Lagrange function F .

```
(%i101) BH : Bessian (F, g, [K,L]);
```

```
(BH) 
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 2L \\ 1 & 2L & 2K \end{pmatrix}$$

```

Evaluate the bordered Hessian matrix at the critical point cp of interest.

```
(%i102) BH : at (BH, soln);
```

```
(BH) 
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 4 \\ 1 & 4 & 2 \end{pmatrix}$$

```

```
(%i103) BHtest (BH, 1, 2);
```

relative maximum

```
(%o103) [LPM3=6]
```

With $m = 1$, $n = 2$, $2^*m + 1 = 3$, $m + n = 3$, only one LPM to check. $(-1)^n = +1$, so for a relative maximum we need $LPM3 > 0$.

Leydold and Petry (by error) define $Y : K^2 * L$ in line 1 of their Maxima solution, and get the solution $L = 1$, $K = 2$ after that error.

3.8 Francisco Feri Example, log transformation of objective func.

Francisco Feri, Chair, Dept. of Economics, Royal Holloway, Univ. of London, has posted some pdf's on the web dealing with optimization problems.

[https://intranet.royalholloway.ac.uk/economics/documents/pdf/courseformsandinfo/ec5555lecture6\(2013\)-constrainedoptimisation.pdf](https://intranet.royalholloway.ac.uk/economics/documents/pdf/courseformsandinfo/ec5555lecture6(2013)-constrainedoptimisation.pdf)

Maximize $f = x^3 * y$ subject to the constraint $x + y = 6$.

```
(%i104) [x, y];
```

```
(%o104) [x, y]
```

```
(%i106) f : x^3*y;
```

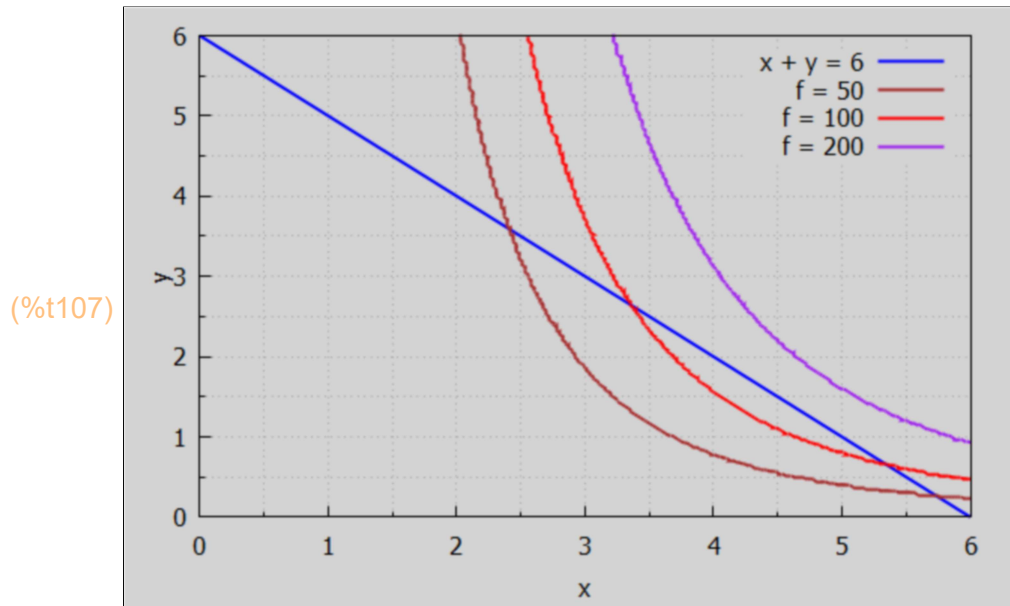
```
g : x + y;
```

```
(f) x^3 y
```

```
(g) y+x
```


3.8.1 Graphical Exploration

```
(%i107) wxdraw2d ( xlabel = "x", ylabel = "y",
  key = "x + y = 6", explicit (6 - x, x, 0, 6),
  color = brown, key = "f = 50", implicit ( f = 50, x, 0, 6, y, 0, 6),
  color = red, key = "f = 100", implicit ( f = 100, x, 0, 6, y, 0, 6),
  color = purple, key = "f = 200", implicit ( f = 200, x, 0, 6, y, 0, 6))$
```



3.8.2 optimum (f, [x,y], 6 - g)

```
(%i108) [f, g];
```

```
(%o108) [x3 y, y+x]
```

(%i109) optimum (f, [x, y], 6 - g);

$$\text{lagrangian} = (x^3 - \text{lam}_1) y - \text{lam}_1 x + 6 \text{lam}_1$$

solve returns $[[x = \frac{9}{2}, y = \frac{3}{2}, \text{lam}_1 = \frac{729}{8}], [x = 0, y = 6, \text{lam}_1 = 0]]$

optimum only evaluates real non-negative solutions

$i = 1$ soln = $[x = \frac{9}{2}, y = \frac{3}{2}, \text{lam}_1 = \frac{729}{8}]$ objsub = $\frac{2187}{16}$

soln = $[x = 4.5, y = 1.5, \text{lam}_1 = 91.125]$ objsub = 136.69

relative maximum

LPM's = $[LPM3 = 81.0]$

$i = 2$ soln = $[x = 0, y = 6, \text{lam}_1 = 0]$ objsub = 0

soln = $[x = 0.0, y = 6.0, \text{lam}_1 = 0.0]$ objsub = 0.0

indefinite

LPM's = $[LPM3 = 0.0]$

(%o109) done

(%i110) cp;

(%o110) $[[x = \frac{9}{2}, y = \frac{3}{2}, \text{lam}_1 = \frac{729}{8}], [x = 0, y = 6, \text{lam}_1 = 0]]$

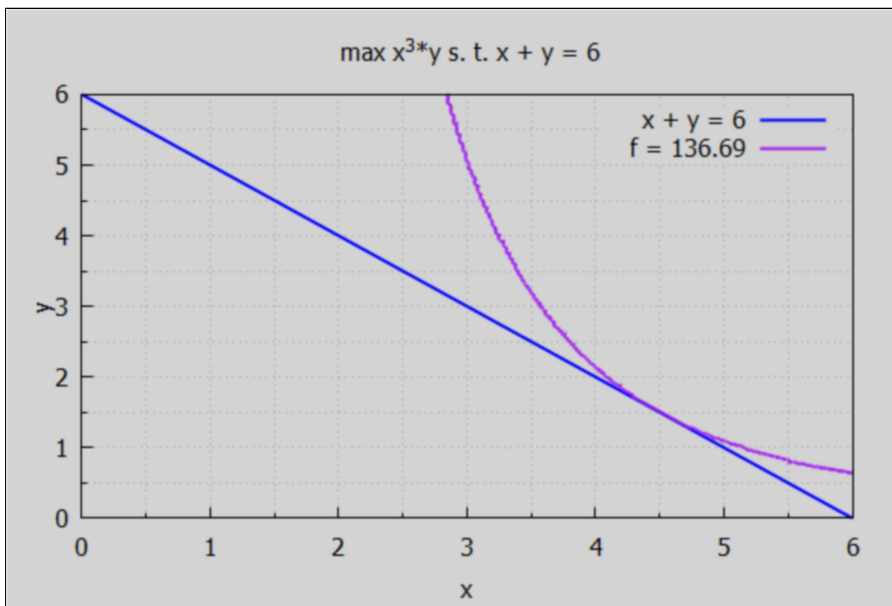
(%i111) cp1 : cp[1], numer;

(cp1) $[x = 4.5, y = 1.5, \text{lam}_1 = 91.125]$

(%i112) fmax : at (f, cp1);

(fmax) 136.69

```
(%i113) wxdraw2d ( xlabel = "x", ylabel = "y",
  title = " max x^3*y s. t. x + y = 6",
  key = "x + y = 6", explicit (6 - x, x, 0, 6),
  color = purple, key = sconcat (" f = ", fmax), implicit ( f = fmax, x, 0, 6, y, 0, 6))$
```



(%t113)

If we instead work with the natural log of $f = x^3y$, and define $f1 = \ln(x^3y) = \ln(y) + 3\ln(x)$,

```
(%i116) logexpand : all$
f1 : log (f);
logexpand : true$
```

```
(f1) log(y) + 3 log(x)
```

```
(%i117) optimum (f1, [x, y], 6 - g);
```

lagrangian = $\log(y) - lam_1 y + 3 \log(x) - lam_1 x + 6 lam_1$

solve returns $[[x = \frac{9}{2}, y = \frac{3}{2}, lam_1 = \frac{2}{3}]]$

optimum only evaluates real non-negative solutions

soln = $[x = \frac{9}{2}, y = \frac{3}{2}, lam_1 = \frac{2}{3}]$ objsub = $3 \log\left(\frac{9}{2}\right) + \log\left(\frac{3}{2}\right)$

soln = $[x = 4.5, y = 1.5, lam_1 = 0.66667]$ objsub = 4.9177

relative maximum

LPM's = $[LPM3 = 0.59259]$

```
(%o117) done
```

We see that the relative maximum of $\log(f)$ is at the same critical point (4.5, 1.5) as the maximum of f .

"BY Hand solution" using f1:

(%i118) [f1, g];

(%o118) [log(y)+3 log(x), y+x]

(%i121) L1 : f1 + λ1*(6 - g);

gradL1 : jacobian ([L1],[x, y, λ1])[1];

solns_In : solve (gradL1, [x, y, λ1]);

(L1) (-y-x+6) λ1 + log(y) + 3 log(x)

(gradL1) [$\frac{3}{x} - \lambda 1, \frac{1}{y} - \lambda 1, -y - x + 6$]

(solns_In) [[$x = \frac{9}{2}, y = \frac{3}{2}, \lambda 1 = \frac{2}{3}$]]

(%i122) soln_In : solns_In[1];

(soln_In) [$x = \frac{9}{2}, y = \frac{3}{2}, \lambda 1 = \frac{2}{3}$]

(%i123) BH : Bhessian (L1, g, [x,y]);

(BH)
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & -\frac{3}{x^2} & 0 \\ 1 & 0 & -\frac{1}{y^2} \end{pmatrix}$$

(%i124) BH : at (BH, soln_In);

(BH)
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & -\frac{4}{27} & 0 \\ 1 & 0 & -\frac{4}{9} \end{pmatrix}$$

(%i125) BHtest (BH, 1, 2);

relative maximum

(%o125) [$LPM3 = \frac{16}{27}$]

3.9 Chiang-Wainwright, p. 360, Two Variable Symbolic Example

Consider a simple two-period model where a consumer's utility is a function of consumption in both periods. Let the consumer's utility function be

$$U(x_1, x_2) = x_1 x_2,$$

where $x_1 > 0$ is consumption in period 1 and $x_2 > 0$ is the consumption in period 2.

The consumer is also endowed with a budget B at the beginning of period 1.

Let $r > 0$ be the market interest rate at which the consumer can choose to borrow or lend across the two periods. The consumer's intertemporal budget constraint is that x_1 and the present value of x_2 add up to B . Thus

$$x_1 + x_2/(1+r) = B.$$

Find the values of x_1 and x_2 which maximize the utility subject to this constraint.

```
(%i126) [x1,x2,r,B,λ];
```

```
(%o126) [x1,x2,r,B,λ]
```

```
(%i130) U : x1*x2;
```

```
g : x1 + x2/(1+r);
```

```
L : U + λ*(B - g);
```

```
gradL : jacobian ([L], [x1, x2, λ])[1];
```

```
(U) x1 x2
```

```
(g) x2
    r+1 + x1
```

```
(L) (- x2
      r+1 - x1 + B) λ + x1 x2
```

```
(gradL) [x2 - λ, x1 - λ
          r+1, - λ
          r+1 - x1 + B]
```

3.9.1 optimum (U, [x1, x2], B - g) Symbolic solution

If we don't give Maxima information about r and B , optimum returns an "indefinite" solution, whereas the solution is actually a relative maximum.

```
(%i131) optimum(U, [x1, x2], B - g);
```

```
lagrangian =
```

$$\frac{((r+1)x_1 - lam_1)x_2 + (-lam_1 r - lam_1)x_1 + lam_1 B r + lam_1 B}{r+1}$$

```
solve returns [[x1 =  $\frac{B}{2}$ , x2 =  $\frac{B r + B}{2}$ , lam1 =  $\frac{B r + B}{2}$ ]]
```

```
optimum only evaluates real non-negative solutions
```

$$soln = [x1 = \frac{B}{2}, x2 = \frac{B r + B}{2}, lam_1 = \frac{B r + B}{2}] \quad objsub = \frac{B(B r + B)}{4}$$

$$soln = [x1 = 0.5 B, x2 = 0.5 (B r + B), lam_1 = 0.5 (B r + B)] \quad objsub = 0.25 B (B r + B)$$

```
indefinite
```

$$LPM's = [LPM3 = \frac{2.0}{r+1.0}]$$

```
(%o131) done
```

But if we give Maxima information that both r and B should be considered positive numbers, optimum (and BHtest) find a relative maximum. Use: `assume(r > 0, B > 0)$`

Note: to remove assumption, use `forget(r > 0, B > 0)$`. To check on status of such assumptions, use `facts()`;

```
(%i135) assume (r > 0, B > 0)$
```

```
optimum(U, [x1, x2], B - g);
```

```
forget (r>0, B >0)$
```

```
facts();
```

```
lagrangian =
```

$$\frac{((r+1)x_1 - lam_1)x_2 + (-lam_1 r - lam_1)x_1 + lam_1 B r + lam_1 B}{r+1}$$

```
solve returns [[x1 =  $\frac{B}{2}$ , x2 =  $\frac{B r + B}{2}$ , lam1 =  $\frac{B r + B}{2}$ ]]
```

```
optimum only evaluates real non-negative solutions
```

$$soln = [x1 = \frac{B}{2}, x2 = \frac{B r + B}{2}, lam_1 = \frac{B r + B}{2}] \quad objsub = \frac{B(B r + B)}{4}$$

$$soln = [x1 = 0.5 B, x2 = 0.5 (B r + B), lam_1 = 0.5 (B r + B)] \quad objsub = 0.25 B (B r + B)$$

```
relative maximum
```

$$LPM's = [LPM3 = \frac{2.0}{r+1.0}]$$

```
(%o133) done
```

```
(%o135) []
```

optimum creates the global list cp.

(%i136) cp;

(%o136) $\left[\left[x1 = \frac{B}{2}, x2 = \frac{B(r+B)}{2}, lam_1 = \frac{B(r+B)}{2} \right] \right]$

(%i137) soln : cp[1], factor;

(soln) $\left[x1 = \frac{B}{2}, x2 = \frac{B(r+1)}{2}, lam_1 = \frac{B(r+1)}{2} \right]$

(%i138) soln : at (soln, lam[1] = λ);

(soln) $\left[x1 = \frac{B}{2}, x2 = \frac{B(r+1)}{2}, \lambda = \frac{B(r+1)}{2} \right]$

(%i139) at (U, soln);

(%o139) $\frac{B^2(r+1)}{4}$

The constraint term doesn't contribute to L at the critical point, since $B - g = 0$ there.

(%i140) at (L, soln);

(%o140) $\frac{B^2(r+1)}{4}$

(%i141) at (B - g, soln);

(%o141) 0

"By hand" SOC:

(%i142) BH : Bhessian (L, g, [x1,x2]);

(BH)
$$\begin{pmatrix} 0 & 1 & \frac{1}{r+1} \\ 1 & 0 & 1 \\ \frac{1}{r+1} & 1 & 0 \end{pmatrix}$$

With one equality constraint, $m = 1$, and 2 variables $(x1, x2)$, $n = 2$, $2*m+1 = 3$, $m+n = 3$ and we only need to check the sign of the determinant of BH, which is the same as LPM3.

(%i143) determinant (BH);

(%o143) $\frac{2}{r+1}$

Since $(-1)^n = (-1)^2 = +1$ and $\det BH > 0$ (with the interest rate $r > 0$), the critical point found is a relative maximum.

(%i146) assume (r > 0)\$

BHtest (BH, 1, 2);

forget (r > 0)\$

relative maximum

(%o145) $[LPM3 = \frac{2}{r+1}]$

3.10 Another Two Variable Symbolic Example

Maximize $f(x,y) = x^a y^b$ subject to $x + y = 10$, where $a > 0$, $b > 0$, and $f(x,y)$ is defined on the set (x,y) with $x \geq 0$, $y \geq 0$.

Step 1. Find all stationary points (critical points) of the Lagrangian.

(%i147) [x,y,a,b,λ];

(%o147) [x,y,a,b,λ]

(%i152) f : x^a*y^b;

g : x + y;

L : f + λ*(10 - g);

gradL : jacobian ([L], [x, y, λ])[1];

soln : solve (gradL, [x,y,λ])[1];

(f) $x^a y^b$

(g) $y+x$

(L) $(-y-x+10) \lambda + x^a y^b$

(gradL) $[a x^{a-1} y^b - \lambda, b x^a y^{b-1} - \lambda, -y-x+10]$

(soln) $[x = \frac{10 a}{b+a}, y = \frac{10 b}{b+a}, \lambda = (b+a) \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^b 10^{b+a-1}]$

(%i153) cp : rest (soln, -1);

(cp) $[x = \frac{10 a}{b+a}, y = \frac{10 b}{b+a}]$

With $a > 0$ and $b > 0$, the solutions $x = x_s$ and $y = y_s$ are positive.

(%i154) fmax : at (f, cp);

$$(fmax) \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^b 10^{b+a}$$

With $a > 0$ and $b > 0$, $fmax > 0$ at the critical point cp.

Next look at the secondary conditions needed for a relative maximum.

(%i155) BH : Bhessian (L, g, [x, y]);

$$(BH) \begin{pmatrix} 0 & 1 & 1 \\ 1 & (a-1) a x^{a-2} y^b & a b x^{a-1} y^{b-1} \\ 1 & a b x^{a-1} y^{b-1} & (b-1) b x^a y^{b-2} \end{pmatrix}$$

Evaluate BH at the critical point found.

(%i156) BHcrit : at (BH, cp);

(BHcrit)

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & (a-1) a \left(\frac{a}{b+a}\right)^{a-2} \left(\frac{b}{b+a}\right)^b 10^{b+a-2} & a b \left(\frac{a}{b+a}\right)^{a-1} \left(\frac{b}{b+a}\right)^{b-1} 10^{b+a-2} \\ 1 & a b \left(\frac{a}{b+a}\right)^{a-1} \left(\frac{b}{b+a}\right)^{b-1} 10^{b+a-2} & (b-1) b \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^{b-2} 10^{b+a-2} \end{pmatrix}$$

With one constraint, $m = 1$, and two variables, $n = 2$, $2^*m+1 = 3$, $n + m = 3$, we only have $LPM3 = \det(BH)$ to check.

(%i157) determinant (BHcrit);

$$(%o157) -(a-1) a \left(\frac{a}{b+a}\right)^{a-2} \left(\frac{b}{b+a}\right)^b 10^{b+a-2} + 2 a b \left(\frac{a}{b+a}\right)^{a-1} \left(\frac{b}{b+a}\right)^{b-1} 10^{b+a-2} - (b-1) b \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^{b-2} 10^{b+a-2}$$

(%i158) expand (%);

$$(%o158) -a^2 \left(\frac{a}{b+a}\right)^{a-2} \left(\frac{b}{b+a}\right)^b 10^{b+a-2} + a \left(\frac{a}{b+a}\right)^{a-2} \left(\frac{b}{b+a}\right)^b 10^{b+a-2} + 2 a b \left(\frac{a}{b+a}\right)^{a-1} \left(\frac{b}{b+a}\right)^{b-1} 10^{b+a-2} - b^2 \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^{b-2} 10^{b+a-2} + b \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^{b-2} 10^{b+a-2}$$

```
(%i159) ratsimp (%);
```

$$\frac{(10^a b^3 + 3 a 10^a b^2 + 3 a^2 10^a b + a^3 10^a) \left(\frac{a}{b+a}\right)^a \left(\frac{b}{b+a}\right)^b 10^{b-2}}{a b}$$

With $a, b > 0$, the determinant of BH (evaluated at the critical point) is positive, and since $(-1)^n = (-1)^2 = 1 > 0$, we have a sufficient condition for a relative maximum.

3.11 Nonlinear 3 Variables Example, 2 Equality Constraints

Maximize $f = x^*y^*z$ subject to both: $x^2 + y^2 = 4$ and $x + z = 2$. The first constraint is nonlinear, which adds a layer of challenge when finding the critical point(s).

```
(%i160) [x, y, z];
```

```
(%o160) [x,y,z]
```

```
(%i163) f : x*y*z;
```

```
g1 : x^2 + y^2;
```

```
g2 : x + z;
```

```
(f) x y z
```

```
(g1) y^2 + x^2
```

```
(g2) z + x
```

This is a $m = 2, n = 3$ problem: $2^*m+1 = 5, m + n = 5$. Only one leading principal minor to check, LPM5.

$(-1)^n = -1$. If LPM5 < 0 , then we have a relative maximum.

$(-1)^m = +1$. If LPM5 > 0 , then we have a relative minimum.

If LPM5 = 0, then we have an indefinite case.

First we optimistically try optimum (func,varL, gL):

```
(%i164) optimum (f, [x, y, z], [4 - g1, 2 - g2]);
```

```
lagrangian = (x y - lam_2) z - lam_1 y^2 - lam_1 x^2 - lam_2 x + 2 lam_2 + 4 lam_1
```

```
solve returns [[x=2,y=0,z=0,lam_1=0,lam_2=0]]
```

```
optimum only evaluates real non-negative solutions
```

```
soln = [x=2,y=0,z=0,lam_1=0,lam_2=0] objsub = 0
```

```
soln = [x=2.0,y=0.0,z=0.0,lam_1=0.0,lam_2=0.0] objsub = 0.0
```

```
indefinite
```

```
LPM's = [LPM5=0.0]
```

```
(%o164) done
```

One solution $x = 2, y = 0, z = 0, f = 0$ was found, which satisfies the constraints, but we suspect the Maxima function solve has difficulties with this problem because of the nonlinearity. Let's revert to a "by hand" mode of using Maxima.

```
(%i165) [f, g1, g2];
```

```
(%o165) [x y z, y^2 + x^2, z + x]
```

```
(%i168) L : f + λ1*(4 - g1) + λ2*(2 - g2);
```

```
gradL : jacobian ([L], [x, y, z, λ1, λ2])[1];
```

```
solns : solve (gradL, [x, y, z, λ1, λ2]);
```

```
(L) (-z - x + 2) λ2 + (-y^2 - x^2 + 4) λ1 + x y z
```

```
(gradL) [-λ2 - 2 x λ1 + y z, x z - 2 y λ1, x y - λ2, -y^2 - x^2 + 4, -z - x + 2]
```

```
(solns) [[x=2, y=0, z=0, λ1=0, λ2=0]]
```

The system of equations to solve are nonlinear, so the Maxima function solve (advertised for solving polynomials) needs help. We can attempt to reduce the number of variables and equations to achieve a solution. Give the 5 first derivatives of L (in the list gradL) the names e1, e2, ..., e5.

```
(%i170) [e1, e2, e3, e4, e5] : gradL$
```

```
matrix (["e1", "e2", "e3", "e4", "e5"],
```

```
[ e1, e2, e3, e4, e5 ]);
```

```
(%o170) (
```

e1	e2	e3	e4	e5
$-\lambda_2 - 2x\lambda_1 + yz$	$xz - 2y\lambda_1$	$xy - \lambda_2$	$-y^2 - x^2 + 4$	$-z - x + 2$

```
)
```

We can simplify by solving e2 and e3 for λ_1 and λ_2

```
(%i172) λ1s : solve (e2, λ1);
```

```
λ2s : solve (e3, λ2);
```

```
(λ1s) [λ1 =  $\frac{xz}{2y}$ ]
```

```
(λ2s) [λ2 = x y]
```

Solve has assumed y is nonzero.

Let's combine these into one list we call λs.

```
(%i173) λs : flatten ( cons ( λ1s, λ2s ));
```

```
(λs) [λ1 =  $\frac{xz}{2y}$ , λ2 = x y]
```

An alternative route would be: $\lambda_s : [\lambda_{1s}[1], \lambda_{2s}[1]]$;

Now replace λ_1 and λ_2 in e_1 , preserving the name e_1 .

```
(%i174) e1 : at ( e1,  $\lambda$ s );
```

```
(e1)  $yz - \frac{x^2 z}{y} - xy$ 
```

The equations to solve are now $e_1 = 0$, $e_4 = 0$, $e_5 = 0$. We can multiply e_1 by y without changing the equation ($e_1 \cdot y = 0$ has the same soln as $e_1 = 0$ if we restrict y to be nonzero). We let e_1 be the name of this product.

```
(%i175) e1 : y*e1, expand;
```

```
(e1)  $y^2 z - x^2 z - xy^2$ 
```

```
(%i176) matrix ( ["e1", "e4", "e5"], [e1, e4, e5] );
```

```
(%o176)  $\begin{pmatrix} e1 & e4 & e5 \\ y^2 z - x^2 z - xy^2 & -y^2 - x^2 + 4 & -z - x + 2 \end{pmatrix}$ 
```

3.11.1 Using eliminate (eqnL, exprL)

Both e_1 and e_4 now have terms with y^2 as a factor. Both e_1 and e_5 contain the factor z .

If we eliminate y^2 and z , using these three equations, we will get an equation in x only. We can use the Maxima function `eliminate ([eqn_1, ..., eqn_n], [x_1, ..., x_k])`. From the Maxima manual:

"Eliminates variables from equations (or expressions assumed equal to zero) by taking successive resultants. This returns a list of $n - k$ expressions with the k variables x_1, \dots, x_k eliminated. First x_1 is eliminated yielding $n - 1$ expressions, then x_2 is eliminated, etc. If $k = n$ then a single expression in a list is returned free of the variables x_1, \dots, x_k . In this case `solve` is called to solve the last resultant for the last variable."

```
(%i177) eliminate ( [e1, e4, e5], [y^2, z] );
```

```
(%o177)  $[-3x^3 + 4x^2 + 8x - 8]$ 
```

```
(%i178) ex : %[1];
```

```
(ex)  $-3x^3 + 4x^2 + 8x - 8$ 
```

`ex` is a cubic equation in x . `solve` usually can deal with a cubic equation.

```
(%i179) xsolns : solve (ex);
```

```
(xsolns) [x = - $\frac{\sqrt{13}+1}{3}$ , x =  $\frac{\sqrt{13}-1}{3}$ , x = 2]
```

Let's assign names to these three solutions for x.

```
(%i180) [x1s, x2s, x3s] : xsolns;
```

```
(%o180) [x = - $\frac{\sqrt{13}+1}{3}$ , x =  $\frac{\sqrt{13}-1}{3}$ , x = 2]
```

Only x2s and x3s imply positive values for x; we reject x1s (which implies a negative x). We Let's first deal with the x3s (x = 2) case, and solve for (y, z) for this case.

Assuming x = 2, the equation e5 = 0 (- z - x + 2 = 0), implies z = 0.

Then equationn e4 = 0 (- y^2 - x^2 + 4 = 0), implies y^2 = 0 which implies y = 0, consistent with eqn e1 = 0, so (x = 2, y = 0, z = 0) and f = 0, a solution we reject.

Repeating this argument using Maxima:

```
(%i181) at (e5, x3s);
```

```
(%o181) -z
```

Then e5 = 0 implies z = 0, hence f = 0.

```
(%i182) at (e4, x3s );
```

```
(%o182) -y^2
```

Then e4 = 0 implies y = 0. We reject the x = 2 soln x3s.

Next look at the case x2s and work with floating point numbers.

```
(%i183) x2s : float (x2s);
```

```
(x2s) x = 0.86852
```

```
(%i184) numer : true$
```

```
(%i185) at (e5, x2s);
```

```
(%o185) 1.1315 - z
```

```
(%i186) solve(% , z);
```

```
(%o186) [z = 1.1315]
```

```
(%i187) z2s : %[1];
```

```
(z2s) z = 1.1315
```

```
(%i188) e4;
```

```
(%o188)  $-y^2 - x^2 + 4$ 
```

```
(%i189) at (e4, x2s);
```

```
(%o189)  $3.2457 - y^2$ 
```

```
(%i190) solve (%, y^2);
```

```
(%o190)  $[y^2 = 3.2457]$ 
```

```
(%i191) fac : sqrt(rhs (%[1]));
```

```
(fac) 1.8016
```

So $y = \pm 1.8016$, and we choose the mathematical solution with the positive value of y to correspond to a real world Economics solution.

```
(%i192) y2s : y = fac;
```

```
(y2s) y = 1.8016
```

```
(%i193) cp2 : [x2s, y2s, z2s];
```

```
(cp2)  $[x = 0.86852, y = 1.8016, z = 1.1315]$ 
```

```
(%i194) at (f, cp2);
```

```
(%o194) 1.7704
```

Thus by helping out solve (using eliminate) we have uncovered a solution (critical point) for which all the variables are positive. Next, find the values of the Lagrange multipliers.

```
(%i195) λs;
```

```
(%o195)  $[\lambda_1 = \frac{xz}{2y}, \lambda_2 = xy]$ 
```

```
(%i196) λs : at (λs, cp2);
```

```
(λs)  $[\lambda_1 = 0.27274, \lambda_2 = 1.5647]$ 
```

```
(%i197) soln : flatten (cons (cp2, λs));
```

```
(soln)  $[x = 0.86852, y = 1.8016, z = 1.1315, \lambda_1 = 0.27274, \lambda_2 = 1.5647]$ 
```

```
(%i198) [L, x, y, z, g1, g2];
```

```
(%o198) [(-z-x+2) λ2 + (-y2-x2+4) λ1 + x y z, x, y, z, y2+x2, z+x]
```

Set up the appropriate bordered Hessian matrix:

```
(%i199) BH : Bhessian (L, [g1, g2], [x, y, z ] );
```

```
(BH) 
$$\begin{pmatrix} 0 & 0 & 2x & 2y & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 2x & 1 & -2\lambda_1 & z & y \\ 2y & 0 & z & -2\lambda_1 & x \\ 0 & 1 & y & x & 0 \end{pmatrix}$$

```

Use the replacement list soln to evaluate BH at the critical point found:

```
(%i200) BH : at (BH, soln);
```

```
(BH) 
$$\begin{pmatrix} 0 & 0 & 1.737 & 3.6032 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1.737 & 1 & -0.54547 & 1.1315 & 1.8016 \\ 3.6032 & 0 & 1.1315 & -0.54547 & 0.86852 \\ 0 & 1 & 1.8016 & 0.86852 & 0 \end{pmatrix}$$

```

In the lower right block we have the 3x3 submatrix of the second order partial derivatives of the Lagrangian function wrt (x,y,z), bordered by the first order partial derivatives of g1 and g2, all evaluated at the critical point.

This is an $m = 2$, $n = 3$ problem, use BHtest (BH, m, n).

```
(%i201) BHtest (BH, 2, 3);
```

```
relative maximum
```

```
(%o201) [LPM5 = -58.798]
```

With $m = 2$, $n = 3$, $2m + 1 = 5$, $m + n = 5$, there is only one LPM to check, $LPM5 = \det(BH)$. $(-1)^n = (-1)^3 = -1$, and $LPM5 < 0$ so a relative maximum.

```
(%i202) determinant (BH);
```

```
(%o202) -58.798
```

```
(%i203) numer : false$
```

4 **Input-Output Analysis [12.6]**

We roughly quote from the beginning of Sec. 5.7 of Chiang & Wainwright:

In its "static version", the input-output analysis of Professor Wassily Leontief, a Nobel Prize winner, deals with the question: "What level of output should each of the n industries in an economy produce, in order that it will just be sufficient to satisfy the total demand for that product?"

In a modern economy where the production of one good requires the input of many other goods as "intermediate goods" in the production process (steel requires coal, iron ore, electricity, etc.), *total demand* x for product i ($x[i]$) will be the summation of all intermediate demand for the *production* of the product plus the "final demand" b for the product arising from consumers, investors, the government, and exporters, as ultimate users of the product.

To simplify the problem, the assumptions made are

1. each industry produces only one homogeneous commodity,
2. each industry uses a fixed "input ratio" for the production of its output,
3. production in each industry is subject to "constant returns to scale", so that a k -fold change in every input will result in an exactly k -fold change in the output

From these assumptions, in order to produce each unit of the j th commodity, the need for the i th commodity must be a fixed amount, which we denote by $a[i,j]$. Specifically, the production of each unit of the j th commodity will require $a[1,j]$ (amount) of the first commodity, $a[2,j]$ amount of the second commodity, ..., and $a[n,j]$ of the n th commodity. The order of the indices is easy to remember: the first index refers to the input, the second to the output, so $a[i,j]$ indicates how much of the i th commodity is used for the production of each unit of the j th commodity.

If we assume prices are a given, we can use "a dollar's worth" of each commodity as its unit. Then the statement $a[3,2] = 0.35$ means that 35 cents worth of the third commodity is required as an input for producing one dollar's worth of the second commodity.

For an n -industry economy, the input coefficients can be arranged into an $n \times n$ matrix A , in which each column specifies the input requirements for the production of one unit (one dollar's worth) of the output requirements for a particular industry.

Here is an example for a four industry economy:

(%i204) A : genmatrix (a, 4, 4);

$$(A) \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix}$$

For this 4 x 4 matrix A, the second column states that to produce a unit (one dollar's worth) of commodity 2, the inputs are: $a_{1,2}$ units of commodity 1, $a_{2,2}$ units of commodity 2, etc. If no industry uses its own products as an input, then the elements in the principal diagonal of matrix A will be zero.

4.1 The Open Model

If the n industries constitute the entirety of the economy, then all their products would be for the sole purpose of meeting the "input demand" of the same n industries (to be used for further production) as against the "final demand" (such as consumer demand, not to be used for further production).

At the same time, all the inputs used in the economy would be in the nature of "intermediate inputs" (those supplied by the n industries) as against "primary inputs" (such as labor and capital, not industrial products). To allow for the presence of final demand (consumer demand) and primary inputs to production (labor, capital, etc), we must include in the model an "open sector" outside of the n -industry network. Such an open sector can accommodate the activities of the consumer households, the government sector, and even foreign countries.

In view of the presence of the open sector, the sum of the elements of each column of the input matrix A must be less than 1. Each column sum represents the "partial input cost" (not including the cost of the primary inputs (labor, capital,...)).

Symbolically we need (using Maxima notation): $\sum (a_{i,j}, i, 1, n) < 1$, for $j = 1, 2, \dots, n$.

If $a[i,j]$ is a "technical coefficient" expressing the value of input from industry i required to produce one dollars worth of product by industry j , the *total demand* for product from industry i is $x[i]$, which can be expressed as the sum of $b[i]$ (final demand by consumers) plus the intermediate demands of the n -industry network.

$$x[i] = a[i,1]*x[1] + a[i,2]*x[2] + \dots + a[i,n]*x[n] + b[i],$$

$b[i]$ is the "final demand" (by consumers, etc) for the product of industry i .

If we move all terms proportional to $x[k]$ to the left of the equal sign, leaving the exogenous variables $b[i]$ on the right hand side, we can write this set of n equations in matrix form:

$$(I - A) \cdot X = B, \text{ or}$$

$$M \cdot X = B$$

where $M = (I - A)$, with I standing for the unit matrix, A is the matrix of technical coefficients, X is a matrix column vector of unknown total demands $x[k]$, and B is a matrix column vector of given final demands $b[k]$.

The matrix $M = (I - A)$ is called the Leontief matrix. As long as the Leontief matrix is nonsingular, we can find its inverse $M^{(-1)} = \text{invert}(M)$, and obtain the unique solution of the n equations using $Xs : \text{invert}(M) \cdot B$

For our $n = 4$ example we first construct lists xL and bL and turn each into a matrix column vector using $X : \text{cvec}(xL)$, for example, where cvec is defined in `Econ1.mac`. `ident(4)` returns a 4×4 unit matrix (1's on the diagonal and 0's elsewhere).

```
(%i213) A : genmatrix (a, 4, 4);
      xL : []$
      bL : []$
      for k thru 4 do (
        xL : cons (x[k], xL),
        bL : cons (b[k], bL))$
      xL : reverse (xL);
      bL : reverse (bL);
      X : cvec (xL);
      B : cvec (bL);
      M : ident(4) - A;
```

$$(A) \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix}$$

$$(xL) [x_1, x_2, x_3, x_4]$$

$$(bL) [b_1, b_2, b_3, b_4]$$

$$(X) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$(B) \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

$$(M) \begin{pmatrix} 1-a_{1,1} & -a_{1,2} & -a_{1,3} & -a_{1,4} \\ -a_{2,1} & 1-a_{2,2} & -a_{2,3} & -a_{2,4} \\ -a_{3,1} & -a_{3,2} & 1-a_{3,3} & -a_{3,4} \\ -a_{4,1} & -a_{4,2} & -a_{4,3} & 1-a_{4,4} \end{pmatrix}$$

```
(%i214) meqns : M . X = B;
```

$$(meqns) \begin{pmatrix} -a_{1,4} x_4 - a_{1,3} x_3 - a_{1,2} x_2 + x_1 (1 - a_{1,1}) \\ -a_{2,4} x_4 - a_{2,3} x_3 + x_2 (1 - a_{2,2}) - x_1 a_{2,1} \\ -a_{3,4} x_4 + x_3 (1 - a_{3,3}) - x_2 a_{3,2} - x_1 a_{3,1} \\ x_4 (1 - a_{4,4}) - x_3 a_{4,3} - x_2 a_{4,2} - x_1 a_{4,1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

The inverse of a matrix M is returned by `invert (M)`, and we know that
 $\text{invert}(M) \cdot M = I$
 returns a unit matrix (don't try to verify this using the above definition of M).

Hence the solution matrix column vector Xs is given by

$Xs : \text{invert}(M) \cdot B$
 (again, don't try to do this symbolically now). See the numerical example below.

4.2 A Numerical Example

Chiang and Wainwright, p. 115, have the following numerical example for a three industry economy. A is the matrix of technical coefficients. Element (i,j) of A can be selected in Maxima using the syntax $A[i,j]$.

```
(%i215) A : matrix ( [0.2, 0.3, 0.2], [0.4, 0.1, 0.2], [0.1, 0.3, 0.2] );
```

$$(A) \begin{pmatrix} 0.2 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.2 \end{pmatrix}$$

```
(%i216) A[1,2];
```

```
(%o216) 0.3
```

The value of $A[1,2] = 0.3$ means that \$0.30 worth of product 1 is required as input for producing \$1 worth of product 2. If we add up the elements of column 2 of the matrix A we get

```
(%i217) sum (A[k, 2], k, 1, 3);
```

```
(%o217) 0.7
```

which is less than 1 as it should be (If it required \$2 worth of input to produce \$1 worth of product 2, production would promptly cease, in a rational world). With this given data, we assume the difference between 0.7 and 1 means that "primary input" such as the cost of labor and capital amounts to \$0.30 dollar's worth for every one dollar's worth of product 2.

To produce one billions dollar's worth of product 2 requires 0.3 billion dollar's worth of product 1.

4.2.1 Leontief(A)

The software file `Econ1.mac` defines a Maxima function `Leontief (A)` which returns the Leontief matrix $(I - A)$.

(%i218) Leontief (A);

$$(\%o218) \begin{pmatrix} 0.8 & -0.3 & -0.2 \\ -0.4 & 0.9 & -0.2 \\ -0.1 & -0.3 & 0.8 \end{pmatrix}$$

(%i219) length (A);

(%o219) 3

Here we are creating the Leontief matrix $M = (I - A)$ "by hand".

(%i220) M : ident (3) - A;

$$(M) \begin{pmatrix} 0.8 & -0.3 & -0.2 \\ -0.4 & 0.9 & -0.2 \\ -0.1 & -0.3 & 0.8 \end{pmatrix}$$

(%i221) length (M);

(%o221) 3

(%i222) rank (M);

(%o222) 3

The rank (see the first section of Dowling12A.wmxm) of M is 3, which is equal to the dimension of M (given by $\text{length}(M)$), so M is nonsingular, and the matrix inverse of M exists.

(%i223) Minv : invert (M);

$$(Minv) \begin{pmatrix} 1.7188 & 0.78125 & 0.625 \\ 0.88542 & 1.6146 & 0.625 \\ 0.54687 & 0.70312 & 1.5625 \end{pmatrix}$$

If the specific (consumer) final-demand column vector (say, the final-output target of a development program) happens to be (in billions of dollars)

(%i224) B : cvec ([10, 5, 6]);

$$(B) \begin{pmatrix} 10 \\ 5 \\ 6 \end{pmatrix}$$

```
(%i225) length (B);
(%o225) 3
```

4.2.2 InputOutput (M, B)

The software file Econ1.mac defines the Maxima function

InputOutput (M, B)

in which $M = I - A$ is the Leontif matrix and B is the "final demand".

InputOutput (M, B) checks some properties of M and returns the total demand matrix column vector solution.

```
(%i226) InputOutput (M, B);
```

```
(%o226)  $\begin{pmatrix} 24.844 \\ 20.677 \\ 18.359 \end{pmatrix}$ 
```

Here we work this problem "by hand": the total demand solution vector X_s (again in billions of dollars) is

```
(%i227) Xs : Minv . B;
```

```
(Xs)  $\begin{pmatrix} 24.844 \\ 20.677 \\ 18.359 \end{pmatrix}$ 
```

```
(%i230) Xs[1];
Xs[1,1];
Xs[2,1];
```

```
(%o228) [24.844]
```

```
(%o229) 24.844
```

```
(%o230) 20.677
```

We see that we can use syntax $X_s[1,1]$ to get the first element of the total demand column vector matrix, $X_s[2,1]$ to get the second element, etc. Alternatively, we can use $X_s[1][1]$, $X_s[2][1]$, etc.

Let $a_0(j)$ be the dollar amount of the primary input (like labor, capital, etc.) used in producing one dollar's worth of the j th commodity. $a_0(j)$ can be calculated by subtracting the sum of the j th column of the input matrix A from 1.

In general, $A[k, j]$ returns the k th row and j th column element of the matrix A.

Here we define a Maxima function depending of j :

```
(%i231) a0(j) := 1 - sum (A[k, j], k, 1, 3)$
```

then we have

```
(%i232) [a0(1), a0(2), a0(3)];
```

```
(%o232) [0.3, 0.3, 0.4]
```

4.2.3 Pinput (A, j)

The Maxima function Pinput (A, j) (primary input j) is defined in Econ1.mac, and does the same job as the Maxima function a0(j) if A is the given matrix of technical coefficients, plus figures out what the final k value should be in the sum (using the dimension of A).

```
(%i233) Pinput (A, 1);
```

```
(%o233) 0.3
```

The Maxima lambda expression can be used to map Pinput onto a list of numbers for a chosen value of A. See the Maxima manual index under lambda.

```
(%i234) map (lambda ([xx], Pinput (A, xx)), [1, 2, 3] );
```

```
(%o234) [0.3, 0.3, 0.4]
```

An important question now arises. The production of the total demand output mix $Xs[k]$ (for $k = 1$ thru 3) must entail a definite required amount of the primary input (labor, capital, etc). Would the amount *required* be consistent with what is *available* in the economy?

On the basis of the values we found for $a0(j)$ (with $j = 1, 2, 3$), the required primary input (PI) (labor, capital, etc) is the sum of the products $a0(k) * Xs[k]$ with $k = 1, 2, 3$.

As an example, for product 1

```
(%i237) Xs[1];
```

```
  a0(1);
```

```
  a0(1) * Xs[1,1];
```

```
(%o235) [24.844]
```

```
(%o236) 0.3
```

```
(%o237) 7.4531
```

The required primary input (PI) is:

```
(%i238) PI : sum (a0(k)*Xs[k,1], k, 1, 3);
```

```
(PI) 21.0
```

4.2.4 Pltot (A, Xs)

The Maxima function Pltot (A, Xs) calculates this required primary input as a function of the matrix of technical coefficients A and the solution vector Xs.

```
(%i239) Pltot (A, Xs);
```

```
(%o239) 21.0
```

so the specific demand (in billions) assumed in our definition of the consumer final demand matrix column vector B will be feasible if and only if the available amount of the primary input (labor, capital,...) is at least \$21 billion.

One notable feature of the previous analysis is that, as long as the input coefficients (technical coefficients) $A[i,j] = a[i,j]$ remain the same numbers, the matrix inverse $Minv$ remains the same, even if we consider a hundred or a thousand different final-demand vectors - such as a spectrum of alternative development targets.

4.3 Problem 12.33 & 12.34

Given the interindustry transaction demand table in millions of dollars below, find the matrix of technical coefficients A. The "Sector of Origin" categories refer to inputs provided to "output" industries (Sector of Destination). Sector of origin category "primary input" refers to input of labor, capital, etc, in millions of dollars, required to produce the gross product by each industry (column). Dowling refers to primary input (in his table) as "Value Added".


```
(%i241) print ("          Sector of Destination ")$
Mtable : matrix (["Sector of Origin", "steel", "coal", "iron", "Auto", "Final-Demand", "Total-demand"],
                ["steel",      80,   20,  110,  230,   160,    600],
                ["coal",      200,   50,   90,  120,   140,    600],
                ["iron",      220,  110,   30,   40,    0,    400],
                ["Auto",      60,  140,  160,  240,   400,  1000],
                ["primary-input",40,  280,   10,  370,   "",    "" ],
                ["Gross Prod.", 600,  600,  400, 1000,   "",    "" ]);
```

(Mtable)

Sector of Origin	steel	coal	iron	Auto	Final-Demand	Total-demand
steel	80	20	110	230	160	600
coal	200	50	90	120	140	600
iron	220	110	30	40	0	400
Auto	60	140	160	240	400	1000
primary-input	40	280	10	370		
Gross Prod.	600	600	400	1000		

To make this table clearer, consider the column Auto production under the category Sector of Destination. Let M stand for million.

To produce \$1,000M worth of autos, the Auto producer needs as inputs, \$230M worth of steel, \$120M worth of coal, \$40M worth of iron, \$240M worth of its own autos, and \$370M worth of "primary inputs" (labor, capital, etc).

If we look at the "Sector of Origin" row Auto, which has the amounts of the Auto industry inputs to various producers, Auto supplies \$60M worth of autos to the steel producer, \$140M worth of autos to the coal producer, \$160M worth of autos to the iron producer, \$240M worth of autos to itself, adding up to \$600M worth of autos. If we then add \$400M worth of autos to consumers (final-demand), we get a total demand of \$1,000M worth of autos (one billion dollars worth).

Note that "gross production" of autos is equal to "total-demand" for autos, etc.

In the technical coefficient matrix A, element A[1,1] (row 1, column 1) is the amount of steel needed as input to produce one dollars worth of steel. From our table we see that it takes \$80M worth of steel input to produce \$600M worth of steel output. Since $80/600 = 0.1333$, it takes \$0.1333 worth of steel input to produce \$1 worth of steel output, so $A[1,1] = 0.1333$. Likewise, it takes \$20M worth of steel to produce \$600M worth of coal, so $A[1,2] = 20/600 = 0.033$.

```
(%i242) A : genmatrix (a, 4, 4);
```

$$(A) \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix}$$

At this point $a[2,3]$ (for example) is an undefined hash array element, and we proceed to assign values to these elements based on the matrix Mtable above.

```
(%i243) a[2,3];
```

```
(%o243) a2,3
```

```
(%i244) f(i,j) := float (Mtable[i+1, j+1] / Mtable[7,j+1])$
```

```
(%i245) f(1,1);
```

```
(%o245) 0.13333
```

```
(%i246) for i thru 4 do
         for j thru 4 do a[i,j] : f(i,j)$
```

```
(%i247) a[1,1];
```

```
(%o247) 0.13333
```

To bind the symbol A to the numerical value matrix we want, we have to force Maxima to evaluate A using two (2) single quotes (') in front.

```
(%i248) A : "A;
```

$$(A) \begin{pmatrix} 0.13333 & 0.033333 & 0.275 & 0.23 \\ 0.33333 & 0.083333 & 0.225 & 0.12 \\ 0.36667 & 0.18333 & 0.075 & 0.04 \\ 0.1 & 0.23333 & 0.4 & 0.24 \end{pmatrix}$$

The amounts of primary inputs $a_0(j)$ required to produce one dollar's worth of the product of industry j are defined by the Maxima function (for this four industry economy):

```
(%i249) a0(j) := 1 - sum (A[k, j], k, 1, 4)$
```

```
(%i250) matrix ( ["steel", "coal", "iron", "Auto"],
                [a0(1), a0(2), a0(3), a0(4)] );
```

```
(%o250)  $\begin{pmatrix} steel & coal & iron & Auto \\ 0.066667 & 0.46667 & 0.025 & 0.37 \end{pmatrix}$ 
```

If we multiply these numbers by the gross production (in millions of dollars) for the respective industry, we get the primary inputs (labor, capital, etc) required for each of the four industries.

```
(%i251) matrix ( ["steel", "coal", "iron", "Auto"],
                [a0(1)*600, a0(2)*600, a0(3)*400, a0(4)*1000]);
```

```
(%o251)  $\begin{pmatrix} steel & coal & iron & Auto \\ 40.0 & 280.0 & 10.0 & 370.0 \end{pmatrix}$ 
```

To perform a consistency check on these values of the elements of the matrix A, we need to define the column vector X whose four elements are the total demands, and the column vector B whose four elements are the final demands.

```
(%i253) X : cvec ([600, 600, 400, 1000] );
         B : cvec ([160, 140, 0, 400]);
```

```
(X)  $\begin{pmatrix} 600 \\ 600 \\ 400 \\ 1000 \end{pmatrix}$ 
```

```
(B)  $\begin{pmatrix} 160 \\ 140 \\ 0 \\ 400 \end{pmatrix}$ 
```

We compare $A \cdot X$ and $X - B$, which should be the same.

```
(%i254) A . X;
```

```
(%o254)  $\begin{pmatrix} 440.0 \\ 460.0 \\ 400.0 \\ 600.0 \end{pmatrix}$ 
```

(%i255) X - B;

(%o255)
$$\begin{pmatrix} 440 \\ 460 \\ 400 \\ 600 \end{pmatrix}$$

Since $A \cdot X = X - B$, this is consistent with our original definition of the total demand vector X , a solution of the equation $(I - A) \cdot X = B$, equivalent to: $X - A \cdot X = B$.

We can also, of course, invert the Leontief matrix $M = (I - A)$ and the matrix product $Minv \cdot B$ should reproduce the total demand vector X .

(%i256) M : ident (4) - A;

(M)
$$\begin{pmatrix} 0.86667 & -0.033333 & -0.275 & -0.23 \\ -0.33333 & 0.91667 & -0.225 & -0.12 \\ -0.36667 & -0.18333 & 0.925 & -0.04 \\ -0.1 & -0.23333 & -0.4 & 0.76 \end{pmatrix}$$

(%i257) rank (M);

(%o257) 4

Since the rank is the same as the dimension, M is nonsingular, and the inverse matrix exists.

(%i258) Minv : invert (M);

(Minv)
$$\begin{pmatrix} 1.7701 & 0.41066 & 0.90645 & 0.64823 \\ 1.0127 & 1.4582 & 0.90855 & 0.58454 \\ 0.94745 & 0.48453 & 1.6758 & 0.45143 \\ 1.0425 & 0.75675 & 1.2802 & 1.8181 \end{pmatrix}$$

$$\begin{array}{l}
 (\%i260) \mathbf{X}; \\
 \text{Minv} \cdot \mathbf{B}; \\
 (\%o259) \begin{pmatrix} 600 \\ 600 \\ 400 \\ 1000 \end{pmatrix} \\
 (\%o260) \begin{pmatrix} 600.0 \\ 600.0 \\ 400.0 \\ 1000.0 \end{pmatrix}
 \end{array}$$

which again shows consistency.

4.3.1 The Existence of Nonnegative Solutions

We roughly quote the Chiang and Wainwright, p. 116 - 117, section dealing with this issue.

In our previous examples, the Leontief matrix $M = (I - A)$ happens to be nonsingular, so solution values $Xs[j]$ exist, and in our examples turned out to be nonnegative, as economic sense requires. Nonnegative values for total demand are not automatic, but depend on the Leontief matrix having specific properties, the "Hawkins-Simon condition", made clear in a 1949 academic article by David Hawkins and Herbert A. Simon.

The theorem proved by Hawkins and Simon states:

a) Given an $n \times n$ matrix M , with $M[i,j] \leq 0$ ($i \neq j$) (ie., with all off-diagonal elements non-positive), and b) a matrix column vector B with all n elements nonnegative, there exists a n dimensional solution matrix column vector Xs , with all elements nonnegative, if and only if all of the leading principal minors of M are positive numbers.

The Leontief matrix $M = (I - A)$ satisfies condition (a), and the final demand matrix column vector B satisfies condition (b). Hence if M is nonsingular, and if all of the leading principal minors of M are positive numbers, we are guaranteed that a unique solution for total demand Xs exists, and that all elements $Xs[j]$ are nonnegative.

4.3.2 Economic Meaning of the Hawkins-Simon Condition

We continue to follow Chiang and Wainwright (pp. 118 - 119)

Consider the two industry case. We assume $a[i,j] \geq 0$ for all $(i, j) = 1, 2$.

We want to redefine A as a 2 x 2 matrix with symbolic elements $a_{[i,j]}$.
 We use `remarray (a)` to remove our previous numerical binding of $a_{[i,j]}$ elements.
 First, we remind the reader of some commonly used array functions.

```
(%i261) a[1,4];
```

```
(%o261) 0.23
```

```
(%i262) arrays;
```

```
(%o262) [a]
```

```
(%i263) arrayinfo (a);
```

```
(%o263) [hashed,2,[1,1],[1,2],[1,3],[1,4],[2,1],[2,2],[2,3],[2,4],
        [3,1],[3,2],[3,3],[3,4],[4,1],[4,2],[4,3],[4,4]]
```

```
(%i264) listarray(a);
```

```
(%o264) [0.13333,0.033333,0.275,0.23,0.33333,0.083333,0.225,0.12,
        0.36667,0.18333,0.075,0.04,0.1,0.23333,0.4,0.24]
```

```
(%i266) remarray (a);
```

```
arrays;
```

```
(%o265) [a]
```

```
(%o266) []
```

```
(%i269) A : genmatrix (a, 2,2);
```

```
I : ident (2);
```

```
M : I - A;
```

```
(A) 
$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

```

```
(I) 
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```

```
(M) 
$$\begin{pmatrix} 1-a_{1,1} & -a_{1,2} \\ -a_{2,1} & 1-a_{2,2} \end{pmatrix}$$

```

We use our function `LPM (M, k)` to compute the leading principal minor of M for $k = 1, 2$.
 The function `LPM` is defined in the file `Econ1.mac`.

What does the condition `LPM (M,1) > 0` require?

```
(%i270) lpm1 : LPM (M,1);
```

```
(lpm1) 1-a1,1
```

$$(\%i271) \text{lpm1} > 0;$$

$$(\%o271) 1 - a_{1,1} > 0$$

This first condition requires that $1 > a[1,1]$, or $a[1,1] < 1$, or that the amount of the first commodity used in the production one dollar's worth of the first commodity be less than one dollar (and can be zero).

What does the condition $\text{LPM}(M, 2) > 0$ require?

$$(\%i272) \text{lpm2} : \text{LPM}(M, 2);$$

$$(\text{lpm2}) (1 - a_{1,1})(1 - a_{2,2}) - a_{1,2}a_{2,1}$$

$$(\%i273) \text{lpm2} : \text{expand}(\text{lpm2});$$

$$(\text{lpm2}) a_{1,1}a_{2,2} - a_{2,2} - a_{1,2}a_{2,1} - a_{1,1} + 1$$

$$(\%i274) \text{lpm2} > 0;$$

$$(\%o274) a_{1,1}a_{2,2} - a_{2,2} - a_{1,2}a_{2,1} - a_{1,1} + 1 > 0$$

This second condition requires that

$$a[1,1] + a[1,2]a[2,1] + a[2,2](1 - a[1,1]) < 1.$$

Since the 3rd term on the left hand side is nonnegative, and could be zero, we must have:

$$a[1,1] + a[1,2]a[2,1] < 1.$$

In terms of economics, $a[1,1]$ measures the *direct* use of commodity 1 as input in the production of commodity 1, and the product $a[1,2]a[2,1]$ measures the *indirect* use - it gives the amount of commod. 1 needed in producing the specific quantity of commod. 2 that is needed in the production of one dollar's worth of commod. 1.

Thus we require that the amounts of commod. 1 used as both direct and indirect inputs in producing one dollar's worth of commod. 1, must add up to less than one dollar.

Thus the Hawkins-Simon condition requires practical restrictions on the production process details that result in a meaningful and viable process, in the sense of Economics.

4.4 Summary of Input/Output Analysis Functions

4.4.1 InputOutput (M, B)

Note that if the determinant of a matrix is not equal to zero, the matrix is said to be "nonsingular", and all its rows and columns are linearly independent, and the rank of the matrix equals its dimension.

But the determinant of an n dimensional matrix ($n \times n$) is the same as the n th leading principal minor. If $LPM(M, n) > 0$, we already satisfy the condition that the determinant of M is not equal to zero, so M is nonsingular, and an inverse of M exists, and a unique solution of the matrix equation $M \cdot X = B$ exists.

The Maxima function `InputOutput(M, B)` simply computes the n leading principal minors of the square matrix M , and if they are all positive numbers, returns the unique solution matrix column vector with all nonnegative elements, otherwise returns "not viable".

The off-diagonal elements of M should be nonpositive numbers.

4.4.2 Leontief (A)

`Leontief(A)` returns the Leontief matrix $(I - A)$ and checks that each element of A is positive.

4.4.3 Pinput (A, j)

`Pinput(A, j)` subtracts the sum of the j th column elements of the matrix A from 1, returning the primary input value (labor, capital, etc) required for the j th industry. A is the matrix of technical coefficients.

4.4.4 Pltot (A, Xs)

`Pltot(A, Xs)` returns the total primary input (labor, capital, etc) required, given the matrix of technical coefficients A and the total demands solution vector Xs of the equation $(I - A) \cdot Xs = B$, in which B is the matrix column vector of final demands.

5 *Eigenvalue Tests for Sign Definiteness [12.7]*

To this point, the sign definiteness of a Hessian and a quadratic form have been tested by using the leading principal minors. Sign definiteness can also be tested by using the characteristic roots of a matrix. Given a square matrix A , if it is possible to find a vector $V \neq 0$ and a scalar c such that $A \cdot V = c \cdot V$.

The scalar c is called the characteristic root, latent root, or eigenvalue of the matrix A , and the vector V is called the characteristic vector, or eigenvector of the matrix A .

To find the eigenvalues and eigenvectors of a given matrix A , we note that, using the identity matrix I , with the property: $I \cdot V = V$, $A \cdot V = c \cdot V$ is equivalent to:

$$A \cdot V = c \cdot (I \cdot V)$$

which can be rearranged as

$$(A - c \cdot I) \cdot V = 0,$$

where $(A - c \cdot I)$ is called the "characteristic matrix of A ".

When using the shorthand way of writing this matrix equation: $(A - c \cdot I) \cdot V = 0$, we are glossing over the notational convenience of writing the matrix on the right hand side of the equation as 0. Taken literally, this is mathematical nonsense. What is meant (by the context) is a column matrix of zeros, such as

(%i275) `zeromatrix (4, 1);`

(%o275)
$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

if A is a 4 x 4 square matrix. We can also use

(%i276) `cvec ([0, 0, 0, 0]);`

(%o276)
$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

With this convenience convention understood, we continue to use the usual matrix equation notation $M = 0$ (caveat emptor).

Since by assumption, $V \neq 0$, the characteristic matrix must be singular, and thus its determinant must vanish. Setting determinant $(A - c \cdot I)$ equal to zero allows for the determination of the eigenvalues and eigenvectors of the matrix A .

If all characteristic roots (c) are positive then A is positive definite,
 else if all c 's are negative then A is negative definite,
 else if all c 's are nonnegative and at least one $c = 0$ then A is positive semidefinite,
 else if all c 's are nonpositive and at least one $c = 0$ then A is negative semidefinite,
 else some c 's are positive and others negative, and A is sign indefinite.

5.1 Example 8

Let A be the 2×2 matrix

```
(%i277) A : matrix ( [-6, 3], [3, -6] );
```

$$(A) \begin{pmatrix} -6 & 3 \\ 3 & -6 \end{pmatrix}$$

Let I be the 2×2 unit matrix

```
(%i278) I : ident (2);
```

$$(I) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

To find the characteristic roots (eigenvalues) of the matrix A , the determinant of the characteristic matrix $(A - c*I)$ is set equal to zero.

```
(%i279) c;
```

```
(%o279) c
```

```
(%i280) eqn : determinant (A - c*I) = 0;
```

$$(eqn) (-c-6)^2 - 9 = 0$$

```
(%i281) solns : solve( eqn, c);
```

```
(solns) [c=-9, c=-3]
```

```
(%i282) [c1, c2] : map ('rhs, solns);
```

```
(%o282) [-9, -3]
```

Testing for sign definiteness, since both characteristic roots (eigenvalues) are negative, the matrix A is negative definite.

5.2 Example 9

Continuing with Example 8, the first root $c_1 = -9$ is now used to find the characteristic vector (eigenvector) of the given matrix A .

```
(%i283) [v11, v12];
```

```
(%o283) [v11, v12]
```

```
(%i284) V1 : cvec ([v11, v12]);
```

```
(V1)  $\begin{pmatrix} v11 \\ v12 \end{pmatrix}$ 
```

We replace c by c_1 (as defined above in Example 8), and V by V_1 in the matrix equation $(A - c_1 I) \cdot V = 0$, we get a matrix equation which implies two ordinary equations.

```
(%i285) (A - c1*I) . V1 = zeromatrix (2,1);
```

```
(%o285)  $\begin{pmatrix} 3 v12 + 3 v11 \\ 3 v12 + 3 v11 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 
```

The two equations are identical and require $v_{12} = -v_{11}$, so add the normalization condition $\text{transpose}(V_1) \cdot V_1 = 1$, which is satisfied if $v_{11} = \pm 1/\sqrt{2}$

```
(%i286) V1 : at (V1, v12 = -v11);
```

```
(V1)  $\begin{pmatrix} v11 \\ -v11 \end{pmatrix}$ 
```

```
(%i287) eqn : transpose (V1) . V1 = 1;
```

```
(eqn)  $2 v11^2 = 1$ 
```

```
(%i288) solve (eqn, v11);
```

```
(%o288)  $[v11 = -\frac{1}{\sqrt{2}}, v11 = \frac{1}{\sqrt{2}}]$ 
```

We can choose either of the two solutions, choose the second.

```
(%i289) V1 : cvec ([ 1/sqrt(2), - 1/sqrt(2) ]);
```

```
(V1)  $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$ 
```

check normalization

(%i290) transpose (V1) . V1;

(%o290) 1

(%i291) A . V1 = c1*V1;

$$(\%o291) \begin{pmatrix} -\frac{9}{\sqrt{2}} \\ \frac{9}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} -\frac{9}{\sqrt{2}} \\ \frac{9}{\sqrt{2}} \end{pmatrix}$$

Next, look for the eigenvector V2, using $c = c_2$ as the corresponding eigenvalue of A.

(%i292) V2 : cvec ([v21, v22]);

$$(\%V2) \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$$

(%i293) (A - c2*I) . V2 = zeromatrix (2,1);

$$(\%o293) \begin{pmatrix} 3 v_{22} - 3 v_{21} \\ 3 v_{21} - 3 v_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Both equations imply we need $v_{22} = v_{21}$, and adding the normalization condition gives

(%i294) V2 : cvec ([1/sqrt(2), 1/sqrt(2)]);

$$(\%V2) \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

(%i295) transpose (V2) . V2;

(%o295) 1

(%i296) A . V2 = c2*V2;

$$(\%o296) \begin{pmatrix} -\frac{3}{\sqrt{2}} \\ -\frac{3}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} -\frac{3}{\sqrt{2}} \\ -\frac{3}{\sqrt{2}} \end{pmatrix}$$

5.3 eigenvalues (M)

Maxima has a function `eigenvalues (M)` which returns a list of two lists. The first list is a list of eigenvalues found, the second list is a list of multiplicities of the listed eigenvalues.

```
(%i297) [eivals, mult] : eigenvalues (A);
```

```
(%o297) [[-9, -3], [1, 1]]
```

```
(%i298) eivals;
```

```
(%o298) [-9, -3]
```

```
(%i299) [c1, c2] : eivals;
```

```
(%o299) [-9, -3]
```

```
(%i300) c1;
```

```
(%o300) -9
```

5.4 eigenvectors (M)

Maxima has a function `eigenvectors (M)` which returns a list of two lists. The first list is the same as is returned using `eigenvalues (M)` and has two lists: 1. a list of eigenvalues found, 2. a list of the multiplicities. The second list returned by `eigenvectors` contains a list of eigenvectors (in a list form) for each eigenvalue found.

```
(%i301) [EI, EV] : eigenvectors (A);
```

```
(%o301) [[[-9, -3], [1, 1]], [[1, -1], [1, 1]]]
```

```
(%i302) EI;
```

```
(%o302) [[-9, -3], [1, 1]]
```

```
(%i303) [eivals, mult] : EI;
```

```
(%o303) [[-9, -3], [1, 1]]
```

```
(%i304) eivals;
```

```
(%o304) [-9, -3]
```

```
(%i305) [c1, c2] : eivals;
```

```
(%o305) [-9, -3]
```

```
(%i306) c1;
```

```
(%o306) -9
```

```
(%i307) EV;
(%o307) [[[1,-1]],[[1,1]]]
```

```
(%i308) EV[1];
(%o308) [[1,-1]]
```

```
(%i309) EV[1][1];
(%o309) [1,-1]
```

```
(%i310) [v11, v12] : EV[1][1];
(%o310) [1,-1]
```

```
(%i311) v11;
(%o311) 1
```

```
(%i312) EV[2];
(%o312) [[1,1]]
```

```
(%i313) EV[2][1];
(%o313) [1,1]
```

```
(%i314) [v21, v22] : EV[2][1];
(%o314) [1,1]
```

At this point `eigenvectors(A)` has given us un-normalized eigenvector component in a list form. We can use (for example) our `cvec` function to convert to a column matrix as we ultimately need.

```
(%i315) V1 : cvec ([v11, v12]);
```

```
(V1)  $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 
```

V1 is not yet normalized.

```
(%i316) transpose (V1) . V1;
(%o316) 2
```

```
(%i317) V2 : cvec ([v21, v22]);
```

```
(V2)  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ 
```

```
(%i318) transpose (V2) . V2;
```

```
(%o318) 2
```

5.5 Problem 12.43

Use eigenvalues to determine sign definiteness for the matrix

```
(%i319) A : matrix ( [4, 6, 3], [0, 2, 5], [0, 1, 3] );
```

$$(A) \begin{pmatrix} 4 & 6 & 3 \\ 0 & 2 & 5 \\ 0 & 1 & 3 \end{pmatrix}$$

```
(%i320) [eivals, mult] : eigenvalues (A);
```

```
(%o320) [[- \frac{\sqrt{21}-5}{2}, \frac{\sqrt{21}+5}{2}, 4], [1, 1, 1]]
```

```
(%i321) [c1, c2, c3] : float (eivals);
```

```
(%o321) [0.20871, 4.7913, 4.0]
```

With all three eigenvalues positive, A is positive definite.

5.6 Problem 12.46 (a)

Find the characteristic roots (eigenvalues) for the matrix A:

```
(%i322) A : matrix ( [6, 6], [6, -3] );
```

$$(A) \begin{pmatrix} 6 & 6 \\ 6 & -3 \end{pmatrix}$$

```
(%i323) [eivals, mult] : eigenvalues (A);
```

```
(%o323) [[-6, 9], [1, 1]]
```

With one eigenvalue negative, and the other positive, A is sign indefinite.

6 Binding and Nonbinding Constraints in Economics

6.1 Reference

[https://intranet.royalholloway.ac.uk/economics/documents/pdf/courseformsandinfo/ec5555lecture6\(2013\)-constrainedoptimisation.pdf](https://intranet.royalholloway.ac.uk/economics/documents/pdf/courseformsandinfo/ec5555lecture6(2013)-constrainedoptimisation.pdf)

A constraint is binding if at the optimum the constraint function holds with equality (sometimes called an equality constraint) giving a boundary solution somewhere on the constraint itself.

Otherwise the constraint is non-binding or slack (sometimes called an inequality constraint).

If the constraint is binding we can use the Lagrangian technique.

Often we can use our economic understanding to tell us if a constraint is binding.

– Example: a non-satiated consumer will always spend all her income so the budget constraint will be satisfied with equality.

But in general we do not know whether a constraint will be binding ($=$, $>$ or $<$).

In this case we use a technique which is related to the Lagrangian, but which is slightly more general called linear programming, or in the case of non-linear inequality constraints, non-linear programming or Kuhn-Tucker programming after its main inventors.

6.2 Example 1

A firm chooses output x to maximize a profit function $\pi = -x^2 + 10x - 6$. Because of a labor shortage, the firm cannot produce an output higher than $x = 4$.

What are the objective and constraint functions?

The objective function: $\pi = -x^2 + 10x - 6$

The constraint: $x \leq 4$ or $0 \leq 4 - x$

(%i324) x;

(%o324) x

Maxima will not allow an assignment to be made to the symbol π . So use Pr to represent profit π ,

(%i326) Pr: $-x^2 + 10x - 6$;
soln : solve (diff (Pr, x), x);

(Pr) $-x^2 + 10x - 6$

(soln) [x=5]

(%i327) at (Pr, soln);

(%o327) 19

We can determine if this is a local maximum by looking at the sign of the second derivative.

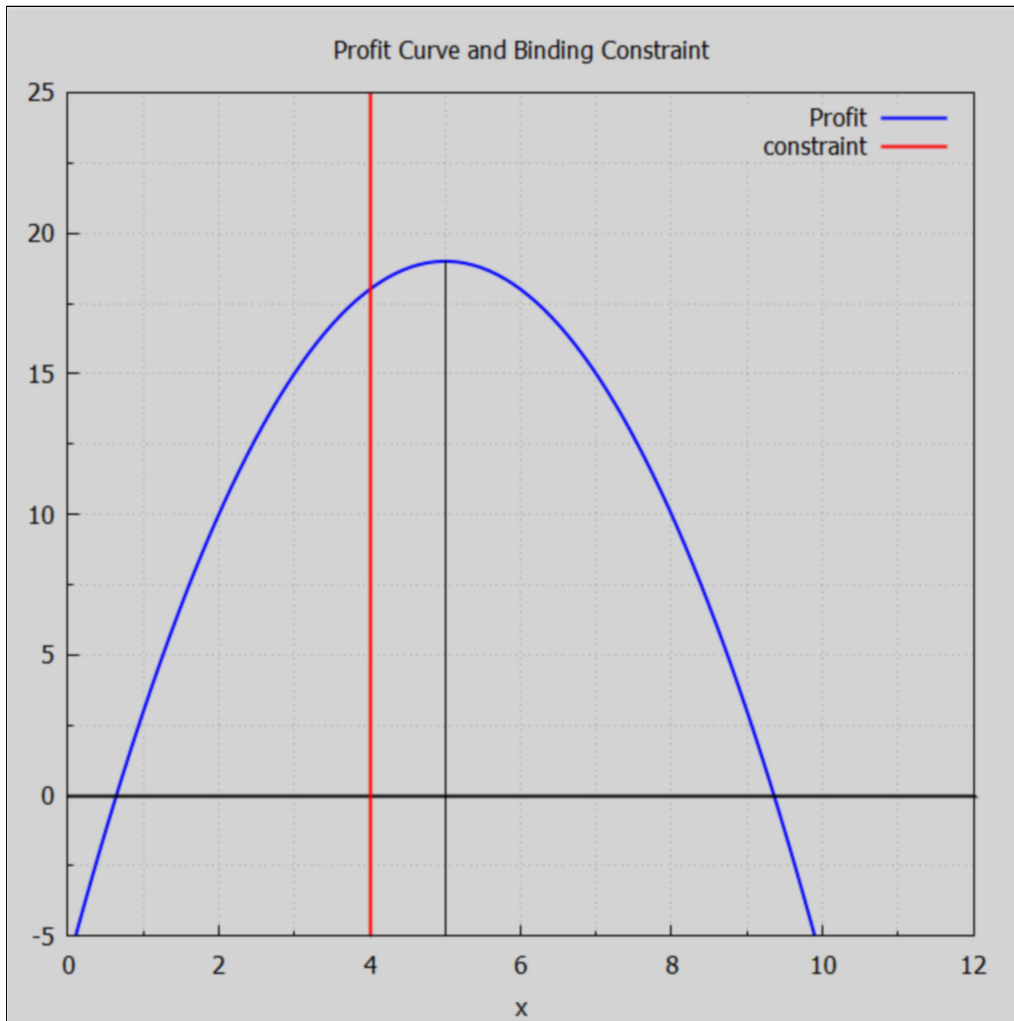
(%i328) at (diff (Pr, x, 2), soln);

(%o328) -2

So the profit Pr will be maximum if the output $x = 5$, but our constraint is that $x \leq 4$, so the constraint is binding. The optimum solution, taking into account the constraint, is $x = 4$, a solution "on the boundary" of the constraint $x \leq 4$ (as close to the constraint-free optimum as possible).

```
(%i329) wxdraw2d (xlabel = "x", yrange = [-5, 25],
    title = "Profit Curve and Binding Constraint",
    key = "Profit", explicit (Pr, x, 0, 12),
    color = black, key = "", explicit (0, x, 0, 12),
    line_width = 1, parametric (5, yy, yy, -5, 19), color = red,
    line_width = 2, key = "constraint", parametric (4, yy, yy, -5, 25) ),
    wxplot_size = [680, 680]$
```

(%t329)



```
(%i330) [at (Pr, x = 4), at (Pr, x = 5)];
```

```
(%o330) [18, 19]
```

So the given constraint $x \leq 4$ is binding in this problem. A hypothetical constraint $x \leq 6$ would NOT be binding, since the constraint-free optimum ($x = 5$) is accessible without violating the $x \leq 6$ constraint.