Free and Constrained Optimization

TABLE OF CONTENTS


```
(%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

Dowling11-12.wxmx uses Maxima to work some of the problems in Ch. 11 and 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");

We use the function killAB()$ at the start of some sections. This function is defined in Econ1.mac, and kills all bindings except for the functions defined in Econ1.mac (and thus avoids having to constantly reload Econ1.mac).

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

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

The main subjects of Ch. 11 and 12 are the use of matrices and determinants in economic analysis, such as rank of a matrix, the discriminant, leading principal minors, the Jacobian matrix, the Hessian matrix, the bordered Hessian matrix, and the use of eigenvalues determination of sign definiteness. Also discussed in some detail is the subject of Input-Output Analysis in an open economy.

Included here is a discussion of the relations between the Taylor expansion theorems and Hessian matrix tests for the nature of an extremum, and a Maxima derivation of the sign pattern criteria for leading principal minors of the Hessian matrix (for the cases of two and three variables) in detecting minima and maxima, which are not part of Dowling's text.

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

Quoting Math 2640, Introduction to Optimization (Leed's Univ.)
(http://webprod3.leeds.ac.uk/catalogue/dynmodules.asp?Y=202021&M=MATH-2640)

"Optimisation "the quest for the best" plays a major role in financial and economic theory, eg in maximising a company's profits or minimising its production costs. How to achieve such optimality is the concern of this course, which develops the theory and practice of maximising or minimising a function of many variables, either with or without constraints. This course lays a solid foundation for progression onto more advanced topics, such as dynamic optimisation, which are central to the understanding of realistic economic and financial scenarios."

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
----
text: 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 Mathematic methods for optimization.
https://reference.wolfram.com/language/tutorial/ConstrainedOptimizationIntroduction.html

3 *rank (amatrix), Linear Independence [11.1]*

If the determinant of a matrix equals zero, the determinant is said to "vanish" and the matrix is termed "singular".

A "singular matrix" is one in which there exists linear dependence between at least two rows or columns.

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.

If linear dependence exists in a system of equations, the system as a whole will have an infinite number of possible solutions, making a unique solution impossible.

The rank of a matrix is defined as the maximum number of linearly independent rows or columns in the matrix. Given a square matrix of order n
 if rank (A) = n, then A is nonsingular, |A| # 0, and there is no linear dependence.
 if rank (A) < n, then A is singular, |A| = 0, and there is linear dependence.

The Maxima function rank (M) computes the rank of the square matrix M.
The Maxima function determinant calculates the determinant of a matrix.
|M| can be found using determinant (M).

3.1 Example 1

```
(%i3) A : matrix ( [6, 4], [7, 9] );
      B : matrix ( [4, 6], [6, 9] );
```

(A) $\begin{pmatrix} 6 & 4 \\ 7 & 9 \end{pmatrix}$

(B) $\begin{pmatrix} 4 & 6 \\ 6 & 9 \end{pmatrix}$

The determinant of a 2 x 2 matrix is called a "second order determinant."
The determinant is only defined for a square matrix.

```
(%i5) determinant (A);
      determinant (B);
(%o4) 26
(%o5) 0
```

Since |A| # 0, the matrix A is nonsingular and there is no linear dependence between any of its rows and columns.

```
(%i6) rank (A);
(%o6) 2
```

The rank of A is 2 which is equal to its dimensions, indicating a nonsinular matrix.

Since |B| = 0, the matrix B is singular and linear dependence exists between its rows and columns. In this example, row 2 = 1.5 times row1, and col 2 = 1.5 times col 1.

(%i7) rank (B);
(%o7) 1

The rank of B is 1 which is less than the dimensions of B (n = 2), and since rank(B) = 1, there is only one linearly independent row and one linearly independent column in B.

4 The Minors of a Matrix   Minor (M, i, j)   [11.3]

We bind the symbol A now to a 3 x 3 matrix with elements a[i,j] using the Maxima function genmatrix. The a[i,j] elements are components of a Maxima "hash array." The elements a[i,j] are not (initially) bound to any specific values.

(%i8) A : genmatrix (a, 3, 3);

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

The (display2d = false) appearance of this result can be revealed using the Maxima function grind.

(%i9) grind(%)$

matrix([a[1,1],a[1,2],a[1,3]],[a[2,1],a[2,2],a[2,3]],[a[3,1],a[3,2],a[3,3]])$

The *Maxima function* minor (M, i, j) returns a *submatrix* of M gotten by removing from M row i and column j. In conventional mathematics terminology, a "minor of a matrix" is not a submatrix but rather a determinant of the submatrix. Dowling p. 226 uses the term "subdeterminant" for a minor of a matrix.

(%i10) minor (A, 1,1);

(%o10) $\begin{pmatrix} a_{2,2} & a_{2,3} \\ a_{3,2} & a_{3,3} \end{pmatrix}$

(%i11) grind(%)$

matrix([a[2,2],a[2,3]],[a[3,2],a[3,3]])$

The Maxima function Minor (M, i, j), defined in Econ1.mac, conforms with conventional math terminology (and Dowling), producing a scalar by taking the determinant of a submatrix:

(%i12) fundef (Minor);

(%o12) Minor(*MM*, *mm*, *nn*) := determinant(minor(*MM*, *mm*, *nn*))

Here we find the determinant of the submatrix defined by deleting row 1 and column 1 of the 3 x 3 matrix A, using our Maxima function Minor.

```
(%i13) Minor (A, 1, 1);
```

(%o13) $a_{2,2}\,a_{3,3}-a_{2,3}\,a_{3,2}$

```
(%i14) grind(%)$
```

a[2,2]*a[3,3]-a[2,3]*a[3,2]$

```
(%i15) Minor (A, 3, 3);
```

(%o15) $a_{1,1}\,a_{2,2}-a_{1,2}\,a_{2,1}$

5 Leading Principal Minors of a Matrix LPM (M, j) [12.2]

For an n x n square matrix there are n "leading principal minors".

A basic minor of a matrix is the determinant of a square matrix that is of maximal size with nonzero value. For an n × n nonsingular square matrix, there are n leading principal minors.

For the matrix A, a square 3 x 3 matrix, there are 3 "leading principal minors" which we can obtain using our Maxima function LPM (amatrix, num), defined in Econ1.mac.

The leading principal minor A1 is simply the element A[1,1], produced by deleting every row but the first, and deleting every column but the first.

```
(%i17) A[1,1];
       LPM (A, 1);
```

(%o16) $a_{1,1}$

(%o17) $a_{1,1}$

The leading principal minor A2 is the determinant of the submatrix produced by deleting all but the first 2 rows and columns of A.

```
(%i18) LPM (A, 2);
```

(%o18) $a_{1,1}\,a_{2,2}-a_{1,2}\,a_{2,1}$

The leading principal minor A3 is the determinant of the matrix produced by deleting all but the first 3 rows and columns of A, which, since A is a 3 x 3 matrix, is simply the whole matrix A, since A is defined as a 3 x 3 matrix.

(%i19) LPM (A, 3);

(%o19) $a_{1,1}(a_{2,2}a_{3,3}-a_{2,3}a_{3,2})-a_{1,2}(a_{2,1}a_{3,3}-a_{2,3}a_{3,1})+a_{1,3}(a_{2,1}a_{3,2}-a_{2,2}a_{3,1})$

(%i20) determinant (A);

(%o20) $a_{1,1}(a_{2,2}a_{3,3}-a_{2,3}a_{3,2})-a_{1,2}(a_{2,1}a_{3,3}-a_{2,3}a_{3,1})+a_{1,3}(a_{2,1}a_{3,2}-a_{2,2}a_{3,1})$

6 jacobian (funcList, varList), Linear Indepence [12.1]

Maxima has the function
jacobian (funcList, varList)
which computes the Jacobian matrix, which can be used to test for functional independence, both linear and nonlinear. A Jacobian matrix is composed of all the first-order partial derivatives of a system of equations, arranged in an ordered sequence. The list [f1, f2, f3] is interpreted as [f1 = 0, f2 = 0, f3 = 0] is the list of equations to be solved.

jacobian (funcList, varList) returns the Jacobian matrix of the list of functions funcList with respect to the list of variables varList. The (i, j)-th element of the Jacobian matrix is diff (funcList[i], varList[j]).

Here is a symbolic example, using Maxima, for jacobian ([f1, f2, f3], [x1, x2, x3]).

(%i21) depends ([f1, f2, f3], [x1, x2, x3]);

(%o21) [f1(x1,x2,x3),f2(x1,x2,x3),f3(x1,x2,x3)]

(%i22) J : jacobian([f1, f2, f3], [x1, x2, x3]);

(J) $$\begin{pmatrix} \frac{d}{d\,x1}f1 & \frac{d}{d\,x2}f1 & \frac{d}{d\,x3}f1 \\ \frac{d}{d\,x1}f2 & \frac{d}{d\,x2}f2 & \frac{d}{d\,x3}f2 \\ \frac{d}{d\,x1}f3 & \frac{d}{d\,x2}f3 & \frac{d}{d\,x3}f3 \end{pmatrix}$$

(%i23) grind(%)$

```
matrix(['diff(f1,x1,1),'diff(f1,x2,1),'diff(f1,x3,1)],
      ['diff(f2,x1,1),'diff(f2,x2,1),'diff(f2,x3,1)],
      ['diff(f3,x1,1),'diff(f3,x2,1),'diff(f3,x3,1)])$
```

The first row elements are the first derivatives of f1 with respect to all three variables.
The first column elements are the first derivatives of all the functions with respect to x1.

If we are given a set of three equations
f1 = 0, f2 = 0, f3 = 0,
then if the determinant of the jacobian is NOT equal to zero, the three equations ARE functionally independent.

If the determinant of the jacobian IS equal to zero, then the equations are NOT independent.

If the list of equations are actually a list of the first derivatives of an expression with respect to the independent variables, the determinant of the Jacobian matrix is the same as the determinant of the Hessian matrix (vide infra), which computes all second order derivatives of the given expression, and the latter determinant is, in turn, the same as the "n'th leading principal minor" of that Hessian n x n matrix (where the number of independent variables is n).

6.0.1 Example 1

Given the pair of equations
f1 = 5 x1 + 3 x2 = 0,
f2 = 25 x1^2 + 30 x1 x2 + 9 x2^2 = 0,
we first set up the jacobian matrix of first derivatives.

The Maxima function killAB() is defined in Econ1.mac. This function kills all except the functions defined in Econ1.mac, so we don't have to keep reloading Econ1.mac when we want to use its functions again.

f1 and f2 are defined as Maxima expressions (rather than Maxima functions).
The implied equations are f1 = 0 and f2 = 0.

```
(%i3) killAB()$
      f1 : 5*x1  + 3*x2;
      f2 : 25*x1^2 + 30*x1*x2 + 9*x2^2;
      J : jacobian ([f1, f2], [x1, x2]);
```

(f1) $3\,x2+5\,x1$

(f2) $9\,x2^2+30\,x1\,x2+25\,x1^2$

(J) $$\begin{pmatrix} 5 & 3 \\ 30\,x2+50\,x1 & 18\,x2+30\,x1 \end{pmatrix}$$

Then we use the Maxima functions determinant and expand to evaluate the determinant of the jacobian matrix (Dowling refers to this as the "Jacobian determinant") symbolically.

```
(%i4) JD : determinant (J), expand;
(JD)  0
```

Since the determinant of the jacobian evaluates to zero, the two equations f1 = 0, and f2 = 0, are not functionally independent. The functional dependence is easily seen if we examine f1^2.

```
(%i6) f1^2, expand;
      f2;
```

(%o5) $9\,x2^2 + 30\,x1\,x2 + 25\,x1^2$

(%o6) $9\,x2^2 + 30\,x1\,x2 + 25\,x1^2$

7 *Quadratic Forms and the Discriminant [12.3]*

Determinants (in the form of leading principle minors) may be used to test for positive or negative definiteness of any quadratic form. The determinant |D| of a quadratic form is called a "discriminant".

The matrix D referred to here is a symmetric matrix such that a quadratic form q in an arbitrary number of variables can be written as
q = transpose(u) . (D u)
in which u is written as a matrix column vector.

A quadratic form involving two variables (x,y) might be given as
q = a x^2 + b x y + c y^2,
and we can find a symmetric matrix D by placing the coefficients of the square terms on the principle diagonal and dividing the coefficients of the nonsquared term equally between the off-diagonal positions:

```
(%i7) D : matrix ( [a, b/2], [b/2, c]);
```

(D) $\begin{pmatrix} a & \frac{b}{2} \\ \frac{b}{2} & c \end{pmatrix}$

We then evaluate the leading principal minors.

(%i8) D1 : LPM (D,1);

(D1) a

(%i9) D2 : LPM (D,2);

(D2) $a\,c-\frac{b^2}{4}$

If D1 > 0 and D2 > 0, then q is positive definite and q is positive for all values of the variables (x,y) as long as x and y are not both zero.

If D1 < 0 and D2 > 0, then q is negative definite and q is negative for all values of the variables (x,y) as long as x and y are not both zero.

Note that D2 is just the determinant of D. If D2 is not greater than zero, q is not "sign-definite" and q may assume both positive and negative values.

7.1 Two Variable Quadratic Form LPM Test

We can derive these conditions, starting with our general two variable quadratic form expression q = a x^2 + b x y + c y^2, by adding and subtracting a term proportional to y^2 with a coefficient chosen to allow us to write q in the form
  q = a (x + d y)^2 + e y^2
We need to find the values of d and e which makes these alternative expressions equal.

In the following, we want expr to equal zero.

(%i12) ex0 : a*x^2 + b*x*y + c*y^2;
ex1 : a*(x + d*y)^2 + e*y^2;
expr : ex0 - ex1, expand;

(ex0) $c\,y^2+b\,x\,y+a\,x^2$

(ex1) $a\,(d\,y+x)^2+e\,y^2$

(expr) $-e\,y^2-a\,d^2\,y^2+c\,y^2-2\,a\,d\,x\,y+b\,x\,y$

Now we want the coefficient of y^2 to equal zero, and separately, we want the coefficient of x*y to equal zero, which gives us two equations in two unknowns.

(%i13) ey2 : coeff(expr,y,2);

(ey2) $-e-a\,d^2+c$

(%i14) exy : ratcoef (expr, x*y);

(exy) $b-2\,a\,d$

(%i15) soln : solve ([ey2, exy], [d, e]);

(soln) $[[d=\frac{b}{2a}, e=\frac{4ac-b^2}{4a}]]$

(%i16) soln : soln[1];

(soln) $[d=\frac{b}{2a}, e=\frac{4ac-b^2}{4a}]$

(%i17) e : at(e, soln);

(e) $\frac{4ac-b^2}{4a}$

Writing the two variable quadratic form q in the form of ex1

(%i18) ex1;

(%o18) $a\,(d\,y+x)^2+e\,y^2$

q is definitely positive if x and y are not both equal to zero, and if a > 0 and if e > 0.
But a = D1, and e is D2/D1.

(%i19) [D1, D2, expand(D2/D1), expand (e)];

(%o19) $[a, a\,c-\frac{b^2}{4}, c-\frac{b^2}{4a}, c-\frac{b^2}{4a}]$

So the general two variable quadratic can be written as
q = D1 (x + d y)^2 + (D2/D1) y^2

Since the symbols e, D1, and D2 are already bound to expressions, let's kill the binding of all three first.

(%i21) kill(e, D1, D2)$
subst ([a = D1, e = D2/D1], ex1);

(%o21) $D1\,(d\,y+x)^2+\frac{D2\,y^2}{D1}$

In this form, it is clear that (with x and y not both zero)
q is positive definite (PD) if D1 > 0 and D2 > 0,
q is negative definite (ND) if D1 < 0 and D2 > 0.

7.2 Example 3 (2 var)

Given the quadratic form
z = 2 x^2 + 5 x y + 8 y^2,
form the symmetric matrix D as described above and test for sign-definiteness of the given quadratic form.

(%i22) D : matrix ([2, 5/2], [5/2, 8]);

(D) $\begin{pmatrix} 2 & \frac{5}{2} \\ \frac{5}{2} & 8 \end{pmatrix}$

(%i23) LPM (D, 1);

(%o23) 2

(%i24) LPM (D, 2);

(%o24) $\frac{39}{4}$

(%i25) %, numer;

(%o25) 9.75

Since both D1 and D2 are greater than zero, z is positive definite and z is positive for all nonzero values of the variables (x,y).

7.2.1 Qtest (amatrix)

The Maxima function Qtest, defined in Econ1.mac, tests a symmetric numerical matrix , and prints a list of lists: [["LPM1", LPM1], ["LPM2", LPM2],...,["LPMn", LPMn]]. with the values of the leading principal minors of the given matrix.

(%i26) Qtest (D);

positive definite

$[[LPM1,2],[LPM2,\frac{39}{4}]]$

(%o26) [2.0,9.75]

7.3 Three Variable Quadratic Form LPM Test

Start with general three variable quadratic form
q = a x^2 + b x y + c x z + d y^2 + e y z + f z^2,
which can be written as a matrix equation with D a symmetric 3 x 3 matrix of coefficients and u a matrix column vector with elements (x,y,z), such that
q = transpose(u) . D . u

```
(%i1) killAB()$
D : matrix ( [a, b/2, c/2],  [b/2, d, e/2], [c/2, e/2, f] );
```

(D) $$\begin{pmatrix} a & \frac{b}{2} & \frac{c}{2} \\ \frac{b}{2} & d & \frac{e}{2} \\ \frac{c}{2} & \frac{e}{2} & f \end{pmatrix}$$

```
(%i2) u : matrix ( [x], [y], [z] );
```

(u) $$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

```
(%i3) transpose (u) . D . u, expand;
```

(%o3) $f\,z^2+e\,y\,z+c\,x\,z+d\,y^2+b\,x\,y+a\,x^2$

which equals our starting expression for a third order quadratic q.

We now calculate the three leading principal minors of the matrix D.

```
(%i4) D1 : LPM (D, 1);
```

(D1) a

```
(%i5) D2 : LPM (D, 2);
```

(D2) $a\,d-\frac{b^2}{4}$

```
(%i6) D3 : LPM (D, 3);
```

(D3) $$a\left(d\,f-\frac{e^2}{4}\right)-\frac{b\left(\frac{b\,f}{2}-\frac{c\,e}{4}\right)}{2}+\frac{c\left(\frac{b\,e}{4}-\frac{c\,d}{2}\right)}{2}$$

We now want to prove that (with x, y and z not all zero)
q is positive definite (PD) if D1 > 0 and D2 > 0 and D3 > 0,
q is negative definite (ND) if D1 < 0 and D2 > 0, and D3 < 0.

We look for g, h, k, l, and m (5 unknowns) such that we can write q as a sum of squares in the form (with a, b, c, d, e, f taken as given):

q = a (x + g y + h z)^2 + k (y + l z)^2 + m z^2

If a, k, and m are all positive, q is positive for any values of (x,y,z) (not all zero), and q is positive definite.
If a, k, and m are all negative, q is negative for any value of (x,y,z) (not all zero), and q is negative definite.

Let ex0 be the general starting form of q, ex1 be the alternative expression of q, and expr be the difference between ex0 and ex1.

We require expr to be equal to zero in order to determine g, h, k, l, and m.

(%i9)
```
ex0 : a*x^2 + b*x*y + c*x*z + d*y^2 + e*y*z + f*z^2;
ex1 : a*(x + g*y + h*z)^2 + k*(y + l*z)^2 + m*z^2;
expr : ex0 - ex1, expand;
```

(ex0) $f z^2+e y z+c x z+d y^2+b x y+a x^2$

(ex1) $k (l z+y)^2+a (h z+g y+x)^2+m z^2$

(expr) $-m z^2-k l^2 z^2-a h^2 z^2+f z^2-2 k l y z-2 a g h y z+e y z-2 a h x z+c x z-k y^2-a g^2 y^2+d y^2-2 a g x y+b x y$

We want ex0 and ex1 to be equal, hence expr equal to zero. Hence the coefficient of z^2 to equal zero, the coefficient of x*y to be zero, etc.

(%i10)
```
ez2 : coeff (expr,z^2);
```

(ez2) $-m-k l^2-a h^2+f$

(%i11)
```
exy : ratcoef (expr, x*y);
```

(exy) $b-2 a g$

(%i12)
```
exz : ratcoef (expr, x*z);
```

(exz) $c-2 a h$

(%i13)
```
ey2 : coeff (expr, y^2);
```

(ey2) $-k-a g^2+d$

(%i14) eyz : ratcoef (expr, y*z);

(eyz) $-2\,k\,l-2\,a\,g\,h+e$

We then ask solve to take the five equations, ez2 = 0, exy = 0, etc and come up with symbolic solutions for the five unknowns.

(%i15) solns : solve ([ez2, exy, exz, ey2, eyz], [g, h, k, l, m]);

(solns) $$[[g=\frac{b}{2\,a},h=\frac{c}{2\,a},k=\frac{4\,a\,d-b^2}{4\,a},l=\frac{2\,a\,e-b\,c}{4\,a\,d-b^2},m=\frac{(4\,a\,d-b^2)\,f-a\,e^2+b\,c\,e-c^2\,d}{4\,a\,d-b^2}]]$$

(%i16) soln : solns[1];

(soln) $$[g=\frac{b}{2\,a},h=\frac{c}{2\,a},k=\frac{4\,a\,d-b^2}{4\,a},l=\frac{2\,a\,e-b\,c}{4\,a\,d-b^2},m=\frac{(4\,a\,d-b^2)\,f-a\,e^2+b\,c\,e-c^2\,d}{4\,a\,d-b^2}]$$

We bind the symbol k to the result returned in soln.

(%i17) k : at (k, soln);

(k) $$\frac{4\,a\,d-b^2}{4\,a}$$

We want to show that k is the same as D2/D1.

(%i18) [D1, D2, expand (D2/D1), expand (k)];

(%o18) $$[a,a\,d-\frac{b^2}{4},d-\frac{b^2}{4\,a},d-\frac{b^2}{4\,a}]$$

We see that k = D2/D1.

Next bind the symbol m to the result returned in soln.

(%i19) m : at (m, soln);

(m) $$\frac{(4\,a\,d-b^2)\,f-a\,e^2+b\,c\,e-c^2\,d}{4\,a\,d-b^2}$$

We want to show that m is the same as D3/D2.

(%i20) [expand (D3/D2), expand (m)];

(%o20) $$\left[-\frac{b^2 f}{4\,a\,d-b^2}+\frac{a\,d\,f}{a\,d-\frac{b^2}{4}}-\frac{a\,e^2}{4\,a\,d-b^2}+\frac{b\,c\,e}{4\,a\,d-b^2}-\frac{c^2 d}{4\,a\,d-b^2},\frac{4\,a\,d\,f}{4\,a\,d-b^2}-\frac{b^2 f}{4\,a\,d-b^2}-\frac{a\,e^2}{4\,a\,d-b^2}+\frac{b\,c\,e}{4\,a\,d-b^2}-\frac{c^2 d}{4\,a\,d-b^2}\right]$$

(%i21) is (equal (expand (D3/D2), expand (m)));

(%o21) ***true***

So the general three variable quadratic can be written as
q = D1 (x + g y + h z)^2 + (D2/D1) (y + l z)^2 + (D3/D2) z^2

Since the symbols k, m, D1, D2, and D3 are already bound to expressions, let's kill the binding of all five first.

(%i23) kill (k, m, D1, D2, D3)$
subst ([a = D1, k = D2/D1, m = D3/D2], ex1);

(%o23) $$\frac{D2\,(l\,z+y)^2}{D1}+D1\,(h\,z+g\,y+x)^2+\frac{D3\,z^2}{D2}$$

In this form, it is clear that (with x, y and z not all zero)
q is positive definite (PD) if D1 > 0 and D2 > 0 and D3 > 0,
q is negative definite (ND) if D1 < 0 and D2 > 0, and D3 < 0.
QED.

7.4 Problem 12.6 (3 var)

Given the quadratic form in three variables (x1, x2, x3)
y = 5 x1^2 - 6 x1 x2 + 3 x2^2 - 2 x2 x3 + 8 x3^2 - 3 x1 x3,
check for sign definiteness.

The coefficients of the squared terms continue to go on the principal diagonal, while the coefficient of x1 x2 is equally divided between D[1,2] and D[2,1], etc.

(%i1) killAB()$
D : matrix ([5, -3, -3/2], [-3, 3, -1], [-3/2, -1, 8]);

(D) $\begin{pmatrix} 5 & -3 & -\frac{3}{2} \\ -3 & 3 & -1 \\ -\frac{3}{2} & -1 & 8 \end{pmatrix}$

(%i2) D1 : LPM (D, 1);

(D1) 5

(%i3) D2 : LPM (D, 2);

(D2) 6

(%i4) D3 : LPM (D, 3);

(D3) $\frac{109}{4}$

(%i5) %, numer;

(%o5) 27.25

D1, D2, and D3 are all positive and thus the quadratic form y is positive definite and y is positive for all values of (x1, x2, x3) such that not all three are simultaneously zero.

Note that D3 is the same as the determinant of D.

(%i6) Qtest (D);

positive definite

$[[LPM1,5],[LPM2,6],[LPM3,\frac{109}{4}]]$

(%o6) [5.0,6.0,27.25]

7.5 Problem 12.7 (a) (3 var)

y = - 2 x1^2 + 4 x1 x2 - 5 x2^2 + 2 x2 x3 - 3 x3^2 + 2 x1 x3.

(%i1) killAB()$
D : matrix ([-2, 2, 1], [2, -5, 1], [1, 1, -3]);

(D) $\begin{pmatrix} -2 & 2 & 1 \\ 2 & -5 & 1 \\ 1 & 1 & -3 \end{pmatrix}$

(%i2) D1 : LPM (D, 1);
(D1) −2

(%i3) D2 : LPM (D, 2);
(D2) 6

(%i4) D3 : LPM (D, 3);
(D3) −7

Since D1 < 0, D2 > 0, and D3 > 0, y is a negative definite quadratic form and
y < 0 for all values of x1, x2, x3 as long as they are not all zero simultaneously.

(%i5) Qtest (D);
negative definite
[[LPM1,−2],[LPM2,6],[LPM3,−7]]
(%o5) *[−2.0,6.0,−7.0]*

8 Taylor's Theorem and the Hessian Matrix

In 1 dimension, Taylor's theorem, expanding about the point x, and with dx a vanishingly small scalar of either sign,

f (x + dx) = f(x) + dx fx + (1/2!) dx^2 fxx + ...

In n dimensions, Taylor's theorem is, with x standing for variables
(x1, x2, ..., xn) and dx standing for (dx1, dx2,....,dxn), and in the following
the third term assumes dx is a matrix column vector:

f(x + dx) = f(x) + dx . grad(f) + (1/2!) transpose(dx) . (H dx) + ...,
where H is the n x n Hessian matrix with components
H[i,j] = diff (f, xi,1,xj,1) if i # j,
H[i,j] = diff (f, xi, 2) if i = j.

The general Taylor series formula is derived by considering f(x + λ dx) as a function of λ, and expanding about λ = 0 using the Taylor expansion formula for a function of one variable, λ, together with the chain rule. Then putting λ = 1 gives the above expansion theorem in n dimensions.

For n = 2, we have the two variable case of the Taylor expansion theorem, expanding about the point (x,y), and with h and k vanishingly small scalars of either sign,

f(x + h, y + k) = f(x, y) + h ∂f/∂x + k ∂f/∂y +
(1/2) (h^2 ∂^2f/∂x^2 + 2 h k ∂^2f/∂x∂y + k^2 ∂^2f/∂y^2) + ...

We can write the quadratic term in terms of the Hessian matrix, making use of Young's theorem: ∂²f/∂x∂y = ∂²f/∂y∂x. In the following we use the Maxima function hessian (F, [X1, X2]) for a function of two variables. Here, f is not bound to any Maxima expression or Maxima function, so we simply tell Maxima that there is some theoretical dependence of f on x and y.

```
(%i1) killAB()$
      depends (f, [x, y]);
```

(%o1) $[f(x,y)]$

That allows us to construct a symbolic 2 x 2 Hessian matrix, which we call H here.

```
(%i2) H : hessian (f, [x, y]);
```

(H) $$\begin{pmatrix} \frac{d^2}{d\,x^2} f & \frac{d^2}{d\,x\,d\,y} f \\ \frac{d^2}{d\,x\,d\,y} f & \frac{d^2}{d\,y^2} f \end{pmatrix}$$

The 2 x 2 Hessian matrix is composed of the second order direct partials (like fxx) on the principle diagonal, and the second order cross partials (like fxy) off the principal diagonal. Maxima automatically uses the general result that ∂²f/∂x∂y = ∂²f/∂y∂x. and H is then obviously symmetric.

Let dX be a column matrix with elements (h,k):

```
(%i3) dX : matrix ([h], [k] );
```

(dX) $$\begin{pmatrix} h \\ k \end{pmatrix}$$

The transpose of dX is a row vector with elements (h,k).

```
(%i4) transpose (dX);
```

(%o4) $\begin{pmatrix} h & k \end{pmatrix}$

We can multiply conformable Maxima matrices using spaces and a single period (.)

```
(%i5) transpose (dX) . dX;
```

(%o5) k^2+h^2

(%i6) transpose (dX) . H . dX;

(%o6) $$k\left(\left(\frac{d^2}{d\,y^2}\,f\right)k+\left(\frac{d^2}{d\,x\,d\,y}\,f\right)h\right)+h\left(\left(\frac{d^2}{d\,x\,d\,y}\,f\right)k+\left(\frac{d^2}{d\,x^2}\,f\right)h\right)$$

(%i7) expand(%);

(%o7) $$\left(\frac{d^2}{d\,y^2}\,f\right)k^2+2\left(\frac{d^2}{d\,x\,d\,y}\,f\right)h\,k+\left(\frac{d^2}{d\,x^2}\,f\right)h^2$$

which is (except for the overall multiplier (1/2)) the quadratic term in the Taylor expansion of f(x+h, y+k).

We can also show the third order symbolic Hessian matrix form.

(%i2) killAB()$
depends (f, [x,y,z])$
hessian (f, [x,y,z]);

(%o2) $$\begin{pmatrix}\frac{d^2}{d\,x^2}\,f & \frac{d^2}{d\,x\,d\,y}\,f & \frac{d^2}{d\,x\,d\,z}\,f\\ \frac{d^2}{d\,x\,d\,y}\,f & \frac{d^2}{d\,y^2}\,f & \frac{d^2}{d\,y\,d\,z}\,f\\ \frac{d^2}{d\,x\,d\,z}\,f & \frac{d^2}{d\,y\,d\,z}\,f & \frac{d^2}{d\,z^2}\,f\end{pmatrix}$$

Notice that the Maxima function hessian automatically uses Young's theorem:
$\partial^2 f/\partial y\partial x = \partial^2 f/\partial x\partial y$.

9 *Unconstrained Optimization*

If there are no constraints, we have the case "free optimization".
Let f(x1, x2, ..., xn) be the "objective function" of n variables.

9.1 optimum (f, varL), optimumAll (f, varL)

The quickest route to finding optimum values is to use optimum(f,varL) defined in our software file Econ1.mac. optimum screens out values returned by solve that include imaginary parts and given real values, screens out negative values. optimum is really meant for Economics problems in which the search variables are real and non-negative.

If you want to accept all solutions returned by solve, then use optimumAll, with the same syntax otherwise. See our examples.

In more detail, optimum (f, [x1,x2,....,xn]); or optimumAll (f, [x1,x2,...,xn]);

9.2 Do it "by hand"...

Otherwise, do it "by hand" (with some help from our other functions).

1. Find the point(s) for which grad(f) = 0. A list of the expressions diff(f, xj) can be found using gradf : jacobian ([f], [x1, x2, ..., xn])[1];

If the derivatives involve simple polynomials, you can then use Maxima's solve function as in:
solns : solve (gradf, [x1, x21, x3,..., xn]).
(If solve has difficulties, or you want to check on solve, you can try to_poly_solve.)

If solve is successful is finding some or all of the solutions you will get a list of a list, as in [[x1 = 2, x2 = -1/2,...]] for one critical point, which you can extract using
cp : solns[1] --> [x1 = 2, x2 = -1/2,...].

If solve returns a list of , say, two critical points where the first derivatives of f vanish, like
[[x1 = 2, x2 = -1/2,...], [x1 = 5, x2 = -4,....]],
you could extract these two critical points using

[cp1, cp2] : solns;

and then cp1 is bound to the list [x1 = 2, x2 = -1/2,...], and cp2 is bound to the list [x1 = 5, x2 = -4,....].

2. You can then find the value of the objective function f at each of these points using
fcp1 : at (f, cp1),
fcp2 : at (f, cp2).
That will quickly give you a clue as to which is the "biggest" and which is the "smallest" value of f.

3. To see if a candidate critical point cp satisfies the second order conditions for a relative maximum or minimum,
a.) Calculate the Hessian matrix of f, using the Maxima function hessian (expr, varList), for example,
H : hessian (f, [x1, x2, ..., xn]) --> n x n square matrix of second derivatives of f.

b.) H could turn out to be a purely numerical matrix. In that case, proceed with step c. Otherwise, we recommend that you turn H into a purely numerical matrix which will depend on the numerical values of the critical point being investigated.

For example
Hcp1 : at (H, cp1);

c.) Calculate the "leading principal minors" of the purely numerical Hessian matrix Hcp1. You can use the Maxima function LPM (M, k) for k = 1, 2,..., n.
For example
H1 : LPM (Hcp1, 1);
H2 : LPM (Hcp1, 2;
.....
Hn : LPM (Hcp1,n);

9.3 Sufficient Conditions vs. Necessary Conditions

We quote from Ch. 1 of Optimization Methods in Economics by John Baxley, retired professor in the Mathematics Dept. at Wake Forest Univ.

https://users.wfu.edu/~baxley/m254book.pdf

Max-min problems play a central role in every calculus course. Finding relative (local) maxima and minima using the derivative and applying the first or second derivative test is the name of the game in curve-sketching as well as the “applied” problems in the calculus books.

The student who comes to economics from such calculus courses often feels betrayed. Slowly it becomes evident that economists do not spend their time finding maxima and minima. In fact, quite the opposite is true. Unlike the typical math problem where one “finds the maximum”, the economist assumes that the economic agent (firm, consumer, etc.) is instinctively maximizing. The fundamental assumption is that somehow such economic agents have a built in computer or natural instinct which leads them to maximizing behavior. The central question for the economist is not: find the maximum, but: how will the agent adjust maximizing behavior if some variable which he cannot control undergoes a change.

For example, how will the quantity of snack crackers sold in the marketplace change if the price of a related good like Coca-cola rises? This question assumes that consumers are maximizing their utility and as they face higher prices for Coke, they will make adjustments in their expenditures which may effect the amount of snack crackers which are sold.

One of the interesting sidelights of this state of affairs is that economists deeply wish that second order conditions were necessary rather than sufficient for an optimum. You will recall the second derivative test:

If c is a critical point of a function f(x) (i.e. f'(c) = 0), then f''(c) < 0 is a sufficient condition for c to be a maximum. However, this condition is not necessary, for f(x) = 9 − x^4 has a maximum at the critical point c = 0 but f''(0) = 0.

If we are math students and are on the prowl for maxima, the second derivative test can be used to determine if c is a maximum, but if we are economists we want the thought to flow the other way: if we know that c gives a maximum, we would like to conclude that f''(c) < 0. Unfortunately, we can only conclude that f''(c) ≤ 0.

The possibility that f''(c) = 0 is often disastrous for economic analysis, because in the analysis this value occurs in a denominator and leads to division by zero. Wishful thinking has led many economists to argue that this disaster is somehow very unlikely and can be safely ignored; some economists have actually referred to the disaster as “pathological” in nature. The example f(x) = 9−x^4 does not look very pathological!

As we proceed, you will find that we will often have to assume that second order conditions known only to be sufficient actually hold at the maximum. We don’t really have a viable choice. Either we make this assumption and draw an interesting economics conclusion, or we don’t make the assumption and no conclusion can be drawn.

9.4 Example 2: Extremum of f(x, y) [12.2]

Let z be a Maxima expression corresponding to: z = 3 x^2 - x y + 2 y^2 - 4 x - 7 y + 12.
Find any critical points and use the Hessian matrix test to examine the nature of such.

I will use f instead of z.

(%i3) f : 3*x^2 - x*y + 2*y^2 - 4*x - 7*y + 12;

(f) $2\,y^2-x\,y-7\,y+3\,x^2-4\,x+12$

9.4.1 optimum(f,varL)

The quickest route is optimum (f,[x,y]).

(%i4) optimum (f, [x, y]);

lagrangian = $2\,y^2-x\,y-7\,y+3\,x^2-4\,x+12$

cp1 $[x=1,y=2]$ *, relative minimum, value =* 3.0

(%o4) *done*

optimum has found one (real and non-negative) critical point (x=1,y=2) at
which f has a local minimum. For this kind of problem, optimum calls solve and CPtest.
The solutions returned by solve are screened to avoid solutions which include
imaginary numbers, and given real numbers, avoids non-negative values for (x,y).

optimum defines a global list cp.

(%i5) cp;

(%o5) $[[x=1,y=2]]$

(%i6) cp[1];

(%o6) $[x=1,y=2]$

(%i7) at (f, cp[1]);

(%o7) 3

9.4.2 Do it "by hand"...

Let's now take the "long way home" and do things more or less "by hand".
We first let gradf be a list of the first derivatives of f wrt x and y. The fastest way
is to use the Maxima function jacobian, as shown here:

(%i8) gradf : jacobian ([f], [x, y])[1];

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

```
(%i9)    solns : solve (gradf, [x, y]);
(solns)  [[x=1,y=2]]
```

Maxima's solve function has found one critical point. Let's call the "replacement rules" cp1.

```
(%i10)   cp1 : solns[1];
(cp1)    [x=1,y=2]
```

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

You can either use

```
(%i11)   subst (cp1, f);
(%o11)   3
```

or

```
(%i12)   at (f, cp1);
(%o12)   3
```

or the interactive work form:

```
(%i13)   f, cp1;
(%o13)   3
```

Next we check the second order conditions (SOC) for a relative optimum, given a critical point which satisfies the first order condition (FOC) that the first derivatives (wrt x and y) of f simultaneously vanish.

Define the Hessian matrix H, using the Maxima function hessian (expr, varList).

```
(%i14)   H : hessian (f, [x, y] );
```

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

The Hessian test first requires that the Hessian matrix be evaluated for x and y at a critical point (where the first derivatives vanish). Since for this example, the Hessian matrix is already purely a set of numbers, we can skip that step.

For any two variable expression or function f(x,y), n = 2, and the Hessian test is that we have a relative minimum if the two "leading principal minors" H1 and H2 are both positive.

We can use our Maxima function LPM (amatrix, num) to return these values, which need to be reduced to numerical values for our test.

We have a relative maximum if the first leading principal minor H1 < 0 and the second leading principal minor H2 > 0.

We call H1 the first leading principal minor; it is the minor gotten by taking the determinant of the matrix left after deleting all rows except the first row and deleting all columns except the first column, which leaves a matrix with only one element H[1,1] = fxx = 6. The determinant of such a one element matrix is simply the value of H[1,1] = 6.

```
(%i15) H1 : LPM (H, 1);
(H1)   6
```

We call H2 the second leading principal minor; the determinant of the matrix left after removing all but the first two rows and removing all but the first two columns - here we have the original matrix H which is a 2 x 2 matrix.

```
(%i16) H2 : LPM (H, 2);
(H2)   23
```

H1 and H2 are both positive, so we have a relative minimum. d^2f is a positive definite quadratic form at the given critical point

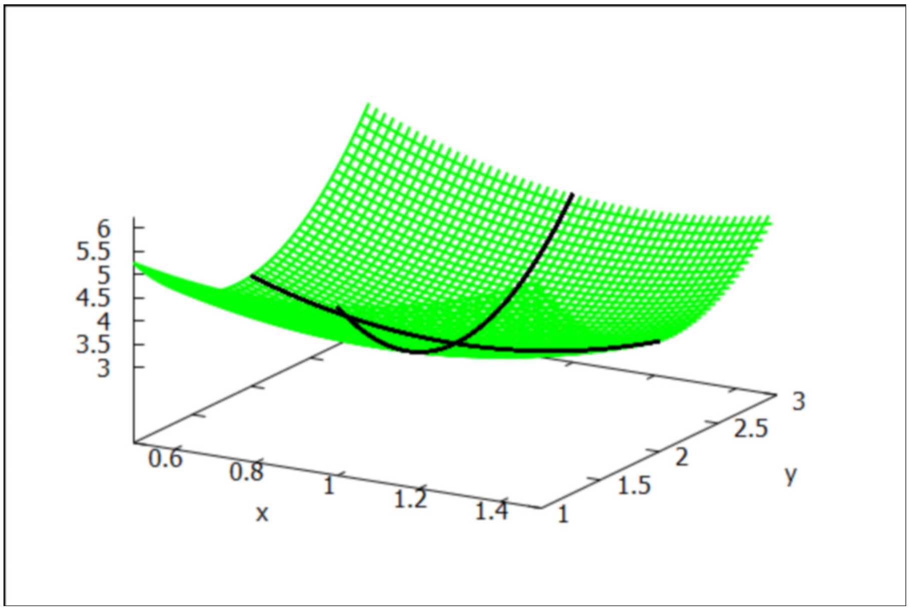
9.4.3 plotCP (expr, critPt)

The Maxima function plotCP (f, critPt) is defined in Econ1.mac, and is useful for the n = 2 case of unconstrained minima or unconstrained maxima.
This function shows a 3d plot of the surface f(x,y) near the given critical point.

(%i17) plotCP (f, cp1)$

surface of $2\,y^2 - x\,y - 7\,y + 3\,x^2 - 4\,x + 12$

near critical point = $[x=1, y=2]$

(%t17) 

1. Left click the plot once to select it; a border will appear.
2. Right click once to get a menu.
3. Left click once on "Popout interactively".
4. Expand the plot to full screen.
5. Rotate the plot using your cursor (holding left button down), to see surface behavior from different angles.
6. Close the gnuplot window to return to the wxMaxima worksheet.

9.4.4 CPtest (expr, critPts)

The case of the nature of the candidate extremum point (or list of points) of an expression or function is also evaluated by the function CPtest (expr, critPts), defined in Econ1.mac.

CPtest internally uses the Hessian matrix method, looking at the leading principal minors. If there are only two variables in the expression expr, CPtest looks at the signs of both zxx and zyy, and is able to distinguish an inflection point from a saddle point.

(%i18) CPtest (f, cp1)$

cp1 $[x=1, y=2]$ *, relative minimum, value =* 3.0

9.4.5 eigenvalues (numerical-Hessian-matrix)

We can also use the eigenvalues test for a minimum: if all eigenvalues are positive, we have a relative minimum

We use the Maxima function eigenvalues (numerical-Hessian-matrix), which, if successful, returns a list of two lists, the first being a list of eigenvalues found, the second being a list of the multiplicities of each of the eigenvalues found.

```
(%i19) [eivals, eimult] : eigenvalues (H);
```

(%o19) $[[5-\sqrt{2},\sqrt{2}+5],[1,1]]$

We are interested in the numerical values of the eigenvalues.

```
(%i20) float (eivals);
```

(%o20) $[3.5858,6.4142]$

or, you can use (in interactive work)

```
(%i21) eivals, numer;
```

(%o21) $[3.5858,6.4142]$

Since all eigenvalues of the numerical Hessian matrix (evaluated at the critical point) are positive, the Maxima expression f(x,y) has a relative minimum at the critical point.

9.4.6 Analyze (expr, critPt)

The function Analyze (expr, critpts) is limited to dealing with an expression which is a function of just two variables (as we have here):

```
(%i22) Analyze (f, cp1)$
```

1 *cp* = $[x=1,y=2]$ [*relative minimum*, *value* = 3.0]
secondDeriv = $[6,4,-1]$

9.5 Example 4: Extrema of f (x1, x2, x3) [12.4]

First the quick way with optima (f,vL).

(%i25) varList : [x1, x2, x3];
f : -5*x1^2 + 10*x1 + x1*x3 - 2*x2^2 + 4*x2 + 2*x2*x3 - 4*x3^2;
optimum (f, varList);

(varList) $[x1,x2,x3]$

(f) $-4\,x3^2+2\,x2\,x3+x1\,x3-2\,x2^2+4\,x2-5\,x1^2+10\,x1$

lagrangian = $-4\,x3^2+2\,x2\,x3+x1\,x3-2\,x2^2+4\,x2-5\,x1^2+10\,x1$

$cp1\ [x1=\frac{2^3\,3}{23},x2=\frac{2^2\,7}{23},x3=\frac{2\,5}{23}]$, *relative maximum, value* = 7.6522

(%o25) *done*

Next take the longer path, starting with the first derivatives of f.

(%i26) gradf : jacobian ([f], varList)[1];

(gradf) $[x3-10\,x1+10,2\,x3-4\,x2+4,-8\,x3+2\,x2+x1]$

(%i27) J : jacobian (gradf, varList);

(J) $\begin{pmatrix} -10 & 0 & 1 \\ 0 & -4 & 2 \\ 1 & 2 & -8 \end{pmatrix}$

(%i28) JD : determinant (J);

(JD) -276

Since the Jacobian determinant is NOT equal to zero, the three first derivative equations diff (f,xk) = 0, are functionally independent, and we expect to get a solution from solve.

(%i29) solns : solve (gradf, varList);

(solns) $[[x1=\frac{24}{23},x2=\frac{28}{23},x3=\frac{10}{23}]]$

(%i30) cp : solns [1];

(cp) $[x1=\frac{24}{23},x2=\frac{28}{23},x3=\frac{10}{23}]$

(%i31) cp, numer;

(%o31) $[x1=1.0435,x2=1.2174,x3=0.43478]$

Value of f at cp:

(%i32) float (at (f, cp));

(%o32) 7.6522

Hessian matrix

(%i33) H : hessian (f, varList);

(H) $\begin{pmatrix} -10 & 0 & 1 \\ 0 & -4 & 2 \\ 1 & 2 & -8 \end{pmatrix}$

The Hessian matrix H is purely numerical in this example, so we skip the normal step of evaluating H at the critical point cp found, and proceed to the next step of calculating the leading principal minors of H as it stands.

n = # of variables = 3, so we evaluate the three leading principal minors of the numerical matrix H.

(%i34) [LPM (H,1), LPM (H, 2), LPM (H, 3)];

(%o34) $[-10, 40, -276]$

The sign pattern of the three leading principal minors implies the Taylor series for d^2f, evaluated at the critical point, is a negative definite quadratic in (dx1, dx2, dx3) which implies a relative maximum at cp.

(%i35) CPtest (f, cp);

cp1 $[x1=\frac{24}{23}, x2=\frac{28}{23}, x3=\frac{10}{23}]$ *, relative maximum, value =* 7.6522

(%o35) *done*

We can make a 3d plot of two variables at a time near the critical point using plotCP (expr, critPt). One way to carry this out is to define a Maxima function F(x1, x2, x3) and call plotCP using this Maxima function with one of the arguments replaced by one of the three coordinates of the critical point.

(%i36) cpn : float (cp);

(cpn) $[x1=1.0435, x2=1.2174, x3=0.43478]$

(%i37) [x10, x20, x30] : map ('rhs, cpn);

(%o37) $[1.0435, 1.2174, 0.43478]$

We need to place two single quotes in front of the expression symbol f to force evaluation in this definition of the Maxima function F.

(%i38) F (x1, x2, x3) := ''f;

(%o38) $F(x1,x2,x3):=-4\,x3^2+2\,x2\,x3+x1\,x3-2\,x2^2+4\,x2-5\,x1^2+10\,x1$

Let cpn12 be a two variable critical point list for the variables (x1, x2).

(%i39) cpn12 : rest (cpn, -1);

(cpn12) $[x1=1.0435, x2=1.2174]$

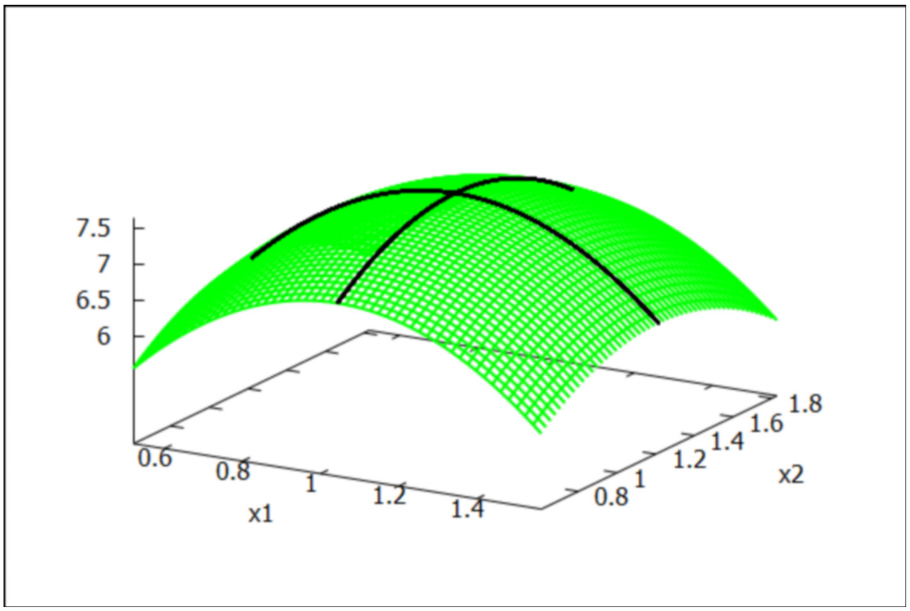
In the following call to plotCP, we replace x3 by the number x30.

(%i40) plotCP (F (x1, x2, x30), cpn12);

surface of $-2\,x2^2+4.8696\,x2-5\,x1^2+10.435\,x1-0.75614$

near critical point = $[x1=1.0435, x2=1.2174]$

(%t40)



(%o40)

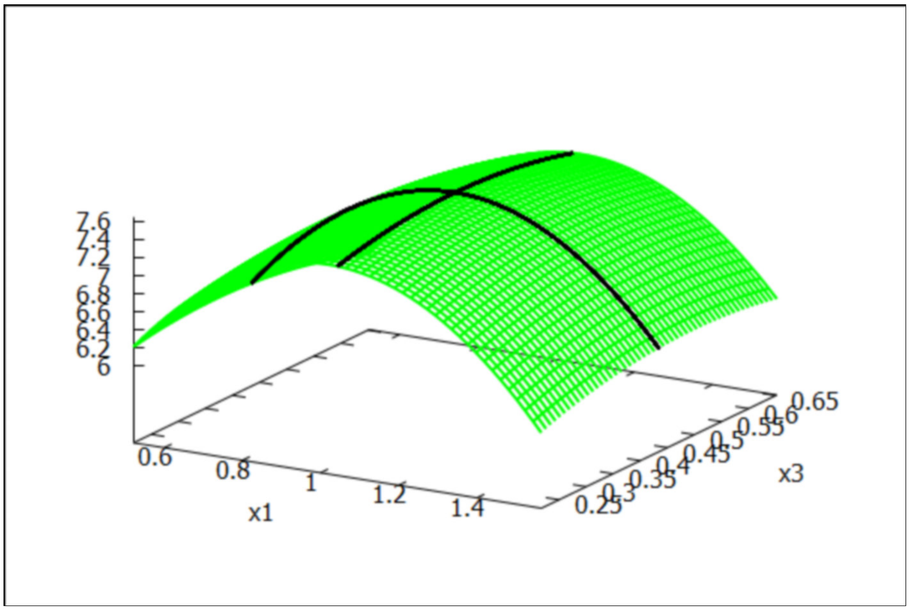
(%i41) cp13 : [cpn[1], cpn[3]];

(cp13) $[x1=1.0435, x3=0.43478]$

(%i42) plotCP (F (x1,x20, x3), cp13)$

surface of $-4\ x3^2 + x1\ x3 + 2.4348\ x3 - 5\ x1^2 + 10\ x1 + 1.9055$
near critical point = $[x1 = 1.0435, x3 = 0.43478]$

(%t42)



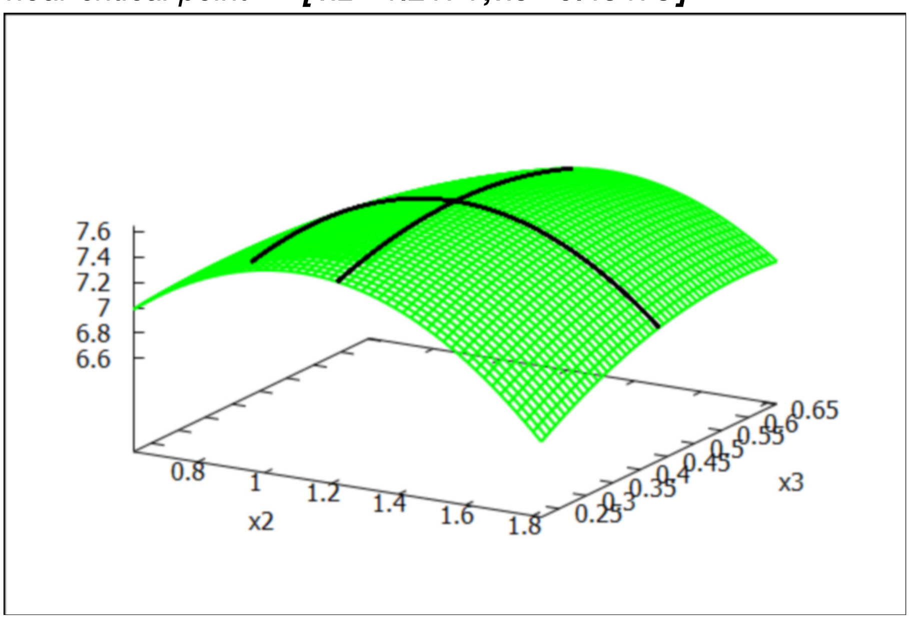
(%i43) cp23 : rest (cpn, 1);

(cp23) $[x2 = 1.2174, x3 = 0.43478]$

(%i44) plotCP (F (x10, x2, x3), cp23)$

surface of $-4\ x3^2 + 2\ x2\ x3 + 1.0435\ x3 - 2\ x2^2 + 4\ x2 + 4.9905$
near critical point = $[x2 = 1.2174, x3 = 0.43478]$

(%t44)



9.6 Problem 12.8 Extrema of f(x1, x2, x3)

(%i47) varList : [x1, x2, x3];
f : 3*x1^2 - 5*x1 - x1*x2 + 6*x2^2 - 4*x2 + 2*x2*x3 + 4*x3^2 + 2*x3 - 3*x1*x3;
optimum (f, varList);

(varList) $[x1,x2,x3]$

(f) $4\,x3^2+2\,x2\,x3-3\,x1\,x3+2\,x3+6\,x2^2-x1\,x2-4\,x2+3\,x1^2-5\,x1$

lagrangian = $4\,x3^2+2\,x2\,x3-3\,x1\,x3+2\,x3+6\,x2^2-x1\,x2-4\,x2+3\,x1^2-5\,x1$

(%o47) *no real and non−negative solutions �   returned from solve*

9.6.1 optimumAll (f, varL)

optimum is really designed for Economics problems for which the search variables are non-negative.
Use optimumAll (f, varList) to accept ALL solutions found by solve.

(%i48) optimumAll (f, varList);

lagrangian = $4\,x3^2+2\,x2\,x3-3\,x1\,x3+2\,x3+6\,x2^2-x1\,x2-4\,x2+3\,x1^2$ $-5\,x1$

cp1 $[x1=\frac{25}{28},x2=\frac{23}{56},x3=-\frac{1}{56}]$ *, relative minimum, value =* −
3.0714

(%o48) *done*

(%i49) float (cp);

(%o49) $[[x1=0.89286,x2=0.41071,x3=-0.017857]]$

9.6.2 Do it "by hand"...

Now for the long way:

(%i50) gradf : jacobian ([f], varList)[1];

(gradf) $[-3\,x3-x2+6\,x1-5,2\,x3+12\,x2-x1-4,8\,x3+2\,x2-3\,x1+2]$

(%i51) determinant (jacobian (gradf, varList));

(%o51) 448

The Jacobian determinant is not zero, so the three equations implied by the list gradf are functionally independent, and we expect solve to find a symbolic solution.

(%i52) solns : solve (gradf, varList);

(solns) $[[x1=\frac{25}{28},x2=\frac{23}{56},x3=-\frac{1}{56}]]$

(%i53) cp : solns[1];

(cp) $[x1=\frac{25}{28},x2=\frac{23}{56},x3=-\frac{1}{56}]$

(%i54) at (f, cp);

(%o54) $-\frac{43}{14}$

(%i55) float(%);

(%o55) −3.0714

(%i56) CPtest (f, cp);

cp1 $[x1=\frac{25}{28},x2=\frac{23}{56},x3=-\frac{1}{56}]$ *, relative minimum, value =* −
3.0714

(%o56) *done*

CPtest uses the leading principal minors of the Hessian matrix evaluated at the critical point, but we can check in detail:

(%i57) H : hessian (f, varList);

(H) $\begin{pmatrix} 6 & -1 & -3 \\ -1 & 12 & 2 \\ -3 & 2 & 8 \end{pmatrix}$

Since H is purely numerical, we skip the step of evaluating H at the critical point cp, and proceed with calculating the leading principal minors of this 3 x 3 matrix.

(%i58) [LPM (H,1), LPM (H, 2), LPM (H,3)];

(%o58) [6,71,448]

All three leading principal minors are positive, so the Taylor expansion d^2f about the critical point is a positive definite quadratic in (dx1, dx2, dx3) and the expression f has a relative minimum at that point.

As above, we call plotCP for 3d views of this minimum.

(%i59) [x10, x20, x30] : float (map ('rhs, cp));

(%o59) [0.89286,0.41071,−0.017857]

(%i60) F(x1, x2, x3) := ''f;

(%o60) $F(x1,x2,x3):=4\, x3^2+2\, x2\, x3-3\, x1\, x3+2\, x3+6\, x2^2-x1\, x2-4\, x2+3\, x1^2-5\, x1$

(%i61) grind(%)$

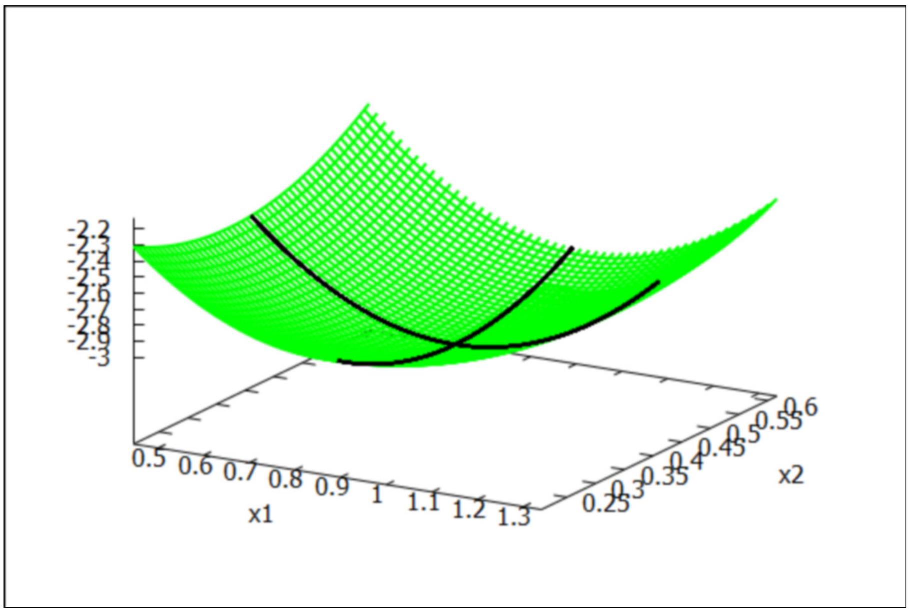
F(x1,x2,x3):=4*x3^2+2*x2*x3-3*x1*x3+2*x3+6*x2^2-x1*x2-4*x2+3*x1^2-5*x1$

(%i62) plotCP (F (x1, x2, x30), rest (cp, -1))$

surface of $6\, x2^2-x1\, x2-4.0357\, x2+3\, x1^2-4.9464\, x1-0.034439$

near critical point = $[x1=\frac{25}{28}, x2=\frac{23}{56}]$

(%t62)

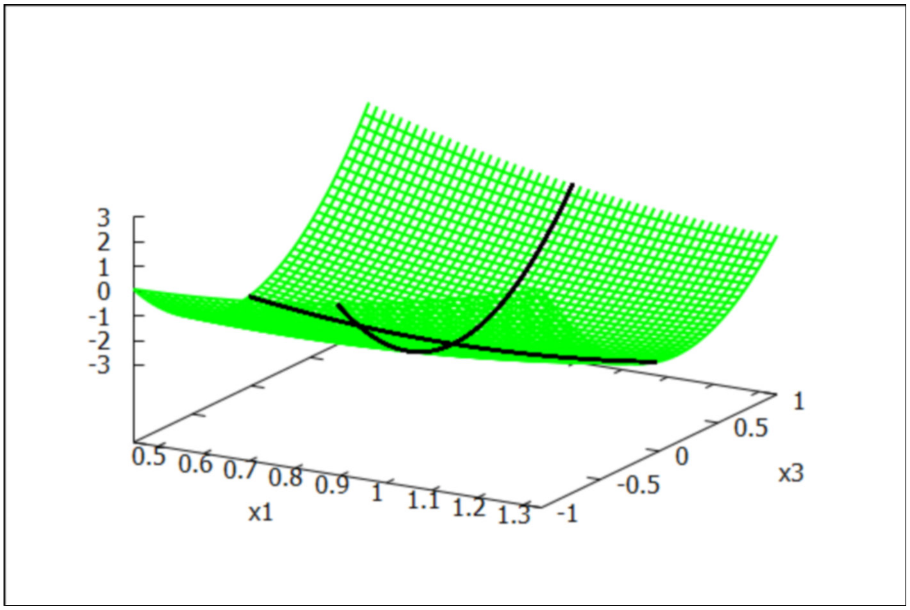


(%i63) plotCP (F (x1, x20, x3), [cp[1], cp[3]])$

surface of $4\,x3^2-3\,x1\,x3+2.8214\,x3+3\,x1^2-5.4107\,x1-0.63074$

near critical point = $[x1=\frac{25}{28}, x3=-\frac{1}{56}]$

(%t63)

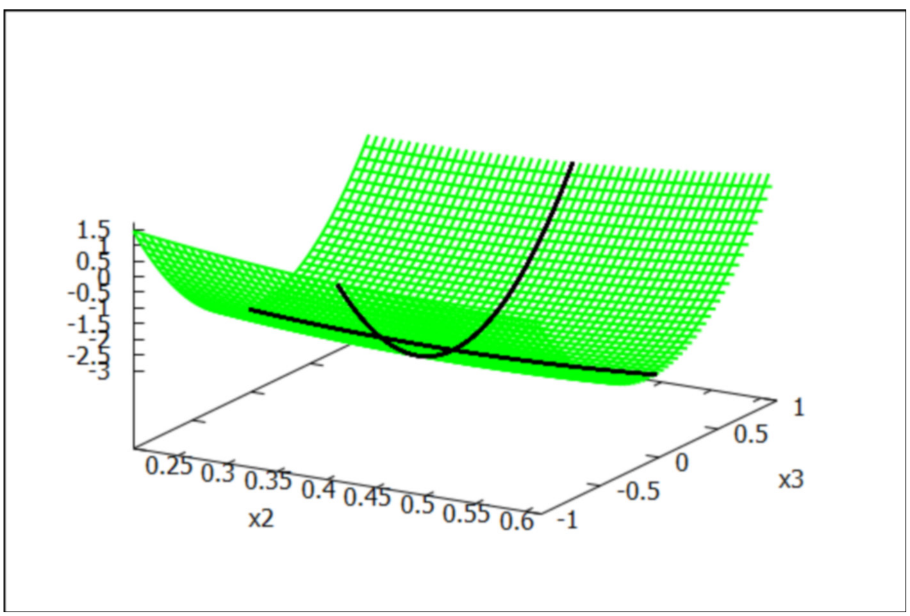


(%i64) plotCP (F (x10, x2, x3), [cp[2], cp[3]])$

surface of $4\,x3^2+2\,x2\,x3-0.67857\,x3+6\,x2^2-4.8929\,x2-2.0727$

near critical point = $[x2=\frac{23}{56}, x3=-\frac{1}{56}]$

(%t64)



10 Optimization Constrained by Equalities [12.5]

10.1 optimum (f, varL, g), optimum (f, varL, [g1, g2]), etc.

For one equality constraint g = 0, use optimum (f, varL, g).
For two equality constraints g1 = 0 and g2 = 0, use optimum (f, varL, [g1, g2]k).
And so on for more equality constraints.

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

Assume the objective function f is a function of n variables, and we have m constraints of form g(x1, x2, ...,xn) = 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 function :
L(x1, x2, ..., xn, λ1, λ2, ...,λm) = f(x1,x2,...,xn) + λ1 (c[1] - g[1] + λ2 (c[2] - g[2]) + ... + λm (c[m] - g[m]),
in which g[k] is the kth constraint function, depending in general on all n variables (x1, x2, ...,xn), and the kth constraint condition is g[k] = c[k],
gList is the list [g[1], g[2], ..., g[m]], and varList is [x1, x2, ..., xn].

The λk's are Lagrange multipliers, and initially are unknown constants.

If there is only one constraint function (m = 1), the syntax Bhessian(L,g,varList) is acceptable.

The Maxima function Bhessian is defined in Econ1.mac, and the above syntax produces a (m + n) x (m + n) square matrix which will involve the first derivatives of the m constraint functions g[k], and the second derivatives of the Lagrangian function L with respect to (wrt) the n variables in varList. In the most general case this bordered Hessian matrix will be a function of all n variables and all m lagrange multipliers λk.

To test for a maximum or minimum, BH should first be evaluated at a critical point, which is a solution of the m + n equations grad(L) = 0, where grad(L) is the list of the first derivatives of the Lagrangian L wrt x1, x2,..., xn, λ1, λ2,...,λm.

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

The function BHtest (BH, m, n) returns a list of the n - m leading principal minors of the bordered Hessian matrix BH, begining with LPM (BH, 2*m+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 sgn(x) = sign(x). For example, sgn(-2) = -1, sgn(6) = +1.

The number of relevant leading principal minors to check is n - m, in which
n = number of variables, m = number of equality constraints.
In ascending order, these are the determinants LPM (BH, 2*m+1) thru LPM (BH, m + n).
If n - m = 1, there is only one determinant to find (note that det BH = LPM (BH, m + n)), and in that case only condition a.) below is relevant.

A sufficient condition for a relative MAXIMUM is
a.) sgn (det BH) = sgn (LPM (BH, m + n)) = (-1)^n
b.) each successive LPM alternates in sign compared with the previous one.

A sufficient condition for a relative MINIMUM is:
a.) sgn (det BH) = sgn (LPM (BH, m + n)) = (-1)^m
b.) all LPM's have the same sign.

10.3 Chiang-Wainwright Ex. 1 (Sec 12.2, 12.3)

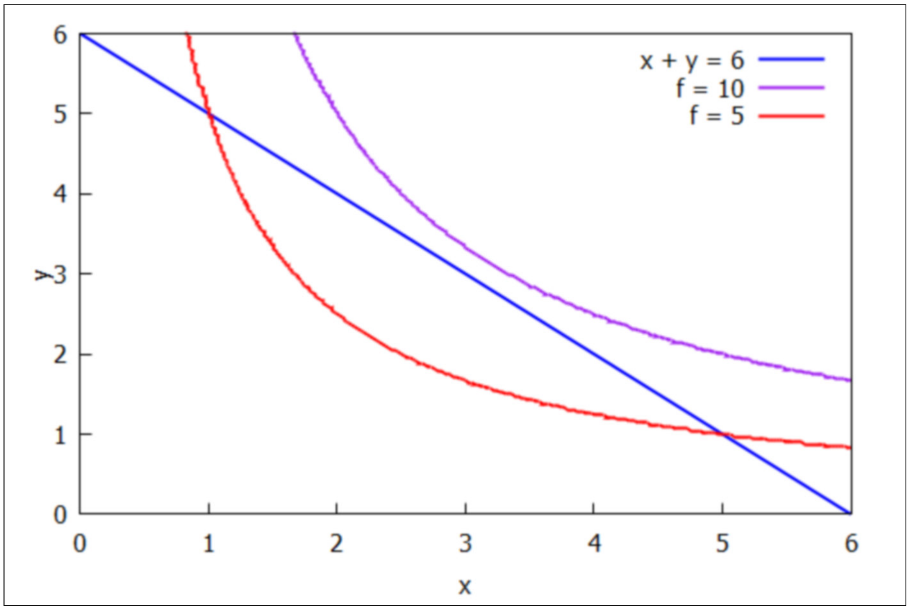
Find the extremum of f = x y subject to x + y = 6. For this example
m = 1 (equality constraint) and n = 2 (variables, x, y).
2 m + 1 = 3, m + n = 3, so there is only one LPM (BH,3) to look at.
(-1)^m = (-1)^1 = -1, so local minimum if LPM (BH,3) < 0.
(-1)^n = (-1)^2 = +1, so local maximum if LPM (BH,3) > 0.

```
(%i66) f : x*y;
       g : x + y;
(f)    x y
(g)    y+x
```

10.3.1 Graphical Exploration

(%i67)
```
wxdraw2d ( xlabel = "x", ylabel = "y",
   key = " x + y = 6", explicit (6 - x, x, 0, 6),
    color = purple, key = " f = 10", implicit ( f = 10, x, 0, 6,y,0,6),
    color = red, key = " f = 5", implicit ( f = 5, x, 0, 6,y,0,6))$
```

(%t67)



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

(%i68)
```
optimum (f, [x,y], 6 - g);
```

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

soln = $[x=3, y=3, lam_1=3]$ *objsub* = 9

soln = $[x=3.0, y=3.0, lam_1=3.0]$ *objsub* = 9.0

relative maximum

LPM's = $[LPM3=2.0]$

(%o68) *done*

10.3.3 Do it "by hand"...

Form the Lagrangian function
 L = f + λ (6 - x - y)
Look for the 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. We use the symbol lam to stand for the Lagrangian multiplier λ.

(%i71) L : f + lam*(6 -g);
gradL : jacobian ([L], [x, y, lam]) [1];
solns : solve (gradL, [x, y, lam]);

(L) $x\,y+lam\,(-y-x+6)$

(gradL) $[y-lam, x-lam, -y-x+6]$

(solns) $[[x=3, y=3, lam=3]]$

(%i72) soln : solns[1];

(soln) $[x=3, y=3, lam=3]$

(%i73) cp : rest (soln, -1);

(cp) $[x=3, y=3]$

(%i74) at (f, cp);

(%o74) 9

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

(%i75) BH : Bhessian (L, g, [x,y]);

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

For this example, Bhessian is already purely numerical, so we can skip the step of evaluating it at the critical values of x, y, and lam. The second derivatives of the Lagrangian function L (wrt x and y) appear in the 2x2 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, n - m = 2 - 1 = 1 is the number of leading principal minors of BH we need to check. 2 m + 1 = 3, m + n = 3, so the single case is LPM (BH, 3).

(%i76) LPM (BH, 3);

(%o76) 2

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.

(%i77) BHtest (BH, 1, 2);

relative maximum

(%o77) $[LPM3=2]$

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 x 3 matrix this submatrix is just BH. So we could have just calculated the determinant directly.

```
(%i78) determinant (BH);
(%o78) 2
```

Our criterion for a local maximum for this m = 1, n = 2 case is:
sgn (LPM (BH, n + m)) = sgn (det BH) = (-1)^n = +1 since n = 2.
Here sgn(x) refers to the sign of x.

Since LPM (BH, 3) = determinant(BH) = 2 > 0, sgn(2) = +1, we have a local maximum.

Dowling makes use of the language: "kth bordered leading principal minor"
BLPM (BH, m, k) where m = number of constraints, then k = m + 1 up to k = n.
For our case, m = 1, kmin = 2, kmax = n = 2, so there is just the determinant
BLPM (BH, 1, 2) --> LPM (BH, 3).

Let Hb stand for "H bar", and Hb2 for the second bordered leading principal minor.
Dowling uses the overbar symbol for BH, but in the absence of that symbol I will
use Hb. Then Dowling's notation is |Hbk| for the kth bordered leading principal minor.
BLPM is defined in Econ1.mac.

```
(%i79) Hb2 : BLPM (BH, 1, 2);
(Hb2) 2
```

10.4 Example 5 [Dowling Ch. 12]

Dowling p. 259
Two variables, one equality constraint.
Optimize the function f = 4 x^2 + 3 x y + 6 y^2 subject to the constraint x + y = 56.

```
(%i80) f : 4*x^2 + 3*x*y + 6*y^2;
```

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

10.4.1 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 = constant curves using implicit.

Define a Maxima function F(x,y), using two single quotes in front of f to force evaluation of the symbol f.

(%i81) F(x,y) := ''f;

(%o81) $F(x,y):=6\,y^2+3\,x\,y+4\,x^2$

(%i82) F(30,30);

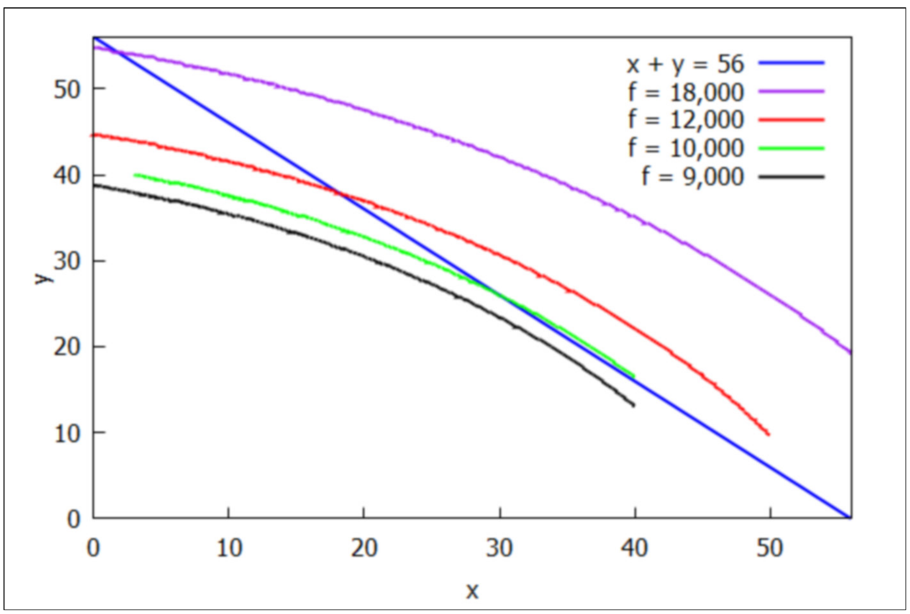
(%o82) 11700

(%i83) F (20,20);

(%o83) 5200

(%i84) 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))$

(%t84)



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

(%i86) g : x + y;
optimum (f, [x,y], 56 - g);

(g) $y+x$

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

$soln = [x=2^2\,3^2, y=2^2\,5, lam_1=2^2\,3\,29]$ $objsub = 9744$

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

relative minimum

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

(%o86) *done*

(%i87) F(36,20);

(%o87) 9744

10.4.3 Do it "by hand"....

(%i90) L : f + lam*(56 - g);
gradL : jacobian ([L], [x, y, lam])[1];
solns : solve (gradL, [x, y, lam]);

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

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

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

(%i91) soln : solns[1];

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

(%i92) 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 one of the critical points 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 ((n - m) = 1) leading principal minor to check, LPM (BH,3).

(%i93) BHtest (BH, 1, 2);

relative minimum

(%o93) $[LPM3=-14]$

(%i94) LPM (BH,3);

(%o94) −14

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

10.5 Prob. 12.28, f(x, y, z)

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

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

```
(%i99) f : 4*x*y*z^2;
       g : x + y + z;
       L : f + lam*(56 - g);
       gradL : jacobian ([L], [x, y, z, lam])[1];
       solns : solve (gradL, [x, y, z, lam]);
```

(f) $4\,x\,y\,z^2$

(g) $z+y+x$

(L) $4\,x\,y\,z^2+lam\,(-z-y-x+56)$

(gradL) $[4\,y\,z^2-lam,4\,x\,z^2-lam,8\,x\,y\,z-lam,-z-y-x+56]$

(solns) $[[x=\%r1,y=56-\%r1,z=0,lam=0],[x=14,y=14,z=28,lam=43904],[x=0,y=0,z=56,lam=0],[x=28,y=28,z=0,lam=0]]$

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

```
(%i100) soln : solns[2];
```

(soln) $[x=14,y=14,z=28,lam=43904]$

```
(%i101) 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 lam is zero at the critical point.

(%i102) subst (cp, [f, L]);

(%o102) [614656,614656]

Construct the bordered Hessian matrix using Bhessian

(%i103) BH : Bhessian(L, g, [x, y, z]);

(BH) $\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 4\,z^2 & 8\,y\,z \\ 1 & 4\,z^2 & 0 & 8\,x\,z \\ 1 & 8\,y\,z & 8\,x\,z & 8\,x\,y \end{pmatrix}$

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 lam does not appear in this matrix.

(%i104) 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.

(%i105) BHtest (BH, 1, 3);

relative maximum

(%o105) [*LPM3*=6272,*LPM4*=−19668992]

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

10.6 Example 6 [Dowling, Ch. 12] Cobb-Douglas Model

Extend the analysis of Example 10 in Dowling Ch. 6 [6.9] 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 K^α L^β

in which α > 0 is the "output elasticity of capital", β < 1 is the "output elasticity of labor", and A > 0 is an "efficiency parameter" measuring the level of technology employed. With PK the price per unit of capital K and PL the price per unit of labor L, the cost of production of q units is

cost = PK K + PL L.

Assume the budget constraint is $108 and PK = 3$/unit of cap., PL = 4$/unit of labor, and we want to optimize q = K^(0.4) L^(0.5) subject to the constraint that 3 K + 4 L = 108.

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.

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

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

```
(%i108) kill(L);
        g : 3*K + 4*L;
        q : ratsimp (K^(0.4)*L^(0.5));
```

(%o106) *done*

(g) $4\,L+3\,K$

(q) $K^{2/5}\sqrt{L}$

```
(%i109) optimum (q, [K,L], 108 - g);
```

lagrangian = $-4\,lam_1\,L+K^{2/5}\sqrt{L}-3\,lam_1\,K+108\,lam_1$

soln = $[K=2^4, L=3\,5, lam_1=\frac{8^{1/5}}{4\sqrt{15}}]$ *objsub* = $\sqrt{15}\,16^{2/5}$

soln = $[K=16.0, L=15.0, lam_1=0.097839]$ *objsub* = 11.741

relative maximum

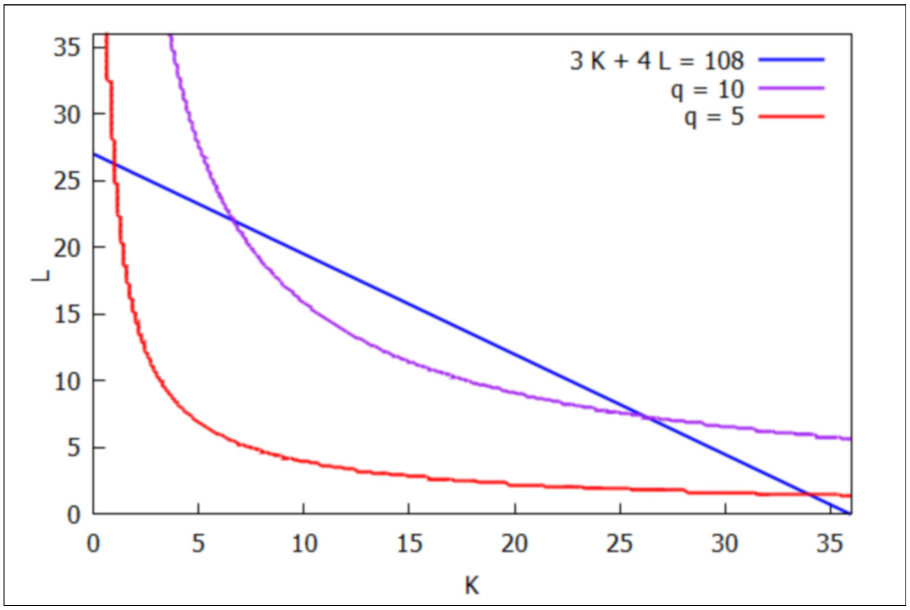
LPM's = $[LPM3=0.52833]$

(%o109) *done*

10.6.2 Graphical Exploration

(%i110) wxdraw2d (xlabel = "K", ylabel = "L",
key = " 3 K + 4 L = 108", explicit ((108 - 3*K)/4, K, 0, 36),
color = purple, key = " q = 10", implicit (q = 10, K, 0, 36, L, 0, 36),
color = red, key = " q = 5", implicit (q = 5, K, 0, 36, L, 0, 36))$

(%t110)



10.6.3 Do it "by hand"....

We include the budget constraint by defining the "Lagrange function"
Q = q + λ (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 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.

```
(%i113) Q : q + lam*(108 - g);
        gradQ : jacobian ([Q], [K,L,lam] )[1];
        solns : solve ( gradQ, [K, L, lam] );
```

(Q) $(-4\,L-3\,K+108)\,lam+K^{2/5}\sqrt{L}$

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

(solns) $\left[\left[K=16,L=15,lam=\frac{8^{1/5}}{4\sqrt{15}}\right]\right]$

```
(%i114) soln : solns[1];
```

(soln) $\left[K=16,L=15,lam=\frac{8^{1/5}}{4\sqrt{15}}\right]$

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

```
(%i115) 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.

```
(%i116) subst (cp, [q, Q] ), numer;
```

(%o116) $[11.741,11.741]$

```
(%i117) BH : Bhessian (Q, g, [K, L]);
```

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

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

```
(%i118) BH : at (BH, cp);
```

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

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

(%i119) BHtest (BH, 1, 2);

relative maximum

(%o119) $$\left[LPM3=4\left(\frac{3}{5\sqrt{15}\,16^{3/5}}+\frac{24\sqrt{15}}{25\,16^{8/5}}\right)-3\left(-\frac{3\,16^{2/5}}{4\,15^{3/2}}-\frac{4}{5\sqrt{15}\,16^{3/5}}\right)\right]$$

(%i120) float(%);

(%o120) $[LPM3=0.52833]$

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

(%i121) BH : float (BH);

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

(%i122) BHtest (BH, 1, 2);

relative maximum

(%o122) $[LPM3=0.52833]$

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

10.7 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 L^2
under the constraint
 g (K, L) = K + L = 3.

(%i124) Y : K*L^2;
g : K + L;

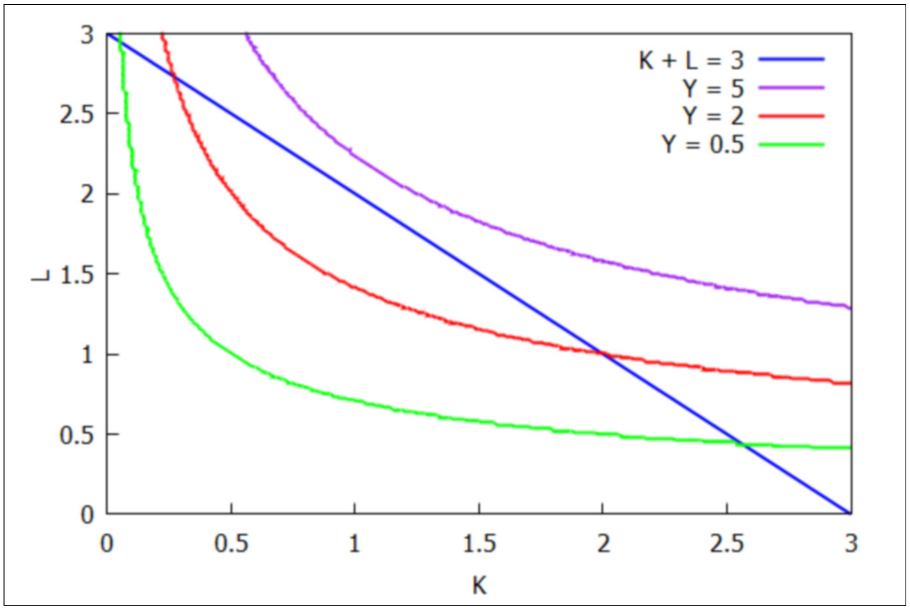
(Y) $K\,L^2$

(g) $L+K$

10.7.1 Graphical Exploration

(%i125) 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 = green, key = " Y = 0.5", implicit (Y = 0.5, K, 0, 3, L, 0, 3))$

(%t125)



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

(%i126) optimum (Y, [K, L], 3 - g);

lagrangian = $K L^2 - lam_1 L - lam_1 K + 3\, lam_1$

----

i = 1 *soln* = $[K=1, L=2, lam_1=2^2]$ *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 = $[LPM3=-6.0]$

(%o126) *done*

optimum has found a maximum at K = 1, L = 2, with Y = 4, 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.

10.7.3 Do it "by hand"...

Let F be the Lagrangian function and lam be the Lagrangian multiplier.

```
(%i129) F : Y + lam*(3 - g);
        gradF : jacobian ([F], [K,L,lam] )[1];
        solns : solve ( gradF, [K, L, lam] );
```

(F) $(-L-K+3)\, lam+K\, L^2$

(gradF) $[L^2-lam, 2\, K\, L-lam, -L-K+3]$

(solns) $[[K=1, L=2, lam=4], [K=3, L=0, lam=0]]$

```
(%i130) soln1 : solns[1];
```

(soln1) $[K=1, L=2, lam=4]$

```
(%i131) cp : rest (soln1,-1);
```

(cp) $[K=1, L=2]$

```
(%i132) at (Y, cp);
```

(%o132) 4

Construct the bordered Hessian matrix using the Lagrange function F.

```
(%i133) BH : Bhessian (F, g, [K,L] );
```

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

Evaluate the bordered Hessian matrix at the critical point cp of interest (Y = 4):

```
(%i134) BH : at (BH, cp);
```

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

```
(%i135) BHtest (BH, 1, 2);
```

relative maximum

(%o135) $[LPM3=6]$

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.

10.8 Francisco Feri Example, log transformation of objective func

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

https://intranet.royalholloway.ac.uk/economics/documents/
pdf/courseformsandinfo/ec5555lecture6(2013)-constrainedoptimsation.pdf

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

```
(%i137) f : x^3*y;
        g : x + y;
```

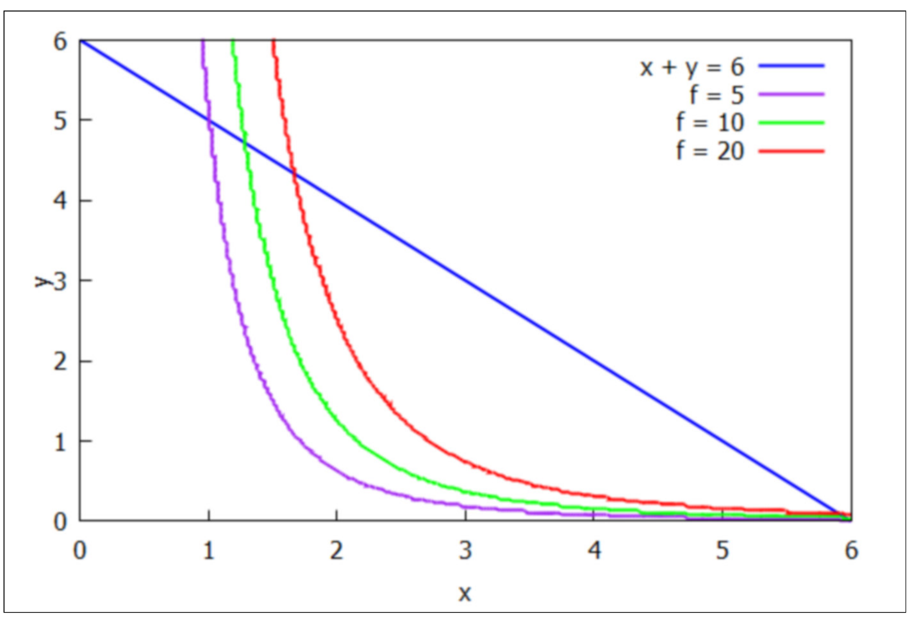
(f) $x^3\,y$

(g) $y+x$

10.8.1 Graphical Exploration

```
(%i138) wxdraw2d ( xlabel = "x", ylabel = "y",
          key = "x + y = 6", explicit (6 - x, x, 0, 6),
           color = purple, key = " f = 5", implicit ( f = 5, x, 0, 6, y, 0, 6),
           color = green, key = " f = 10", implicit ( f = 10, x, 0, 6, y, 0, 6),
           color = red, key = " f = 20", implicit ( f = 20, x, 0, 6, y, 0, 6))$
```

(%t138)



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

```
(%i139) optimum (f, [x, y], 6 - g);
```

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

----

i = 1 soln = $[x=\frac{3^2}{2}, y=\frac{3}{2}, lam_1=\frac{3^6}{2^3}]$ objsub = $\frac{2187}{16}$

soln = $[x=4.5, y=1.5, lam_1=91.125]$ objsub = 136.69

relative maximum

LPM's = $[LPM3=81.0]$

----

i = 2 soln = $[x=0, y=2\,3, lam_1=0]$ objsub = 0

soln = $[x=0.0, y=6.0, lam_1=0.0]$ objsub = 0.0

indefinite

LPM's = $[LPM3=0.0]$

(%o139) *done*

10.8.3 Maxima's log function: mini-tutorial, logexpand

Feri uses a log transformation of an objective function to simplify the analysis. Maxima's log function stands for the natural logarithm ln.

ln(f) is a one-to-one monotonic strictly increasing transformation, so if we find a critical point cp of ln(f), then cp is also a critical point of f.

Maxima pays attention to the setting of a flag called logexpand, which is set to true (by default). Here we interrogate the current setting of logexpand, and show the behavior of Maxima's log function with that setting. We then change the setting to "all".

```
(%i1) killAB()$
      logexpand;
```

(%o1) ***true***

```
(%i2) log (exp(A));
```

(%o2) A

```
(%i3) exp (log (A));
```

(%o3) A

```
(%i4) log(x^3);
```

(%o4) $3 \log(x)$

(%i5) log (a*b);

(%o5) log(a b)

(%i6) log(x^3*y);

(%o6) log(x^3 y)

If you change the default setting, using logexpand : all, you get different behavior.

(%i7) logexpand : all;

(logexpand) *all*

(%i8) log(x^3);

(%o8) 3 log(x)

(%i9) log (a*b);

(%o9) log(b)+log(a)

(%i10) log (x^3*y);

(%o10) log(y)+3 log(x)

Here we return the setting to the default:

(%i11) logexpand : true;

(logexpand) ***true***

Use L for the Lagrange function symbol.

```
(%i15) kill(L)$
       L : f + lam* (6 - g);
       gradL : jacobian ([L], [x, y, lam])[1];
       solns : solve (gradL, [x, y, lam]);
```

(L) (6−g) lam+f

(gradL) [0,0,6−g]

solve: dependent equations eliminated: (1 2)

(solns) []

Redefine the objective function as 3 ln(x) + ln(y), which is the natural log of our original objective function x^3*y. Let f1 stand for ln(f), and L1 stand for the new Lagrange function, and lam1 the new Lagrange multiplier.

```
(%i20) f1 : 3*log (x) + log (y);
       g : x + y;
       L1 : f1 + lam1* (6 - g);
       gradL1 : jacobian ([L1], [x, y, lam1])[1];
       solns_ln : solve (gradL1, [x, y, lam1 ]);
```

(f1) $\log(y)+3\log(x)$

(g) $y+x$

(L1) $\log(y)+lam1\,(-y-x+6)+3\log(x)$

(gradL1) $[\frac{3}{x}-lam1,\frac{1}{y}-lam1,-y-x+6]$

(solns_ln) $[[x=\frac{9}{2},y=\frac{3}{2},lam1=\frac{2}{3}]]$

Looking for the location of the maximum of ln(f) has found the same point (x = 9/2, y = 3/2) as we found above.

Using optimum, note the difference between using log(f) as the objective function vs. using f1 (as defined above).

```
(%i21) optimum(log(f), [x,y], 6 - g);
```

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

solve: dependent equations eliminated: (2)

soln = $[x=\%r2,y=-(\%r2-6),lam_1=0]$ *objsub* = $\log(f)$

soln = $[x=\%r2,y=-1.0\,(\%r2-6.0),lam_1=0.0]$ *objsub* = $\log(f)$

indefinite

LPM's = $[LPM3=0.0]$

(%o21) *done*

```
(%i22) optimum(f1, [x,y], 6 - g);
```

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

soln = $[x=\frac{3^2}{2},y=\frac{3}{2},lam_1=\frac{2}{3}]$ *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 = $[LPM3=0.59259]$

(%o22) *done*

We will get the same behavior using log(f) IF we set logexpand to all first.

(%i24) logexpand : all;
optimum(log(f), [x,y], 6 - g);

(logexpand) *all*

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

solve: dependent equations eliminated: (2)

$soln = [x = \%r3, y = -(\%r3 - 6), lam_1 = 0] \quad objsub = \log(f)$

$soln = [x = \%r3, y = -1.0\,(\%r3 - 6.0), lam_1 = 0.0] \quad objsub = \log(f)$

indefinite

$LPM's = [LPM3 = 0.0]$

(%o24) *done*

Return the flag logexpand to its default setting (true).

(%i25) logexpand : true;

(logexpand) ***true***

At the critical point ln(f) = ln(2187/16) ~ 4.9177

(%i26) log (2187/16), numer;

(%o26) 4.9177

optimum returns the global list cp.

(%i27) cp;

(%o27) $[[x = \%r3, y = -(\%r3 - 6), lam_1 = 0]]$

(%i28) soln : cp[1];

(soln) $[x = \%r3, y = -(\%r3 - 6), lam_1 = 0]$

(%i29) cp1 : rest (soln, -1);

(cp1) $[x = \%r3, y = -(\%r3 - 6)]$

(%i30) at (f, cp1);

(%o30) f

(%i31) float(%);

(%o31) f

This is a m = 1 constraint, n = 2 variables example. There are (n - m) = 1 leading principal minors of the bordered Hessian matrix BH to examine, LPM (BH, m+n) = LPM (BH,3).

(%i32) BH : Bhessian (L1, g, [x,y]);

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

(%i33) BH : at (BH, cp1);

(BH) $\begin{pmatrix} 0 & 1 & 1 \\ 1 & -\frac{3}{\%r3^2} & 0 \\ 1 & 0 & -\frac{1}{(6-\%r3)^2} \end{pmatrix}$

(%i35) LPM (BH,3);
(-1)^2;

(%o34) $\frac{3}{\%r3^2}+\frac{1}{(6-\%r3)^2}$

(%o35) 1

Since LPM (BH, 3) > 0 and n = 2, cp is a relative maximum.

(%i36) BHtest (BH, 1, 2);

relative maximum

(%o36) $[LPM3=\frac{3}{\%r3^2}+\frac{1}{(6-\%r3)^2}]$

Our original objective function's value at the critical point cp is:

(%i37) at (x^3*y, cp1), numer;

(%o37) $(6-\%r3)\ \%r3^3$

To have a global maximum we need the lagrangian L1 to be concave. Construct the Hessian matrix of L1

(%i38) H : hessian (L1, [x, y, lam1]);

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

(%i39) matrix (["H1", "H2", "H3"], [H1 : LPM (H,1), H2 : LPM (H, 2), H3 : LPM (H, 3)]);

(%o39)
$$\begin{pmatrix} H1 & H2 & H3 \\ -\frac{3}{x^2} & \frac{3}{x^2 y^2} & \frac{1}{y^2}+\frac{3}{x^2} \end{pmatrix}$$

For all x > 0, y > 0, H1 < 0, H2 > 0, H3 > 0. The pattern starts out looking like a negative definite Hessian matrix, indicating a global maximum, and hence L1 being strictly concave and implying a global maximum, but H3 > 0 spoils the expected pattern.

Francisco asserts that because the constraint is linear, we instead need to look at the Hessian matrix of the objective function only.

(%i41) f1;
Hf1 : hessian (f1, [x, y]);

(%o40) $\log(y)+3\log(x)$

(Hf1)
$$\begin{pmatrix} -\frac{3}{x^2} & 0 \\ 0 & -\frac{1}{y^2} \end{pmatrix}$$

(%i42) LPM (Hf1,1);

(%o42) $-\frac{3}{x^2}$

which is < 0 for all x > 0.

(%i43) LPM (Hf1, 2);

(%o43) $\frac{3}{x^2 y^2}$

which is > 0 for all x > 0 and all y > 0. Thus the Hessian Hf1 is negative definite (N.D.) and the objective function is strictly concave and cp1 is a global maximum.

10.9 Chiang-Wainwright p. 360 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 (x1, x2) = x1 x2,
where x1 > 0 is consumption in period 1 and x2 > 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 x1 and the present value of x2 add up to B. Thus
x1 + x2/(1+r) = B.
Find the values of x1 and x2 which maximize the utility subject to this constraint.

(%i4)
```
killAB()$
g : x1 + x2/(1+r);
U : x1*x2;
L : U + lam*(B - g);
gradL : jacobian ([L], [x1, x2, lam])[1];
```

(g) $\frac{x2}{r+1}+x1$

(U) $x1\ x2$

(L) $lam\left(-\frac{x2}{r+1}-x1+B\right)+x1\ x2$

(gradL) $[x2-lam, x1-\frac{lam}{r+1}, -\frac{x2}{r+1}-x1+B]$

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

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

(%i5) optimum(U, [x1, x2], B - g);

$lagrangian = \dfrac{((r+1)\,x1-lam_1)\,x2+(-lam_1\,r-lam_1)\,x1+lam_1\,B\,r+lam_1\,B}{r+1}$

$soln = [x1=\dfrac{B}{2}, x2=\dfrac{B\,(r+1)}{2}, lam_1=\dfrac{B\,(r+1)}{2}]$ $objsub = \dfrac{B^2\,(r+1)}{4}$

$soln = [x1=0.5\,B, x2=0.5\,B\,(r+1.0), lam_1=0.5\,B\,(r+1.0)]$

$objsub = 0.25\,B^2\,(r+1.0)$

indefinite

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

(%o5) *done*

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

(%i7) assume (r > 0, B > 0);
optimum(U, [x1, x2], B - g);

(%o6) $[r>0, B>0]$

$lagrangian = \dfrac{((r+1)\,x1-lam_1)\,x2+(-lam_1\,r-lam_1)\,x1+lam_1\,B\,r+lam_1\,B}{r+1}$

$soln = [x1=\dfrac{B}{2}, x2=\dfrac{B\,(r+1)}{2}, lam_1=\dfrac{B\,(r+1)}{2}]$ $objsub = \dfrac{B^2\,(r+1)}{4}$

$soln = [x1=0.5\,B, x2=0.5\,B\,(r+1.0), lam_1=0.5\,B\,(r+1.0)]$ $objsub = 0.25$
$B^2\,(r+1.0)$

relative maximum

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

(%o7) *done*

(%i8) solns : solve (gradL, [x1, x2, lam]), factor;

(solns) $[[x1=\dfrac{B}{2}, x2=\dfrac{B\,(r+1)}{2}, lam=\dfrac{B\,(r+1)}{2}]]$

(%i9) soln : solns[1];

(soln) $[x1=\dfrac{B}{2}, x2=\dfrac{B\,(r+1)}{2}, lam=\dfrac{B\,(r+1)}{2}]$

(%i10) cp : rest (soln, -1);

(cp) $[x1=\frac{B}{2}, x2=\frac{B\,(r+1)}{2}]$

(%i11) subst (cp, [U, L]);

(%o11) $[\frac{B^2\,(r+1)}{4}, \frac{B^2\,(r+1)}{4}]$

(%i12) 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, n - m = 1, and we only need to check the sign of the determinant of BH.

(%i13) determinant (BH);

(%o13) $\frac{2}{r+1}$

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

(%i14) BHtest (BH, 1, 2);

relative maximum

(%o14) $[LPM3=\frac{2}{r+1}]$

10.10 Two Equality Constraints, Nonlinear Three Variables Example

Maximize f = x y z subject to x^2 + y^2 = 4 and x + z = 2.

```
(%i17) f : x*y*z;
       g1 : x^2 + y^2;
       g2 : x + z;
```

(f) $x\,y\,z$

(g1) y^2+x^2

(g2) $z+x$

First we optimistically try optimum (...)

```
(%i18) optimum (f, [x, y, z], [4-g1, 2 - g2]);
```

$lagrangian = (x\,y-lam_2)\,z-lam_1\,y^2-lam_1\,x^2-lam_2\,x+2\,lam_2+4\,lam_1$

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

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

indefinite

$LPM's = [LPM5=0.0]$

(%o18) *done*

One solution x = 2, y = 0, z = 0, f = 0 was found, which satisfies the constraints.
Evidently solve has trouble with this problem.

```
(%i21) L : f + lam1*(4 - g1) + lam2*(2 - g2);
       gradL : jacobian ([L], [x, y, z, lam1, lam2] )[1];
       solns : solve (gradL, [x, y, z, lam1, lam2] );
```

(L) $x\,y\,z+lam2\,(-z-x+2)+lam1\,(-y^2-x^2+4)$

(gradL) $[y\,z-2\,lam1\,x-lam2, x\,z-2\,lam1\,y, x\,y-lam2, -y^2-x^2+4, -z-x+2]$

(solns) $[[x=2, y=0, z=0, lam1=0, lam2=0]]$

The system of equations to solve are nonlinear, so problem is more difficult. We need to reduce the number of variables and equations to achieve a solution.
Gives the five first derivatives names e1, e2, ...

```
(%i22) [e1,e2,e3,e4,e5] : gradL;
```

(%o22) $[y\,z-2\,lam1\,x-lam2, x\,z-2\,lam1\,y, x\,y-lam2, -y^2-x^2+4, -z-x+2]$

we can simplify by solving e2 and e3 for lam1 and lam2

```
(%i24) lam1s : solve (e2,lam1);
       lam2s : solve (e3, lam2);
```

(lam1s) $[lam1=\frac{x\,z}{2\,y}]$

(lam2s) $[lam2=x\,y]$

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

```
(%i25) lams : [ lam1s[1], lam2s[1] ];
```

(lams) $[lam1=\frac{x\,z}{2\,y}, lam2=x\,y]$

Now substitute for lam1 and lam2 in e1, e4, and e5, preserving the names.

(%i26) [e1, e4, e5] : subst (lams, [e1, e4, e5]);

(%o26) $[y\,z-\frac{x^2\,z}{y}-x\,y,-y^2-x^2+4,-z-x+2]$

Because the equations to solve are now e1 = 0, e4 = 0, e5 = 0, we can multiply e1 by y without changing the set of equations.

(%i27) e1 : y*e1, expand;

(e1) $y^2\,z-x^2\,z-x\,y^2$

10.10.1 Using eliminate (alist, qL)

Now eliminate y^2 and z, using these three equations, to get an equation in x.

(%i28) [ex] : eliminate ([e1,e4, e5], [y^2, z]);

(%o28) $[-3\,x^3+4\,x^2+8\,x-8]$

solve usually can deal with a cubic equation.

(%i29) 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.

(%i30) [x1s, x2s, x3s] : xsolns;

(%o30) $[x=-\frac{\sqrt{13}+1}{3},x=\frac{\sqrt{13}-1}{3},x=2]$

Let's first deal with the x = 2 case, and solve for z and y for this case.
We see from the new version of e5 that in this case z = 0.
at (e5, x3s) replaces x by 2 in equation e5 in the first
argument to solve here, where we ask for the value of z implied by the equation
at (e5, x3s) = 0.

(%i31) [z3s] : solve (at (e5, x3s), z);

(%o31) $[z=0]$

We see from e4 that when x = 2, y = 0 also.

(%i32) [y3s] : solve (at (e4, x3s), y);

(%o32) $[y=0]$

For x = 2 the objective function is equal to zero.

(%i33) cp3 : [x3s, y3s, z3s];

(cp3) $[x=2,y=0,z=0]$

(%i34) at (f, cp3);

(%o34) 0

So the candidate critical point (2,0,0) has f(2,0,0) = 0.

Next look at the case x2s.

(%i35) x2s;

(%o35) $x=\frac{\sqrt{13}-1}{3}$

(%i36) [z2s] : solve (at (e5, x2s), z);

(%o36) $[z=-\frac{\sqrt{13}-7}{3}]$

(%i37) solve (at (e4, x2s), y);

(%o37) $[y=-\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},y=\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3}]$

Since we have two solutions for y for this value of x, let's call them y2sm (for minus) and y2sp (for plus).

(%i38) [y2sm, y2sp] : %;

(%o38) $[y=-\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},y=\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3}]$

(%i39) cp2m : [x2s, y2sm, z2s];

(cp2m) $[x=\frac{\sqrt{13}-1}{3},y=-\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},z=-\frac{\sqrt{13}-7}{3}]$

(%i40) cp2p : [x2s, y2sp, z2s];

(cp2p) $[x=\frac{\sqrt{13}-1}{3},y=\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},z=-\frac{\sqrt{13}-7}{3}]$

Finally, the case x1s.

(%i41) x1s;

(%o41) $x=-\frac{\sqrt{13}+1}{3}$

(%i42) [z1s] : solve (at (e5, x1s), z);

(%o42) $[z=\frac{\sqrt{13}+7}{3}]$

(%i43) solve (at (e4, x1s), y);

(%o43) $[y=-\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},y=\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3}]$

(%i44) [y1sm, y1sp] : %;

(%o44) $[y=-\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},y=\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3}]$

(%i45) cp1m : [x1s, y1sm, z1s];

(cp1m) $[x=-\frac{\sqrt{13}+1}{3},y=-\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},z=\frac{\sqrt{13}+7}{3}]$

(%i46) cp1p : [x1s, y1sp, z1s];

(cp1p) $[x=-\frac{\sqrt{13}+1}{3},y=\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},z=\frac{\sqrt{13}+7}{3}]$

All together we have five candidate critical points.

(%i48) critPts : [cp3,cp2m,cp2p,cp1m,cp1p];
float (critPts);

(critPts) $[[x=2,y=0,z=0],[x=\frac{\sqrt{13}-1}{3},y=-\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},z=-\frac{\sqrt{13}-7}{3}],[x=\frac{\sqrt{13}-1}{3},y=\frac{\sqrt{2}\sqrt{\sqrt{13}+11}}{3},z=-\frac{\sqrt{13}-7}{3}],[x=-\frac{\sqrt{13}+1}{3},y=-\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},z=\frac{\sqrt{13}+7}{3}],[x=-\frac{\sqrt{13}+1}{3},y=\frac{\sqrt{2}\sqrt{11-\sqrt{13}}}{3},z=\frac{\sqrt{13}+7}{3}]]$

(%o48) $[[x=2.0,y=0.0,z=0.0],[x=0.86852,y=-1.8016,z=1.1315],[x=0.86852,y=1.8016,z=1.1315],[x=-1.5352,y=-1.2819,z=3.5352],[x=-1.5352,y=1.2819,z=3.5352]]$

(%i49) makelist (at (f, cpn),cpn,critPts), numer;

(%o49) [0,−1.7704,1.7704,6.957,−6.957]

(%i50) BH : Bhessian (L, [g1, g2], [x, y, z]);

$$\text{(BH)} \quad \begin{pmatrix} 0 & 0 & 2\,x & 2\,y & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 2\,x & 1 & -2\,lam1 & z & y \\ 2\,y & 0 & z & -2\,lam1 & x \\ 0 & 1 & y & x & 0 \end{pmatrix}$$

Use the list lams to get rid of lam1 in BH.

(%i51) lams;

(%o51) $[lam1=\frac{x\,z}{2\,y}, lam2=x\,y]$

(%i52) BH : at (BH, lams);

$$\text{(BH)} \quad \begin{pmatrix} 0 & 0 & 2\,x & 2\,y & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 2\,x & 1 & -\frac{x\,z}{y} & z & y \\ 2\,y & 0 & z & -\frac{x\,z}{y} & x \\ 0 & 1 & y & x & 0 \end{pmatrix}$$

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.

The objective function f = x y z has its largest value at cp1m (6.96). Check the SOC using BHtest (BH,m n), or directly using LPM (BH,k) for the relevant leading principal minors of BH evaluated at the critical point cp1m. (We are assuming we want f to be both positive and as large as possible.)

(%i53) BHcp1m : float (at (BH, cp1m));

(BHcp1m)
$$\begin{pmatrix} 0.0 & 0.0 & -3.0704 & -2.5638 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 1.0 \\ -3.0704 & 1.0 & -4.2338 & 3.5352 & -1.2819 \\ -2.5638 & 0.0 & 3.5352 & -4.2338 & -1.5352 \\ 0.0 & 1.0 & -1.2819 & -1.5352 & 0.0 \end{pmatrix}$$

(%i54) BHtest (BHcp1m, 2, 3);

relative maximum

(%o54) $[LPM5 = -130.71]$

With m = 2, n = 3, (-1)^n = (-1)^3 = -1, n - m = 1, there is only one LPM to check, 2 m + 1 = 5, m + n = 5, LPM (BH, 5), which has the same sign as (-1)^n, so a local maximum.

11 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 jth commodity, the need for the ith commodity must be a fixed amount, which we denote by a[i,j]. Specifically, the production of each unit of the jth 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 nth 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 ith commodity is used for the production of each unit of the jth 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 x 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:

(%i55) 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.

11.1 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,...,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 at 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] + ... + 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) . X = B, or
M . 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) = invert (M), and obtain the unique solution of the n equations using Xs : invert(M) . B

For our n = 4 example we first construct lists xL and bL and turn each into a matrix column vector using X : cvec (xL), for example, where cvec is defined in Econ1.mac.

(%i9)
```
killAB()$
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 : diagmatrix (4,1) - 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) $$\begin{pmatrix} 1-a_{1,1} & -a_{1,2} & -a_{1,3} & -a_{1,4} \\ -a_{2,1} & 1-a_{2,2} & -a_{2,3} & -a_{2,4} \\ -a_{3,1} & -a_{3,2} & 1-a_{3,3} & -a_{3,4} \\ -a_{4,1} & -a_{4,2} & -a_{4,3} & 1-a_{4,4} \end{pmatrix}$$

(%i10) 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
invert (M) . 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 : invert (M) . B
(again, don't try to do this symbolically now). See the numerical example below.

11.1.1 A Numerical Example

Chiang and Wainwright, p. 115, have the following numerical example for a three industry economy.

(%i1) killAB()$
A : matrix ([0.2, 0.3, 0.2], [0.4, 0.1, 0.2], [0.1, 0.3, 0.2]);

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

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

(%i2) Leontief (A);

(%o2)
$$\begin{pmatrix} 0.8 & -0.3 & -0.2 \\ -0.4 & 0.9 & -0.2 \\ -0.1 & -0.3 & 0.8 \end{pmatrix}$$

(%i3) length (A);

(%o3) 3

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

```
(%i4) M : diagmatrix (3,1) - A;
```

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

```
(%i5) rank (M);
(%o5) 3
```

The rank (see the section above near the beginning) of M is 3, which is equal to the dimension of M, so M is nonsingular, and the matrix inverse of M exists.

```
(%i6) 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)

```
(%i7) B : cvec ([10, 5, 6] );
```

$$B = \begin{pmatrix} 10 \\ 5 \\ 6 \end{pmatrix}$$

The software file Econ1.mac defines the Maxima function
InputOutput (amatrix, finalDemand)
which checks some properties of amatrix and returns the total demand matrix column vector solution.

```
(%i8) InputOutput (M, B);
```

$$(\%o8) \quad \begin{pmatrix} 24.844 \\ 20.677 \\ 18.359 \end{pmatrix}$$

```
(%i9) length(B);
(%o9) 3
```

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

(%i10) Xs : Minv . B;

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

(%i13) Xs[1];
Xs[1,1];
Xs[2,1];

(%o11) [24.844]

(%o12) 24.844

(%o13) 20.677

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

Let a0(j) be the dollar amount of the primary input (like labor, capital, etc.) used in producing one dollar's worth of the jth commodity, which can be calculated by subtracting the sum of the jth column of the input matrix A from 1.
In general, A[k, j] returns the A matrix element in the kth row and the jth column.

Here we define a Maxima function depending of j:

(%i14) a0(j) := 1 - sum (A[k, j], k, 1, 3)$

then we have

(%i15) [a0(1), a0(2), a0(3)];

(%o15) [0.3,0.3,0.4]

The Maxima function Pinput (techcoeff, num) is defined in Econ1.mac, and does the same job as a0(j), plus figures out what the final k value should be in the sum.

(%i16) Pinput (A,1);

(%o16) 0.3

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.

```
(%i19) Xs[1];
       a0(1);
       a0(1) * Xs[1,1];
(%o17) [24.844]
(%o18) 0.3
(%o19) 7.4531
```

The required primary input (PI) is:

```
(%i20) PI : sum (a0(k)*Xs[k,1], k, 1, 3);
(PI)   21.0
```

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

```
(%i21) PItot (A, Xs);
(%o21) 21.0
```

so the specific demand (in billions) assumed in our definition of the consumer final demand matrix column vector B willl 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.

11.1.2 Problem 12.33 & 12.34

Given the interindustry transaction demand table in millions of dollars below, find the matrix of technical coefficients A. The (vertical) row "origin" categories of industries refer to inputs provided to "output" industry categories as columns (see our discussion below). "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 this (in this table in his book) as "Value Added".

(%i2)
```
killAB()$
print ("                    Sector of Destination " )$
Mtable : matrix (["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,     " ",              " "  ] );
```

Sector of Destination

(Mtable)

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 producer.
To produce $1,000 M 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 production of autos, and $370M worth of "primary inputs" (labor, capital, etc)
to produce $1,000M worth of autos.

If we look at the "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).

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

(%i3) 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.

(%i4) a[2,3];

(%o4) $a_{2,3}$

(%i5) f(i,j) := float (Mtable[i+1, j+1] / Mtable[7,j+1])$

(%i6) f(1,1);

(%o6) 0.13333

```
(%i7) for i thru 4 do
          for j thru 4 do  a[i,j] : f(i,j)$
```

(%i8) a[1,1];

(%o8) 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.

(%i9) 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 a0(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):

(%i10) a0(j) := 1 - sum (A[k, j], k, 1, 4)$

(%i11) matrix (["steel", "coal", "iron", "Auto"],
[a0(1), a0(2), a0(3), a0(4)]);

(%o11)
$$\begin{pmatrix} steel & coal & iron & Auto \\ 0.066667 & 0.46667 & 0.025 & 0.37 \end{pmatrix}$$

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.

(%i12) matrix (["steel", "coal", "iron", "Auto"],
[a0(1)*600, a0(2)*600, a0(3)*400, a0(4)*1000]);

(%o12)
$$\begin{pmatrix} steel & coal & iron & Auto \\ 40.0 & 280.0 & 10.0 & 370.0 \end{pmatrix}$$

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.

(%i14) 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 . X and X - B, which should be the same.

(%i15) A . X;

(%o15)
$$\begin{pmatrix} 440.0 \\ 460.0 \\ 400.0 \\ 600.0 \end{pmatrix}$$

(%i16) X - B;

(%o16)
$$\begin{pmatrix} 440 \\ 460 \\ 400 \\ 600 \end{pmatrix}$$

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

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

(%i17) M : diagmatrix (4,1) - A;

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

(%i18) rank (M);

(%o18) 4

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

(%i19) Minv : invert (M);

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

(%i21) X;
Minv . B;

$$(\%o20) \begin{pmatrix} 600 \\ 600 \\ 400 \\ 1000 \end{pmatrix}$$

$$(\%o21) \begin{pmatrix} 600.0 \\ 600.0 \\ 400.0 \\ 1000.0 \end{pmatrix}$$

which again shows consistency.

11.1.3 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 x n matrix M, with M[i,j] <= 0 (i # 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.

11.1.4 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] >= 0 for all (i, j) = 1, 2.

(%i3) killAB()$
A : genmatrix (a, 2,2);
I : diagmatrix (2, 1);
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 LPM (M, k) to compute the leading principle 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?

(%i4) lpm1 : LPM (M,1);

(lpm1) $1-a_{1,1}$

(%i5) lpm1 > 0;

(%o5) $1-a_{1,1}>0$

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 LPM (M, 2) > 0 require?

(%i6) lpm2 : LPM (M, 2);

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

(%i7) lpm2 > 0;

(%o7) $(1-a_{1,1})(1-a_{2,2})-a_{1,2}\,a_{2,1}>0$

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, 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 amount of commod. 1 used as direct and indirect inputs in producing one dollar's worth of commod. 1, must be 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.

11.1.5 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 x n) is the same as the nth leading principal minor LPM (M, m). 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 . X = B exists.

The Maxima function InputOutput (M, B) simply computes the n leading principle 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.

11.1.6 Leontief (A)

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

11.1.7 Pinput (A, j)

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

11.1.8 PItot (A, Xs)

PItot (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) . Xs = B, in which B is the matrix
column vector of final demands.

12 Characteristic Roots, Eigenvalue Tests [12.7]

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

A . V = c V
the scalar c is called the characteristic root, latent root, or eigenvalue, and the vector V is called the characteristic vector, or eigenvector. Using the identity matrix **I**, I . V = V, so
A . V = c **I** . V
which can be rearranged as
(A - c **I**) . V = 0
where (A - c **I**) is called the characteristic matrix of A. Since by assumption, V # 0, the characteristic matrix must be singular, and thus its determinant must vanish

12.1 Example 8

Let A be the 2 x 2 matrix

(%i1)
```
killAB()$
A : matrix ( [-6, 3], [3, -6] );
```

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

Let **I** be the 2 x 2 unit matrix

(%i2)
```
I : diagmatrix (2,1);
```

(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**) must be equal to zero.

(%i3) eqn : determinant (A - c*I) = 0;

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

(%i4) solns : solve(eqn, c);

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

(%i5) [c1, c2] : map ('rhs, solns);

(%o5) $[-9, -3]$

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

12.2 Example 9

Continuing with Example 8, the first root c1 = -9 in now used to find the characteristic vector (eigenvector).

(%i6) V1 : cvec ([v11, v12]);

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

Replacing c by c1, and V by V1 in the matrix equation (A - c I) . V = 0, we get

(%i7) (A - c1*I) . V1 = zeromatrix (2,1);

(%o7) $\begin{pmatrix} 3\ v12+3\ v11 \\ 3\ v12+3\ v11 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

The two equations are identical and require v12 = - v11, so add the normalization condition transpose (V1) . V1 = 1, which is satisfied if v11 = 1/sqrt(2)

(%i8) V1 : cvec ([1/sqrt(2), - 1/sqrt(2)]);

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

(%i9) transpose (V1) . V1;

(%o9) 1

Next, look for the eigenvector V2, using c = c2 as the eigenvalue.

(%i10) V2 : cvec ([v21, v22]);

(V2) $\begin{pmatrix} v21 \\ v22 \end{pmatrix}$

(%i11) (A - c2*I) . V2 = zeromatrix (2,1);

(%o11) $\begin{pmatrix} 3\, v22 - 3\, v21 \\ 3\, v21 - 3\, v22 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Both equations imply we need v22 = v11, and adding normalization condition gives

(%i12) V2 : cvec ([1/sqrt(2), 1/sqrt(2)]);

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

(%i13) transpose (V2) . V2;

(%o13) 1

12.3 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.

(%i14) [eivals, mult] : eigenvalues (A);

(%o14) [[−9,−3],[1,1]]

(%i15) eivals;

(%o15) [−9,−3]

(%i16) [c1, c2] : eivals;

(%o16) [−9,−3]

(%i17) c1;

(%o17) −9

12.4 eigenvectors (M)

Maxima has a function eigenvectors (M) which returns a list of two lists.
The first list has two lists: 1. a list of eigenvalues found, 2. a list of the multiplicities.
The second list contains a list of eigenvectors for each eigenvalue found.

```
(%i18) [EI, EV] : eigenvectors (A);
(%o18) [[[-9,-3],[1,1]],[[[1,-1]],[[1,1]]]]

(%i19) EI;
(%o19) [[-9,-3],[1,1]]

(%i20) [eivals, mult] : EI;
(%o20) [[-9,-3],[1,1]]

(%i21) eivals;
(%o21) [-9,-3]

(%i22) [c1, c2] : eivals;
(%o22) [-9,-3]

(%i23) c1;
(%o23) -9

(%i24) EV;
(%o24) [[[1,-1]],[[1,1]]]

(%i25) EV[1];
(%o25) [[1,-1]]

(%i26) EV[1][1];
(%o26) [1,-1]

(%i27) [v11, v12] : EV[1][1];
(%o27) [1,-1]

(%i28) v11;
(%o28) 1

(%i29) EV[2];
(%o29) [[1,1]]

(%i30) EV[2][1];
(%o30) [1,1]
```

(%i31) [v21, v22] : EV[2][1];

(%o31) [1,1]

At this point eigenvectors(A) has given us un-normalized eigenvector components.

(%i32) V1 : cvec ([v11, v12]);

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

(%i33) transpose (V1) . V1;

(%o33) 2

(%i34) V2 : cvec ([v21, v22]);

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

(%i35) transpose (V2) . V2;

(%o35) 2

12.5 Problem 12.43

Use eigenvalues to determine sign definiteness for the matrix A =

(%i36) A : matrix ([4, 6, 3], [0, 2, 5], [0, 1, 3]);

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

(%i37) [eivals, mult] : eigenvalues (A);

(%o37) $[[-\frac{\sqrt{21}-5}{2},\frac{\sqrt{21}+5}{2},4],[1,1,1]]$

(%i38) [c1, c2, c3] : float (eivals);

(%o38) [0.20871,4.7913,4.0]

With all three eigenvalues positive, A is positive definite.

12.6 Problem 12.46 (a)

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

```
(%i39) A : matrix ( [6, 6], [6, -3] );
```

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

```
(%i40) [eivals, mult] : eigenvalues (A);
(%o40) [[-6,9],[1,1]]
```

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

13 Binding and Nonbinding Constraints in Economics

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.

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

What are the objective and constraint functions?

The objective function: π = -x^2 + 10x - 6
The constraint: x ≤ 4 or 0 ≤ 4 – x

(%i2)
```
killAB()$
Pr : -x^2 + 10*x - 6;
soln : solve ( diff (Pr,x), x);
```

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

(soln) $[x=5]$

(%i3)
```
at (Pr, soln);
```

(%o3) 19

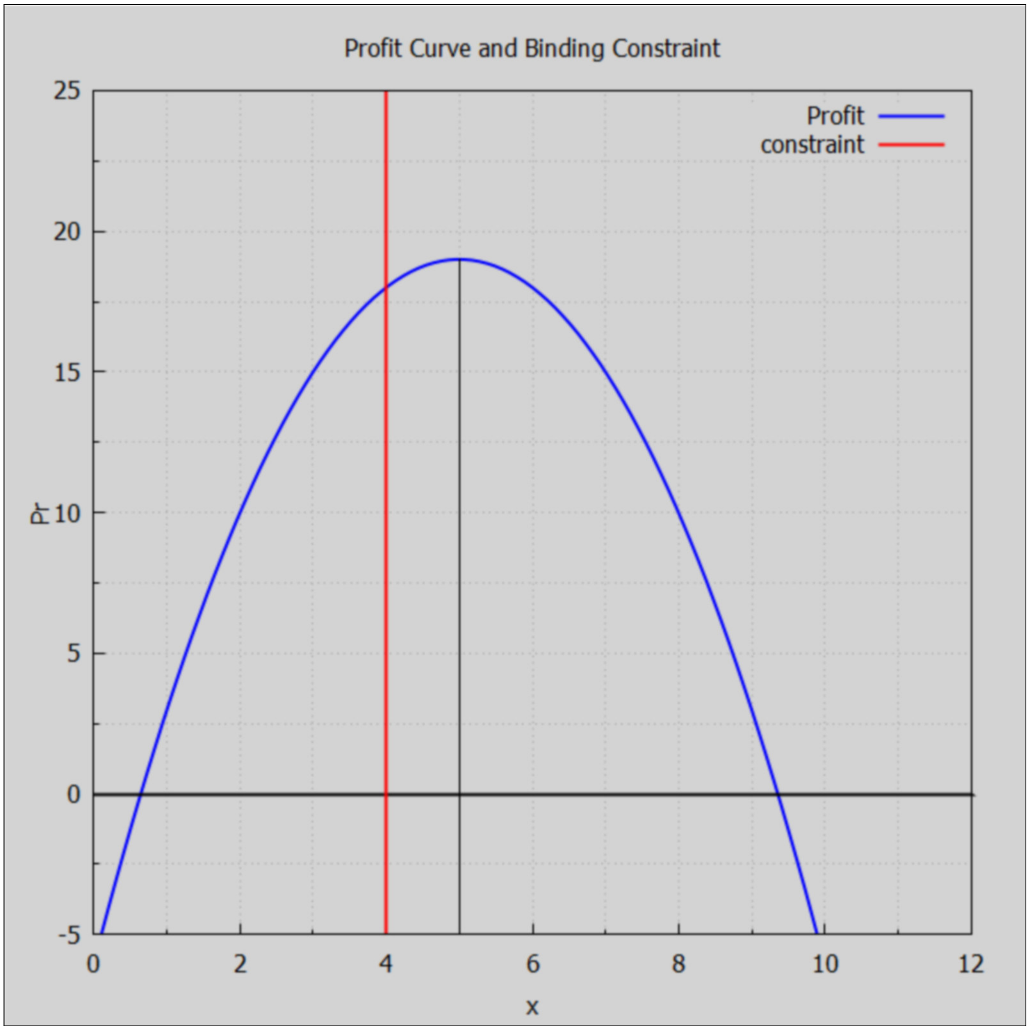
We can determine if this is a local maximum by looking at the sign of the second derivative.

(%i4)
```
at (diff (Pr,x,2), soln);
```

(%o4) -2

So the profit Pr (π) will be maximum if the output x = 5, but our constraint is that x <= 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 <= 4 (as close to the constraint-free optimum as possible).

(%i5)
```
wxdraw2d (xlabel = "x", ylabel = "Pr", yrange = [-5, 25],
   xaxis = true,  background_color = light_gray,
   title = "Profit Curve and Binding Constraint", grid = [2,2],
   key = "Profit",  explicit (Pr, x, 0, 12),
   color = black, key = "", explicit (0, x, 0, 12),
   line_width = 1, parametric (5, t, t, -5, 19), color = red,
   line_width = 2, key = "constraint", parametric (4, t, t, -5, 25) ),
   wxplot_size = [680, 680]$
```

(%t5) 

(%i6)
```
[at (Pr, x = 4),  at (Pr, x = 5)];
```

(%o6) [18,19]

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

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

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

(%i4)
```
killAB()$
f : x^a*y^b;
g : x + y;
L : f + lam*(10 - g);
gradL : jacobian ([L], [x, y, lam])[1];
```

(f) $x^a y^b$

(g) $y+x$

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

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

(%i5)
```
solns : solve (gradL, [x,y,lam]);
```

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

(%i6)
```
soln : solns[1];
```

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

With a > 0 and b > 0, both xs and ys are positive.

(%i7)
```
cp : rest (soln, -1);
```

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

(%i8)
```
[xs, ys] : map ('rhs, cp);
```

(%o8) $[\frac{10\,a}{b+a}, \frac{10\,b}{b+a}]$

With a > 0 and b > 0, f > 0 at cp.

(%i9)
```
at (f, cp);
```

(%o9) $\left(\frac{a}{b+a}\right)^a\left(\frac{b}{b+a}\right)^b 10^{b+a}$

Step 2. Find the set of points (x,y) which simultaneously satisfy the three equations ∂g/∂x = 0, ∂g/∂y = 0, and g (x,y) = 10.

(%i10)
```
gradg : jacobian ([g], [x, y])[1];
```

(gradg) $[1,1]$

Since ∂g/∂x = 1 and ∂g/∂y = 1 for all (x,y), there are no points (x,y) which simultaneously satisfy the conditions: gradg = [0, 0] and g(x,y) = 10.

Step 3. If the set S of points (x,y) on which f and g are defined has boundary points, find all boundary points (x,y) that satisfy x + y = 10.

The boundary points of the set S on which the objective function is defined is the set of points (x,y) with either x = 0 or y = 0. At every such point the objective function is 0.

Step 4. The points found in steps 1, 2, and 3 which yield the largest value of f are the critical points (xs, ys).

(%i11) [xs, ys];

(%o11) $\left[\frac{10\,a}{b+a},\frac{10\,b}{b+a}\right]$

(%i12) at (f, cp);

(%o12) $\left(\frac{a}{b+a}\right)^{a}\left(\frac{b}{b+a}\right)^{b}\,10^{b+a}$

which is greater than 0.

The value of the lagrangian at the maximal solution is the rate of change of the maximal value of the objective function as the constraint is relaxed.

(%i13) lams : at (lam, soln);

(lams) $(b+a)\left(\frac{a}{b+a}\right)^{a}\left(\frac{b}{b+a}\right)^{b}\,10^{b+a-1}$