

Optimization, Part A

TABLE OF CONTENTS

PREFACE -----	1
REFERENCES -----	2
RANK -----	3
THE MINORS OF A MATRIX -----	5
LEADING PRINCIPAL MINORS -----	6
THE JACOBIAN -----	7
QUADRATIC FORMS AND THE DISCRIMINANT -----	9
TAYLOR'S THEOREM AND THE HESSIAN MATRIX -----	19
UNCONSTRAINED OPTIMIZATION -----	22

```
(%i4) load(draw)$ set_draw_defaults(line_width=2, grid = [2,2], point_type = filled_circle,
      head_type = 'nofilled, head_angle = 20, head_length = 0.5,
      background_color = light_gray, draw_realpart=false)$
      fpprintprec : 5$ ratprint : false$
```

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

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

1 **Preface**

Dowling12A.wmxm uses Maxima to review quadratic form analysis and analysis of unconstrained optimization of functions of economic variables, following a small part of Ch. 11 and the first part of Ch. 12 of Introduction to Mathematical Economics (3rd ed), by Edward T. Dowling, (Schaum's Outline Series), McGraw-Hill, 2012. This text is a bargain, with many complete problems worked out in detail. You should compare Dowling's solutions, worked out "by hand", with what we do using Maxima here.

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

Use load ("Econ1.mac");

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

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 and the bordered Hessian matrix.

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

Included in 12A 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 analyzing unconstrained critical points, 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."

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

2 *References*

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

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

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

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

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

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

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

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

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

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

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

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

 Math 2070, Univ. of Sydney Optimization notes

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

 Wolfram summary of Mathematics methods for optimization.

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

3 rank (a matrix), 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 $\text{rank}(A) = n$, then A is nonsingular, $|A| \neq 0$, and there is no linear dependence.
- if $\text{rank}(A) < n$, 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(M)` calculates the determinant of M .

3.1 Example 1

```
(%i7) 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 individual elements of a matrix M can be found using $M[i,j]$, in which i and j are positive integers.

```
(%i8) A[1,2];
(%o8) 4
```

The elements of row 1 of matrix A is returned as a list, using `A[1]`.

```
(%i9) A[1];
(%o9) [6,4]
```

`list_matrix_entries(M)` returns a list of all the elements of the matrix M .

```
(%i10) list_matrix_entries(A);
(%o10) [6,4,7,9]
```

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

```
(%i12) determinant(A);
      determinant(B);
(%o11) 26
(%o12) 0
```

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

```
(%i13) rank (A);
```

```
(%o13) 2
```

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

Since $|B| = 0$, the matrix B is singular and linear dependence exists between its rows and columns. In this example, row 2 = 3/2 times row1, and col 2 = 3/2 times col 1.

```
(%i14) rank (B);
```

```
(%o14) 1
```

The rank of B is 1 which is less than the dimensions of B ($n = 2$), and since $\text{rank}(B) = 1$, 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.

```
(%i15) 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`.

```
(%i16) 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.

Here we get the matrix which results if we remove row 1 and column 1 from A.

```
(%i17) minor (A, 1,1);
```

```
(%o17) 
$$\begin{pmatrix} a_{2,2} & a_{2,3} \\ a_{3,2} & a_{3,3} \end{pmatrix}$$

```

```
(%i18) 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:

```
(%i19) fundef (Minor);
```

```
(%o19) 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.

```
(%i20) Minor (A, 1, 1);
```

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

```
(%i21) grind(%)$
```

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

```
(%i22) Minor (A, 3, 3);
```

```
(%o22)  $a_{1,1} a_{2,2} - a_{1,2} a_{2,1}$ 
```

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 x 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.

```
(%i24) A[1,1];
      LPM (A, 1);
```

```
(%o23) a1,1
```

```
(%o24) a1,1
```

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

```
(%i25) LPM (A, 2);
```

```
(%o25) a1,1 a2,2 - a1,2 a2,1
```

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.

```
(%i26) LPM (A, 3);
```

```
(%o26) a1,1 ( a2,2 a3,3 - a2,3 a3,2 ) - a1,2 ( a2,1 a3,3 - a2,3 a3,1 ) + a1,3
      ( a2,1 a3,2 - a2,2 a3,1 )
```

```
(%i27) determinant (A);
```

```
(%o27) a1,1 ( a2,2 a3,3 - a2,3 a3,2 ) - a1,2 ( a2,1 a3,3 - a2,3 a3,1 ) + a1,3
      ( a2,1 a3,2 - a2,2 a3,1 )
```

6 *jacobian (funcList, varList), Linear Independence [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.

`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])`.

```
(%i28) depends ( [f1, f2, f3], [x1, x2, x3] );
```

```
(%o28) [f1(x1, x2, x3), f2(x1, x2, x3), f3(x1, x2, x3)]
```

```
(%i29) J : jacobian( [f1, f2, f3], [x1, x2, x3]);
```

$$(J) \begin{pmatrix} \frac{d}{dx_1} f_1 & \frac{d}{dx_2} f_1 & \frac{d}{dx_3} f_1 \\ \frac{d}{dx_1} f_2 & \frac{d}{dx_2} f_2 & \frac{d}{dx_3} f_2 \\ \frac{d}{dx_1} f_3 & \frac{d}{dx_2} f_3 & \frac{d}{dx_3} f_3 \end{pmatrix}$$

```
(%i30) grind(J)$
```

```
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 f_1 with respect to all three variables. The first column elements are the first derivatives of all the functions with respect to x_1 .

If we are given a set of three equations

$$f_1 = 0, f_2 = 0, f_3 = 0,$$

then if the determinant of the jacobian J (just defined) is NOT equal to zero, the three equations ARE functionally independent.

If the determinant of the jacobian J 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 $n \times n$ matrix (where the number of independent variables is n). We will approach these concepts slowly.

6.0.1 Example 1

Given the pair of equations

$$f_1 = 5x_1 + 3x_2 = 0,$$

$$f_2 = 25x_1^2 + 30x_1x_2 + 9x_2^2 = 0,$$

we first set up the jacobian matrix of first derivatives of f_1 and f_2 .

f_1 and f_2 are defined as Maxima expressions (rather than Maxima functions). The implied equations are $f_1 = 0$ and $f_2 = 0$.


```
(%i33) 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) 
$$\begin{pmatrix} 5 & 3 \\ 30 x2+50 x1 & 18 x2+30 x1 \end{pmatrix}$$

```

```
(%i36) 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) 
$$\begin{pmatrix} 5 & 3 \\ 30 x2+50 x1 & 18 x2+30 x1 \end{pmatrix}$$

```

```
(%i37) grind(J)$
      matrix([5,3],[30*x2+50*x1,18*x2+30*x1])$
```

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

```
(%i38) 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 $f1^2$.

```
(%i40) f1^2, expand;
      f2;
```

```
(%o39) 9 x2^2+30 x1 x2+25 x1^2
```

```
(%o40) 9 x2^2+30 x1 x2+25 x1^2
```

7 Quadratic Forms and the Discriminant [12.3]

Determinants (in the form of leading principal 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

$$q = \text{transpose}(u) \cdot (D u)$$

in which u is written as a matrix column vector.

The Maxima function `cvec (alist)`, defined in `Econ1.mac`, creates a column matrix whose elements are the elements of the list `alist`.

```
(%i41) u : cvec ([x, y]);
```

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

```

```
(%i42) grind (u)$
```

```
matrix([x],[y])$
```

A quadratic form q 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 principal diagonal and dividing the coefficients of the nonsquared term equally between the off-diagonal positions:

```
(%i43) D : matrix ( [a, b/2], [b/2, c] );
```

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

```

```
(%i44) transpose (u);
```

```
(%o44) 
$$(x \ y)$$

```

```
(%i45) transpose (u) . D . u, expand;
```

```
(%o45) 
$$c y^2 + b x y + a x^2$$

```

We then evaluate the leading principal minors using our Maxima function `LPM(D,j)`. Since D is a 2×2 matrix, there are two leading principal minors.

```
(%i46) D1 : LPM (D,1);
```

```
(D1) 
$$a$$

```

(%i47) $D2 : \text{LPM}(D,2);$

(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 for the definiteness of a two variable quadratic form, 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.

Let $ex0$ be the first form, $ex1$ be the second form, and $expr$ being the difference. In the following, we want $expr$ to equal zero.

(%i50) $ex0 : a*x^2 + b*x*y + c*y^2;$
 $ex1 : a*(x + d*y)^2 + e*y^2;$
 $expr : ex0 - ex1, \text{expand};$

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

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

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

(%i51) $\text{facsum}(expr, y^2, x*y);$

(%o51) $-(e + a d^2 - c) y^2 - (2 a d - b) x y$

To arrive at $expr = 0$, we need 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.

(%i52) $ey2 : \text{ratcoeff}(expr, y^2);$

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

```
(%i53) exy : ratcoef (expr, x*y);
```

```
(exy) b-2 a d
```

```
(%i54) soln : solve ([ey2, exy], [d, e]);
```

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

```
(%i55) soln : soln[1];
```

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

```
(%i56) e : at( e, soln);
```

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

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

```
(%i57) ex1;
```

```
(%o57) 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 $D2/D1$, as we show here:

```
(%i58) [D1, D2, expand(D2/D1), expand (e)];
```

```
(%o58) [a, a c -  $\frac{b^2}{4}$ , c -  $\frac{b^2}{4a}$ , c -  $\frac{b^2}{4a}$ ]
```

So the general two variable quadratic form q 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.

```
(%i60) kill(e, D1, D2)$  
subst ([a = D1, e = D2/D1], ex1);
```

```
(%o60) 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 variable quadratic form)

Given the quadratic form z with numerical coefficients

$$z = 2x^2 + 5xy + 8y^2,$$

form the symmetric matrix D as described above and test for sign-definiteness of the given quadratic form.

```
(%i61) D : matrix ( [2, 5/2], [5/2, 8] );
```

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

The leading principal minor D_1 is the determinant of the matrix produced by deleting all but the first row and first column, which determinant is just the matrix element $D[1,1] = 2$.

```
(%i62) determinant( matrix ([2]));
```

```
(%o62) 2
```

Using our Maxima function $LPM(D, j)$ for the j 'th leading principal minor of the matrix D , we get:

```
(%i63) LPM (D, 1);
```

```
(%o63) 2
```

The leading principal minor D_2 is the determinant of the matrix produced by deleting all but the first two rows and the first two columns of the matrix D , which will produce the determinant of D

```
(%i64) determinant (D);
```

```
(%o64)  $\frac{39}{4}$ 
```

```
(%i65) LPM (D, 2);
```

```
(%o65)  $\frac{39}{4}$ 
```

```
(%i66) %, numer;
```

```
(%o66) 9.75
```

Since both D_1 and D_2 are greater than zero, z is positive definite and z is positive for all values of the variables (x,y) , excepting the case in which both variables are zero.

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.

(%i67) **Qtest (D);**

positive definite

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

(%o67) **[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

$$q = \text{transpose}(u) \cdot D \cdot u$$

(%i69) **kill(a,b,c,d,e,f)\$**

D : matrix ([a, b/2, c/2], [b/2, d, e/2], [c/2, e/2, f]);

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

(%i70) **u : matrix ([x], [y], [z]);**

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

(%i71) **transpose (u) . D . u, expand;**

(%o71) **$f z^2 + e y z + c x z + d y^2 + b x y + a x^2$**

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

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

(%i72) D1 : LPM (D, 1);

(D1) a

(%i73) D2 : LPM (D, 2);

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

(%i74) 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 .

(%i78) kill (g, h, k, l, m)\$

$ex0 : a*x^2 + b*x*y + c*x*z + d*y^2 + e*y*z + f*z^2;$

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

$expr : ex0 - ex1, expand;$

(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.

```
(%i79) ez2 : coeff (expr, z^2);
```

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

```
(%i80) exy : ratcoef (expr, x*y);
```

```
(exy) b - 2 a g
```

```
(%i81) exz : ratcoef (expr, x*z);
```

```
(exz) c - 2 a h
```

```
(%i82) ey2 : coeff (expr, y^2);
```

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

```
(%i83) 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.

```
(%i84) solns : solve ([ez2, exy, exz, ey2, eyz], [g, h, k, l, m]);
```

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

```
(%i85) soln : solns[1];
```

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

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

```
(%i86) k : at (k, soln);
```

```
(k)  $\frac{4ad - b^2}{4a}$ 
```

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


```
(%i87) [D1, D2, expand (D2/D1), expand (k)];
```

```
(%o87) [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.

```
(%i88) 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.

```
(%i89) [expand (D3/D2), expand (m)];
```

```
(%o89) [ -  $\frac{b^2 f}{4 a d - b^2} + \frac{a d f}{a d - \frac{b^2}{4}} - \frac{a e^2}{4 a d - b^2} + \frac{b c e}{4 a d - b^2} - \frac{c^2 d}{4 a d - b^2}, \frac{4 a d f}{4 a d - b^2}$   


$$- \frac{b^2 f}{4 a d - b^2} - \frac{a e^2}{4 a d - b^2} + \frac{b c e}{4 a d - b^2} - \frac{c^2 d}{4 a d - b^2} ]$$

```

```
(%i90) is (equal (expand (D3/D2), expand (m) ));
```

```
(%o90) 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.

```
(%i91) ex1;
```

```
(%o91) k (l z + y)^2 + a (h z + g y + x)^2 + m z^2
```

```
(%i93) kill (k, m, D1, D2, D3)$  

subst ([a = D1, k = D2/D1, m = D3/D2], ex1);
```

```
(%o93) 
$$\frac{D2 (l z + y)^2}{D1} + D1 (h z + g y + x)^2 + \frac{D3 z^2}{D2}$$

```

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 variable quadratic form)

Given the quadratic form in three variables (x_1, x_2, x_3)
 $y = 5x_1^2 - 6x_1x_2 + 3x_2^2 - 2x_2x_3 + 8x_3^2 - 3x_1x_3$,
 check for sign definiteness.

The coefficients of the squared terms continue to go on the principal diagonal, while the coefficient of x_1x_2 is equally divided between $D[1,2]$ and $D[2,1]$, etc.

```
(%i94) D : matrix ( [5, -3, -3/2], [-3, 3, -1], [-3/2, -1, 8] );
```

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

```
(%i95) D1 : LPM (D, 1);
```

```
(D1) 5
```

```
(%i96) D2 : LPM (D, 2);
```

```
(D2) 6
```

```
(%i97) D3 : LPM (D, 3);
```

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

```
(%i98) %, numer;
```

```
(%o98) 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 (x_1, x_2, x_3) such that not all three are simultaneously zero.

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

(%i99) Qtest (D);

positive definite

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

(%o99) [5.0,6.0,27.25]

7.5 Problem 12.7 (a) (3 variable quadratic form)

Use discriminants to determine the sign definiteness of the following function:

$$y = -2x_1^2 + 4x_1x_2 - 5x_2^2 + 2x_2x_3 - 3x_3^2 + 2x_1x_3.$$

(%i100) D : matrix ([-2, 2, 1], [2, -5, 1], [1, 1, -3]);

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

(%i101) D1 : LPM (D, 1);

(D1) -2

(%i102) D2 : LPM (D, 2);

(D2) 6

(%i103) D3 : LPM (D, 3);

(D3) -7

Since $D_1 < 0$, $D_2 > 0$, and $D_3 > 0$, y is a negative definite quadratic form and $y < 0$ for all values of x_1, x_2, x_3 as long as they are not all zero simultaneously.

(%i104) Qtest (D);

negative definite

[[LPM1,-2],[LPM2,6],[LPM3,-7]]

(%o104) [-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, is

$$f(x + dx) = f(x) + dx f_x + (1/2!) dx^2 f_{xx} + \dots$$

In n dimensions, Taylor's theorem is, with x standing for variables (x_1, x_2, \dots, x_n) and dx standing for $(dx_1, dx_2, \dots, dx_n)$, and in the following the third term assumes dx is a matrix column vector:

$$f(x + dx) = f(x) + dx \cdot \text{grad}(f) + (1/2!) \text{transpose}(dx) \cdot (H dx) + \dots,$$

where H is the $n \times n$ Hessian matrix with components

$$H[i,j] = \text{diff}(f, x_i, 1, x_j, 1) \text{ if } i \neq j,$$

$$H[i,j] = \text{diff}(f, x_i, 2) \text{ if } i = j.$$

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

$$f(x + h, y + k) = f(x, y) + h \frac{\partial f}{\partial x} + k \frac{\partial f}{\partial y} + (1/2) (h^2 \frac{\partial^2 f}{\partial x^2} + 2 h k \frac{\partial^2 f}{\partial x \partial y} + k^2 \frac{\partial^2 f}{\partial y^2}) + \dots$$

We can write the quadratic term in terms of the Hessian matrix, making use of Young's theorem: $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$. In the following we use the Maxima function `hessian(f, [x1, x2])` for a function of two variables. In the following, 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 .

```
(%i105) [f, x, y];
```

```
(%o105) [f, x, y]
```

```
(%i106) depends (f, [x, y]);
```

```
(%o106) [f(x, y)]
```

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

```
(%i107) 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 f_{xx}) on the principal diagonal, and the second order cross partials (like f_{xy}) off the principal diagonal. Maxima automatically uses the general result that $\partial^2 f / \partial x \partial y = \partial^2 f / \partial y \partial x$, and H is then obviously symmetric.

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

```
(%i108) [h, k];
```

```
(%o108) [h, k]
```

```
(%i109) dX : matrix ([h], [k] );
```

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

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

```
(%i110) transpose (dX);
```

```
(%o110) (h k)
```

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

```
(%i111) transpose (dX) . dX;
```

```
(%o111)  $k^2 + h^2$ 
```

```
(%i112) transpose (dX) . H . dX;
```

```
(%o112)  $k \left( \left( \frac{d^2}{d y^2} f \right) k + \left( \frac{d^2}{d x d y} f \right) h \right) + h \left( \left( \frac{d^2}{d x d y} f \right) k + \left( \frac{d^2}{d x^2} f \right) h \right)$ 
```

```
(%i113) expand(%);
```

```
(%o113)  $\left( \frac{d^2}{d y^2} f \right) k^2 + 2 \left( \frac{d^2}{d x d y} f \right) h k + \left( \frac{d^2}{d x^2} f \right) h^2$ 
```

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.

```
(%i114) z;
```

```
(%o114) z
```

```
(%i116) depends (f, [x,y,z])$
        hessian (f, [x,y,z]);
```

$$(\%o116) \begin{pmatrix} \frac{d^2}{d x^2} f & \frac{d^2}{d x d y} f & \frac{d^2}{d x d z} f \\ \frac{d^2}{d x d y} f & \frac{d^2}{d y^2} f & \frac{d^2}{d y d z} f \\ \frac{d^2}{d x d z} f & \frac{d^2}{d y d z} f & \frac{d^2}{d z^2} f \end{pmatrix}$$

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 $f(x_1, x_2, \dots, x_n)$ be the "objective function" of n variables.

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

The quickest route to finding unconstrained 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 that `solve` returns real values, screens out real 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 $\text{grad}(f) = 0$. A list of the expressions $\text{diff}(f, x_j)$ 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, x2, x3, ..., xn]).
```

(If solve has difficulties, or you want to check on solve, you can try to `_poly_solve`.)

If solve is successful in finding some or all of the solutions, you will get a list of a list, as in `[[x1 = 2, x2 = -1/2, ...]]` for a single 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, such as `[[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

```
f1 : at (f, cp1),
```

```
f2 : 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 $k = 1, 2, \dots, n$ (defined in `Econ1.mac`).

For example

```
H1 : LPM (Hcp1, 1);
```

```
H2 : LPM (Hcp1, 2);
```

```
.....
```

```
Hn : LPM (Hcp1, n);
```

The leading principal minors are the determinants of submatrices of $Hcp1$.

d.) A sufficient condition for a relative MINIMUM is that all the determinants

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 $x = 0$ 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 $f''(c) \leq 0$.

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: $z = 3x^2 - xy + 2y^2 - 4x - 7y + 12$.
Find any critical points and use the Hessian matrix test to examine the nature of such.

I will use f instead of z .

```
(%i117) 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]).

```
(%i118) optimum (f, [x, y]);
```

```
lagrangian = 2 y^2 - x y - 7 y + 3 x^2 - 4 x + 12
```

```
solve returns [[x=1,y=2]]
```

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

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

```
(%o118) 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.

```
(%i119) cp;
```

```
(%o119) [[x=1,y=2]]
```

```
(%i120) cp[1];
```

```
(%o120) [x=1,y=2]
```

```
(%i121) at (f, cp[1]);
```

```
(%o121) 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:

```
(%i122) gradf : jacobian ( [f], [x, y])[1];
```

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

```
(%i123) 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.

```
(%i124) cp1 : solns[1];
```

```
(cp1) [x=1, y=2]
```

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

You can either use

```
(%i125) subst (cp1, f);
```

```
(%o125) 3
```

or

```
(%i126) at (f, cp1);
```

```
(%o126) 3
```

or the interactive work form (not recommended by the Maxima experts for deep reasons):

```
(%i127) f, cp1;
```

```
(%o127) 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).

```
(%i128) 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 $f(x,y)$ 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 minimum if the first leading principal minor $H1 > 0$ and the second leading principal minor $H2 > 0$ as well.

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 $H[1,1] = f_{xx} = 6$. The determinant of such a one element matrix is simply the value of $H[1,1] = 6$.

```
(%i129) 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 2×2 matrix.

```
(%i130) 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. We used this function in Dowling, Ch. 5 (Dowling05.wmxm). This function shows a 3d plot of the surface $f(x,y)$ near the given critical point.

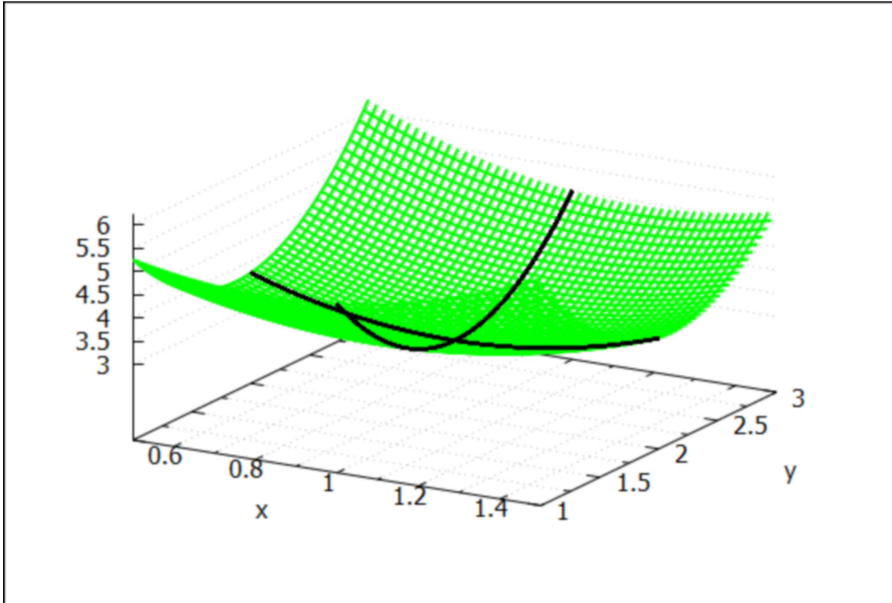
```
(%i131) [f, cp1];
```

```
(%o131) [2 y^2 - x y - 7 y + 3 x^2 - 4 x + 12, [x=1, y=2]]
```

(%i132) plotCP (f, cp1)\$

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

(%t132)



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 z_{xx} and z_{yy} , and is able to distinguish an inflection point from a saddle point.

(%i133) 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.

```
(%i134) [eivals, eimult] : eigenvalues (H);
```

```
(%o134) [[5-√2,√2+5],[1,1]]
```

We are interested in the numerical values of the eigenvalues.

```
(%i135) float (eivals);
```

```
(%o135) [3.5858,6.4142]
```

or, you can use (in interactive work)

```
(%i136) eivals, numer;
```

```
(%o136) [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) (see also our Ch. 5 work):

```
(%i137) Analyze (f, cp1)$
```

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

9.5 Ch. 12, Example 4: Extrema of $f(x_1, x_2, x_3)$

Find the critical point(s) and their nature for $f(x_1,x_2,x_3)$ defined by the expression:

$$f = -5x_1^2 + 10x_1 + x_1x_3 - 2x_2^2 + 4x_2 + 2x_2x_3 - 4x_3^2.$$

First the quick way with our Maxima function `optima (f,vL)`.

```
(%i140) 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
      solve returns [[x1 = 24/23, x2 = 28/23, x3 = 10/23]]
      optimum only evaluates real non-negative solutions
      cp1 [x1 = 24/23, x2 = 28/23, x3 = 10/23] , relative maximum, value = 7.6522
(%o140) done
```

```
(%i141) cp;
(%o141) [[x1 = 24/23, x2 = 28/23, x3 = 10/23]]
```

```
(%i142) float(cp);
(%o142) [[x1 = 1.0435, x2 = 1.2174, x3 = 0.43478]]
```

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

```
(%i143) gradf : jacobian ([f], varList)[1];
(gradf) [x3 - 10 x1 + 10, 2 x3 - 4 x2 + 4, -8 x3 + 2 x2 + x1]
```

```
(%i144) J : jacobian ( gradf, varList );
(J)      
$$\begin{pmatrix} -10 & 0 & 1 \\ 0 & -4 & 2 \\ 1 & 2 & -8 \end{pmatrix}$$

```

```
(%i145) JD : determinant (J);
(JD)      -276
```

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

```
(%i146) solns : solve ( gradf, varList);
(solns) [[x1 = 24/23, x2 = 28/23, x3 = 10/23]]
```

```
(%i147) cp : solns [1];
```

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

```
(%i148) cp, numer;
```

```
(%o148)  $[x1 = 1.0435, x2 = 1.2174, x3 = 0.43478]$ 
```

Value of f at cp:

```
(%i149) float (at (f, cp));
```

```
(%o149) 7.6522
```

Hessian matrix

```
(%i150) 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.

```
(%i151) [LPM (H,1), LPM (H, 2), LPM (H, 3) ];
```

```
(%o151)  $[-10, 40, -276]$ 
```

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.

```
(%i152) CPtest (f, cp);
```

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

```
(%o152) 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.


```
(%i153) cpn : float (cp);
(cpn) [x1=1.0435,x2=1.2174,x3=0.43478]
```

```
(%i154) [x10, x20, x30] : map ('rhs, cpn);
(%o154) [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.

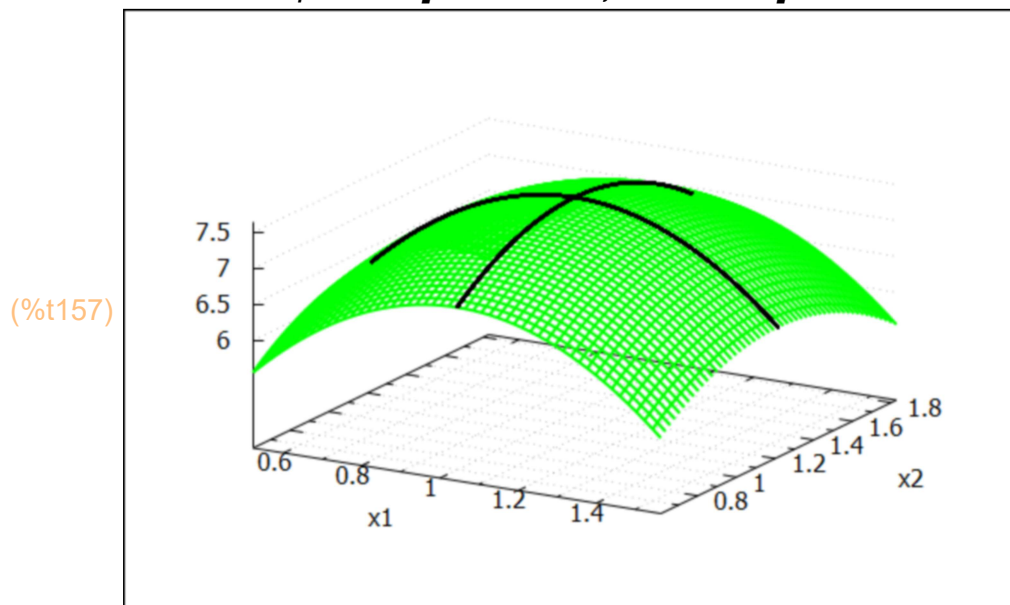
```
(%i155) F (x1, x2, x3) := "f;
(%o155) F (x1, x2, x3) := -4 x32 + 2 x2 x3 + x1 x3 - 2 x22 + 4 x2 - 5 x12 + 10 x1
```

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

```
(%i156) cpn12 : rest (cpn, -1);
(cpn12) [x1=1.0435, x2=1.2174]
```

In the following call to plotCP, we replace x3 by the number x30 = 0.43...

```
(%i157) plotCP (F (x1, x2, x30), cpn12);
surface of -2 x22 + 4.8696 x2 - 5 x12 + 10.435 x1 - 0.75614
near critical point = [x1=1.0435, x2=1.2174]
```



(%o157)

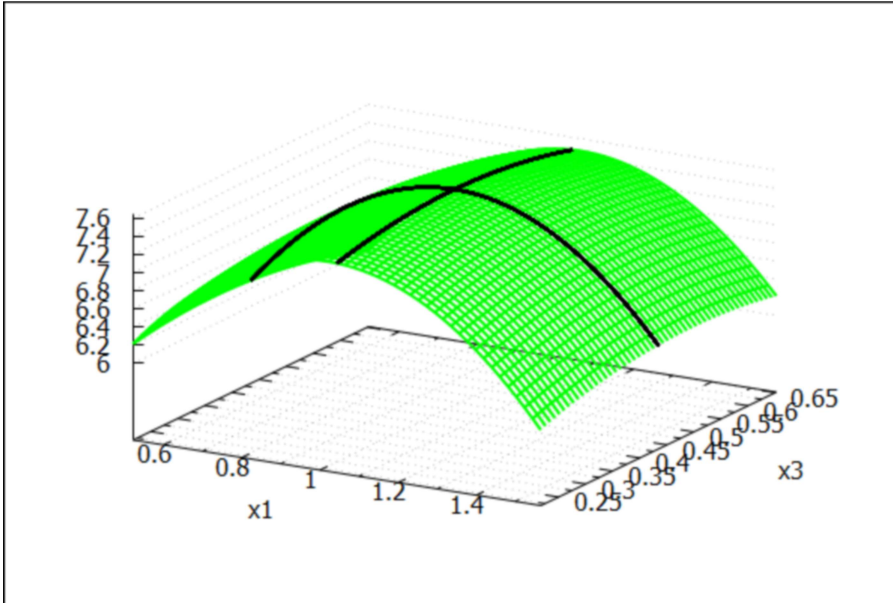
Next we hold x2 constant at x20.

```
(%i158) cp13 : [cpn[1], cpn[3]];
(cp13) [x1=1.0435, x3=0.43478]
```

(%i159) plotCP (F (x1,x20, x3), cp13)\$

surface of $-4 x_3^2 + x_1 x_3 + 2.4348 x_3 - 5 x_1^2 + 10 x_1 + 1.9055$
 near critical point = $[x_1=1.0435, x_3=0.43478]$

(%t159)



Finally, we hold x1 fixed at x10:

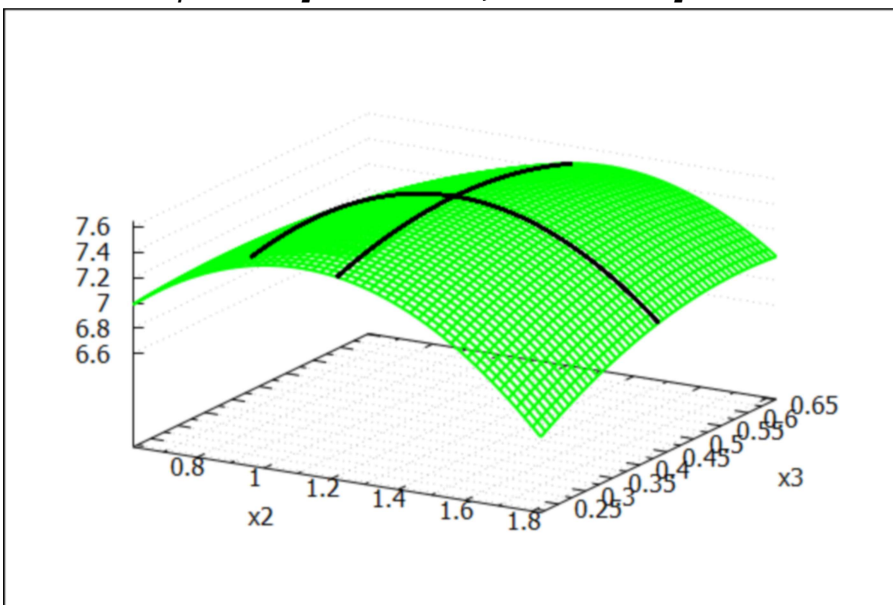
(%i160) cp23 : rest (cpn, 1);

(cp23) $[x_2=1.2174, x_3=0.43478]$

(%i161) plotCP (F (x10, x2, x3), cp23)\$

surface of $-4 x_3^2 + 2 x_2 x_3 + 1.0435 x_3 - 2 x_2^2 + 4 x_2 + 4.9905$
 near critical point = $[x_2=1.2174, x_3=0.43478]$

(%t161)



9.6 Problem 12.8 Extrema of $f(x_1, x_2, x_3)$

Investigate optimizing critical points for

$$f = 3x_1^2 - 5x_1 - x_1x_2 + 6x_2^2 - 4x_2 + 2x_2x_3 + 4x_3^2 + 2x_3 - 3x_1x_3.$$

```
(%i164) 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
      solve returns [[x1 = 25/28, x2 = 23/56, x3 = - 1/56]]
      optimum only evaluates real non-negative solutions
(%o164) no real non-negative solutions
```

Dowling's solution (in text, by hand) finds one critical point, for which $x_3 < 0$, which optimum ignores. Recall that optimum is designed for Economists, who deal with variables which cannot be negative (usually). A version called optimumAll, with the same syntax as optimum, does not apply the non-negative filter.

9.6.1 optimumAll (f, varL) Use Example

optimum is really designed for Economics problems for which the search variables are assumed to be non-negative.

Use optimumAll (f, varList) to accept ALL solutions found by solve.

```
(%i165) 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 = 25/28, x2 = 23/56, x3 = - 1/56] , relative minimum, value = -
      3.0714
(%o165) done
(%i166) float (cp);
(%o166) [[x1 = 0.89286, x2 = 0.41071, x3 = -0.017857]]
(%i167) at (f, %[1]);
(%o167) -3.0714
```

9.6.2 Finding and Evaluating Critical Points "by hand"...

Now for the long way:

```
(%i168) 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]
```

```
(%i169) determinant (jacobian (gradf, varList ) );
```

```
(%o169) 448
```

The Jacobian determinant is not zero, so the three equations implied by the list gradf, ie., gradf = [0,0,0], are functionally independent, and we expect solve to find a symbolic solution.

```
(%i170) varList;
```

```
(%o170) [x1, x2, x3]
```

```
(%i171) solns : solve (gradf, varList);
```

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

```
(%i172) cp : solns[1];
```

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

```
(%i173) at (f, cp), numer;
```

```
(%o173) -3.0714
```

```
(%i174) CPtest (f, cp);
```

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

```
(%o174) done
```

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

```
(%i175) 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.

```
(%i176) [LPM (H,1), LPM (H, 2), LPM (H,3)];
```

```
(%o176) [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 (dx_1, dx_2, dx_3) and the expression f has a relative minimum at that point and f is locally convex.

As in Example 4 above, we can call plotCP for 3d views of this minimum after some prep:

```
(%i177) [x10, x20, x30] : float (map ('rhs, cp));
```

```
(%o177) [0.89286,0.41071, -0.017857]
```

Use two single quotes (') in front of the expression f to define the Maxima function F .

```
(%i178) F( x1, x2, x3) := 'f;
```

```
(%o178) 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
```

```
(%i179) 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
```

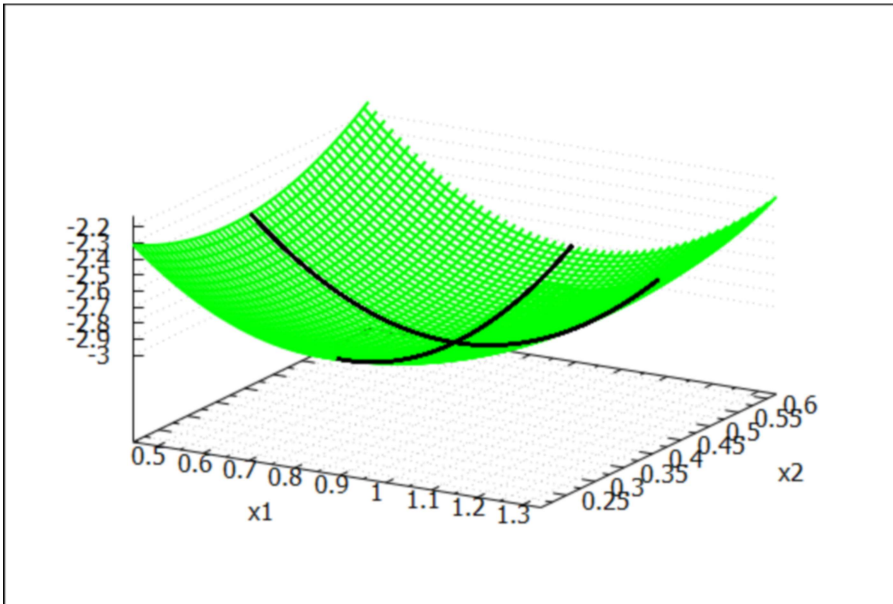
For fixed $x_3 = x_{30}$, look at (x_1, x_2) local behavior:

```
(%i180) plotCP (F (x1, x2, x3), rest (cp, -1))$
```

surface of $6x_2^2 - x_1x_2 - 4.0357x_2 + 3x_1^2 - 4.9464x_1 - 0.034439$

near critical point = $\left[x_1 = \frac{25}{28}, x_2 = \frac{23}{56} \right]$

(%t180)



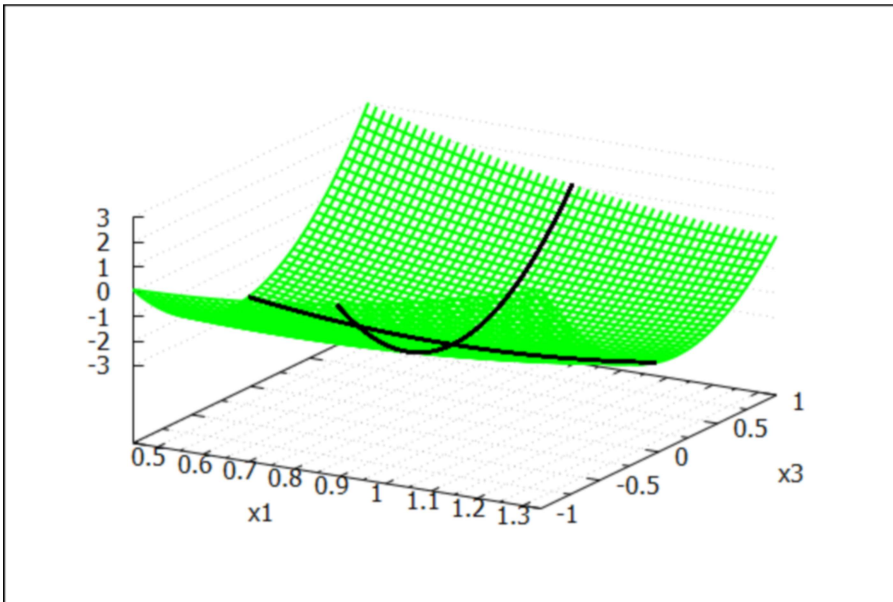
For fixed $x_2 = x_{20}$, look at (x_1, x_3) behavior:

```
(%i181) plotCP (F (x1, x20, x3), [cp[1], cp[3]])$
```

surface of $4x_3^2 - 3x_1x_3 + 2.8214x_3 + 3x_1^2 - 5.4107x_1 - 0.63074$

near critical point = $\left[x_1 = \frac{25}{28}, x_3 = -\frac{1}{56} \right]$

(%t181)



For fixed $x_1 = x_{10}$, look at (x_2, x_3) behavior:

```
(%i182) plotCP (F (x10, x2, x3), [cp[2], cp[3]] )$
```

surface of $4 x_3^2 + 2 x_2 x_3 - 0.67857 x_3 + 6 x_2^2 - 4.8929 x_2 - 2.0727$

near critical point = $\left[x_2 = \frac{23}{56}, x_3 = -\frac{1}{56} \right]$

```
(%t182)
```

