1980

Determiningeons : a computer program for approximating lie generators admitted by dynamical systems

Gregory G. Nagao
University of the Pacific

DETERMININGEQNS:

A Computer Program

For Approximating Lie Generators

Admitted By Dynamical systems

A Thesis

Submitted to

the Faculty of the Department of Physics

University of the Pacific

In Partial Fulfillment of

the Requirements for the Degree of

Master of Science

by

Gregory G. Nagao

August, 1980

Dedicated

with love

to

my wife Sharon

my mother and my father

and

my two sisters

In appreciation for the sacrifices made
in order to let me pursue my studies in physics.

Acknowledgments

The author is especially grateful to Professor Carl E. Wulfman who has given him much more than just guidance and assistance. The overwhelming enthusiasm of professor Wulfman has keyed the inspiration of the author and made this experience one which he will long remember.

Thanks to the University of the Pacific's physics department for bearing with me through this endeavor. The friendships gained here shall always be treasured - especially that of Loren Quan, a true friend never to be forgotten, with whom so many experiences have been shared.

Last, but not least, the author can not express the gratitude felt toward those who truly made this all possible - my wife, my parents, and my two sisters.

Table of Contents

Introduction

As was recognized by some of the most reputable physicists of the world such as Galileo and Einstein, the basic laws of physics must inevitably be founded upon invariance principles. Galilean and special relativity stand as historical landmarks that emphasize this message. It's no wonder that the great developments of modern physics (such as those in elementary particle physics) have been keyed upon this concept.

The modern formulation of classical mechanics (see Abraham and Marsden [1]) is based upon "qualitative" or geometric analysis. This is primarily due to the works of Poincaré. Poincaré showed the value of such geometric analysis in the solution of otherwise insoluble problems in stability theory. It seems that the insights of Poincare have proven fruitful by the now famous works of Kolmogorov, Arnold, and Moser. The concepts used in this geometric theory are again based upon invariance principles, or symmetries.

The work of Sophus Lie from 1873 to 1893 laid the groundwork for the analysis of invariance or symmetry principles in modern physics. His primary studies were those of partial differential equations. This led him to the study of the theory of transformations and inevitably to the analysis of abstract groups and differential geometry. Here we show some further applications of Lie group theory through the use of transformation groups. We emphasize the use of transformation invariance to find conservation laws and dynamical properties in chemical physics.

II. Basic Theory

2-1. Abstract Groups

An abstract group is a set of "points" $g_i \in G$ for $i=0,\ldots,n$, such that, under group "multiplication", we have for $i,j,k = 0,\ldots,n$:

$$
\begin{array}{llll}
\text{(a)} & g_i \cdot g_j = g_k & & \text{closure} \\
\text{(b)} & g_i \cdot (g_j \cdot g_k) = (g_i \cdot g_j) \cdot g_k & & \text{associativity} \\
\text{(c)} & g_o \cdot g_i = g_i = g_i \cdot g_o & \text{for all } g_i & \text{identity} \\
\text{(d)} & g_o = g_i \cdot g_i^{-1} = g_i \cdot g_k = g_k \cdot g_i \;\; ; \; g_k \equiv g_i^{-1} & & \text{unique inverse.}
\end{array} \quad (1)
$$

A continuous (or topological) group is one in which there exists a topological manifold (T) with an algebraic group structure such that the mappings:*

$$
\begin{array}{lll}
\text{(a)} & \Phi: \;\; \sigma \times \tau \rightarrow \sigma\tau & \sigma, \tau \in T \\
\text{(b)} & \psi: \;\; \tau \rightarrow \tau^{-1} &
\end{array} \quad (2)
$$

are continuous.

To facilitate the study of the global properties of groups, it is helpful to define the "building blocks" of the global groups -- Lie groups. Consider a connected component of a continuous group containing the identity element (g_o) and any other element g_k. Then the connectivity requirement implies that any group operation g_k can be built up by products of "infinitesimal" group operations δg_i, i.e.

$$g_k = g_\infty \cdots (g_3 \cdot g_2^{-1}) \cdot (g_2 \cdot g_1^{-1}) \cdot (g_1 \cdot g_o^{-1}) \cdot g_o \quad (3)$$

---

*Gilmore, Robert, Lie Groups, Lie Algebras and Some of Their Applications, pg.63.

where g_i and g_{i+1} lie on a line connecting g_o and g_k and are within a common neighborhood (thus $g_{i+1} \cdot g_i^{-1}$ lies near the identity). A <u>Lie group</u> is a connected component of a continuous group, and a <u>local Lie group</u> is a Lie group in the neighborhood of the identity.

Lie's three theorems and their converses, along with Taylor's theorem, establish an equivalence between Lie groups and Lie algebras. In his three theorems, Lie: (1) connects infinitesimal generators of Lie algebras with elements of a simply connected Lie group, (2) connects the structure constants to the topological group composition function ϕ, and (3) shows that the structure constants for a Lie algebra provide a matrix representation for the algebra. Taylor's theorem then provides the mapping between the Lie algebra and Lie group -- the exponential map:

$$\alpha U \rightarrow e^{\alpha U} \tag{4}$$

where U is the element of a Lie algebra. With these connections, then, it becomes apparent that we need only study the Lie algebras (through the use of transformation groups) to determine the structure of our local Lie group. For our studies of physical systems in this thesis, we will examine the invariance Lie group transformations admitted by our equations of motion and the manifolds upon which they are defined. In this way we shall determine the local Lie groups admitted by the system.

2-2. A Continuous Group of Infinitesimal Transformations

Consider a transformation $T(\alpha)$ parameterized by α whose topological space $\mathcal{T}$ is R^1, which maps an n-dimensional geometric space G^n onto G^n, i.e.

$$\underset{\sim}{x} \rightarrowtail \bar{\underset{\sim}{x}} = g(\underset{\sim}{x},\alpha) \tag{1}$$

where $\underset{\sim}{x} = (x^1, x^2, x^3, \ldots, x^n)$; $\underset{\sim}{x}, \bar{\underset{\sim}{x}} \in G^n$ (G^n, a differentiable manifold)
$\bar{\underset{\sim}{x}} = (\bar{x}^1, \bar{x}^2, \bar{x}^3, \ldots, \bar{x}^n)$.

~~Thus, $T(\alpha)$ is a faithful mapping from G^n to G^n. Furthermore, suppose~~ $T(\alpha)$ satisfies the following relations:

$$\begin{array}{lll}
T(\alpha^i)\,T(\alpha^j) = T(\alpha^k); & i,j,k = 0,1,\ldots,\infty & \text{closure} \\
[T(\alpha^i)\,T(\alpha^j)]\,T(\alpha^k) = T(\alpha^i)[T(\alpha^j)\,T(\alpha^k)] & & \text{associativity} \\
T(\alpha^o)\,T(\alpha^i) = T(\alpha^i)\,T(\alpha^o) = T(\alpha^i) & & \text{identity} \\
T(\alpha^i)\,T^{-1}(\alpha^i) = T^{-1}(\alpha^i)\,T(\alpha^i) = T(\alpha^o) & & \text{unique inverse}
\end{array} \tag{2}$$

where α takes on the values $-\infty \leq \alpha \leq +\infty$. Then $T(\alpha)$ is said to form a <u>one-parameter continuous group of transformations</u>. Furthermore, our group is a Lie group since the topological space is defined as R^1.

If we now impose a local condition (requiring $\alpha \rightarrow \delta\alpha$) on the topological space (parameter space) $\mathcal{T}$ of $T(\alpha)$, then our transformation (which we shall now denote by $T(\delta\alpha)$) is said to form a one-parameter continuous Lie Group of infinitesimal transformations consisting of a topological group in the topological space $\mathcal{T}$ and a geometric space G^n upon which the transformation $T(\delta\alpha)$ acts.*,** The general form for such a transformation is:

---

*Gilmore, <u>Lie Groups, Lie Algebras, and Some of Their Applications</u>, pg. 64.

**Campbell, <u>Continuous Groups</u>, pg. 6.

$$x^i \rightarrowtail \bar{x}^i = x^i + \delta\alpha\xi^i(x^1,\ldots,x^n) \tag{3}$$

where $\delta\alpha$ is an infinitesimal parameter and thus terms of higher order may be neglected.

Consider a function $\phi(\bar{x}^1,\ldots,\bar{x}^n)$. This function may be expanded in terms of $\delta\alpha$ around $\delta\alpha = 0$, neglecting terms of $O(\delta\alpha^2)$ and higher. Then we get

$$\begin{aligned}
\phi(\bar{x}^1,\ldots\bar{x}^n) &= \phi(\bar{x}^1,\ldots,\bar{x}^n)\big|_{\delta\alpha=0} + \partial_{(\delta\alpha)}(\phi)\big|_{\delta\alpha=0}(\delta\alpha) \\
&= \phi(x^1+\delta\alpha\xi^1,\ldots,x^n+\delta\alpha\xi^n)\big|_{\delta\alpha=0} + \partial_{\delta\alpha}\phi(x^1+\delta\alpha\xi^1,\ldots,x^n+\delta\alpha\xi^n)\big|_{\delta\alpha=0}(\delta\alpha) \\
&= \phi(x^1,\ldots,x^n) + (\partial_{\bar{x}^1}\phi)(\partial_{\delta\alpha}\bar{x}^1)\big|_{\delta\alpha=0}(\delta\alpha) + (\partial_{\bar{x}^2}\phi)(\partial_{\delta\alpha}\bar{x}^2)\big|_{\delta\alpha=0}(\delta\alpha) \\
&\quad + \ldots + (\partial_{\bar{x}^n}\phi)(\partial_{\delta\alpha}\bar{x}^n)\big|_{\delta\alpha=0}(\delta\alpha) \\
&= \phi(x^1,\ldots,x^n) + \delta\alpha\xi^1(\partial_{\bar{x}^1}\phi)\big|_{\delta\alpha=0} + \ldots + \delta\alpha\xi^n(\partial_{\bar{x}^1}\phi)\big|_{\delta\alpha=0} \\
&= \phi(x^1,\ldots,x^n) + \delta\alpha[\xi^1\partial_{x^1}\phi + \ldots + \xi^n\partial_{x^n}\phi)
\end{aligned} \tag{4}$$

and letting

$$U = \xi^i\partial_{x^i} \quad ; \qquad i = 1,\ldots,n \tag{5}$$

we have

$$(\bar{x}^1,\ldots,\bar{x}^n) = (1 + \delta\alpha U)\ \phi(x^1,\ldots x^n) \tag{6}$$

where U is termed the <u>generator</u> of a group of infinitesimal transformations.* Note that it is the local character of our group which enables us to simplify our expansion (4) by evaluating at $\delta\alpha=0$. The transformation of (3) can be represented by an operator $e^{\delta\alpha U}$ where U is defined in (5), then (6) becomes:

---

*For a more general treatment of the expansion, see Gilmore, pp.87-91.

$$\phi(\bar{x}^1,\ldots,\bar{x}^n) = e^{\delta\alpha U}\,\phi(x^1,\ldots,x^n)$$
$$= (1 + \delta\alpha U + O(\delta\alpha^2))\;\phi(x^1,\ldots,x^n) \qquad (7)$$
$$= (1 + \delta\alpha\xi^i\partial_{x^i})\;\phi(x^1,\ldots,x^n).$$

The composition law of such an exponential operator is

$$(e^{\delta\alpha_1 U})(e^{\delta\alpha_2 U})\;\phi(x^1,\ldots,x^n) = (1 + \delta\alpha_1 U)(1 + \delta\alpha_2 U)\;\phi(x^1,\ldots,x^n)$$
$$= (1 + \delta\alpha_1 U + \delta\alpha_2 U + \delta\alpha_1\delta\alpha_2 U^2)\;\phi(x^1,\ldots,x^n)$$
$$= (1 + \delta\alpha_1 U + \delta\alpha_2 U)\;\phi(x^1,\ldots,x^n) \qquad (7')$$
$$= (1 + (\delta\alpha_1 + \delta\alpha_2)U)\;\phi(x^1,\ldots,x^n)$$
$$= (1 + \delta\alpha_3 U)\;\phi(x^1,\ldots,x^n)$$

where $\delta\alpha_3 = \delta\alpha_1 + \delta\alpha_2$.

Thus we note that we have a continuous topological group, and furthermore the topological space is R^1 (a manifold) including the identity element $\delta\alpha_0 = 0$. This space is connected as well as simply connected, obeying the relations of (2) where the inverse is defined on the negative half of R^1. Thus, the topological space obeys all the criteria that defines a Lie group. The operator $e^{\delta\alpha U}$ is a local Lie group operator; and $e^{\alpha U}$ (for finite transformations) is a Lie group operator.

2-3. Local Lie Group Invariance Transformations

Equation (4) shows the effect of a one-parameter local Lie group transformation on an arbitrary function $\phi(x ,\ldots,x^n)$. If we were to consider only those transformations which leave the value of our function ϕ invariant, we can see right away that we must simply require that the second term of (4) vanish, i.e.

$$\delta\alpha(\xi^1\partial_{x^1}\phi + \ldots + \xi^n\partial_{x^n}\phi) = 0$$

$$\text{or} \qquad (8)$$

$$\xi^1\partial_{x^1}\phi + \ldots + \xi^n\partial_{x^n}\phi = U\phi = 0.$$

Thus, in words, we must require for the invariance of a function, that the generator of the transformation annihialate the function. The requirement for the invariance of equations is still less stringent than that for a function.

Consider a general equation

$$f(x^1,\ldots,x^n) = h(x^1,\ldots,x^n). \qquad (9)$$

Any such equation can be put into a more general form

$$F(x^1,\ldots,x^n) = 0 \qquad (10)$$

$$\text{where } F(x^1,\ldots,x^n) \equiv f(x^1,\ldots,x^n) - h(x^1,\ldots,x^n).$$

For invariance, it is obvious that we must require that

$$\bar{F}(x^1,\ldots,x^n) \equiv e^{\delta\alpha U}F(x^1,\ldots,x^n) = 0$$

$$\text{or}$$

$$0 = (1 + \delta\alpha U)\, F(x^1,\ldots,x^n) \qquad (11)$$

$$= F(x^1,\ldots,x^n) + \delta\alpha UF(x^1,\ldots,x^n).$$

Or more specifically, for invariance on the manifold defined by equation (10), this becomes

$$\begin{aligned} 0 &= F(x^1,\ldots,x^n)\big|_{(10)} + \delta\alpha UF(x^1,\ldots,x^n)\big|_{(10)} \\ &= \delta\alpha UF(x^1,\ldots,x^n)\big|_{(10)} \\ &= UF(x^1,\ldots,x^n)\big|_{(10)}. \end{aligned} \tag{12}$$

In words, we must require that the generator of the transformation annihialate $F(x^1,\ldots,x^n)$ subject to the condition that $F(x\ ,\ldots,x^n) = 0$.

2-4. A Generalization of Continuous Group Transformations

The transformations of the previous section exemplified a special class of transformations called point transformations; that is, those transformations whose generators are functions of the dependent and independent variables (and not their derivatives). In order to extend the theory of continuous invariance transformations to the invariance of differential equations, we must consider a more general class of transformations called contact transformations. The generators of these transformations are functions of derivatives as well as the independent and dependent variables. In foresight, we have for the generators of point transformations, $U = \xi^i(x^1,\ldots,x^n)\partial_{x^i}$, and for contact transformations, $U = \xi^i(x^1,\ldots,x^n,y^1,\ldots,y^n)\partial_{x^i} + \eta^i(x^1,\ldots,x^n,y^1,\ldots,y^n)\partial_{y^i}$, where the y^i may now be considered the "dependent" variables. In the following sections, we shall show explicitly how these generalizations are made and their applications to the invariance of ordinary differential equations.

2-4.1. Transformation of Infinitesimal Displacements*

In anticipation of applying this transformation theory to differential equations, we now consider the infinitesimal transformations of infinitesimal displacements.

Consider the transformation under $T(\alpha)$

$$\begin{aligned} x^r \to\to\to \bar{x}^r &= e^{\alpha U} x^r \\ &= [1 + \alpha\xi^i\partial_{x^i} + \tfrac{1}{2}\alpha^2(\xi^i)^2\partial_{x^i}\partial_{x^i} + \ldots +]\, x^r \qquad (1) \\ &= x^r + \alpha\xi^r \end{aligned}$$

*A good exposition on the transformation of infinitesimal displacements and contact transformations is given by Wulfman (32).

From (1) it follows that

$$d\bar{x}^r = \partial_{x^i}(x^r + \alpha\xi^r)dx^i = dx^r + \alpha\partial_{x^i}\xi^r dx^i; \qquad \text{(sum over i)} \tag{2}$$

and

$$d\bar{x}^r - dx^r = \alpha\partial_{x^i}\xi^r\, dx^i. \tag{3}$$

The effect of an infinitesimal transformation $T(\delta\alpha)$ on the infinitesimal displacement dx^r is obtained as $\alpha \to \delta\alpha$ in equation (3):

$$\begin{aligned} d\bar{x}^r - dx^r &= \delta\alpha\partial_{x^i}\xi^r\, dx^i \\ &= \delta\alpha(d\xi^r) \end{aligned} \tag{4}$$

where $d\xi^r = \partial_{x^i}\xi^r\, dx^i$.

We now define a generator U' which corresponds to the generator of transformations of infinitesimal displacements. Thus,

$$d\bar{x}^r = e^{\alpha U'} dx^r \tag{5}$$

$$\text{where } U' = U + d\xi^r\partial_{(dx^r)} = U + \dot{\xi}^r\partial_{(dx^r)} \quad ; \quad \dot{\xi}^r \equiv d\xi^r.$$

2-4.2 Contact Transformations

A contact transformation of the variables $x^j, x^\lambda, x^{\lambda j}$ leaves invariant the Pffafian form

$$dx^\lambda - \sum_j x^{\lambda j}dx^j = 0; \qquad j = 1,\ldots,n,\ \lambda > n \tag{6}$$

where we have separated the variables into "dependent" (x^j) and "independent" $(x^\lambda, x^{\lambda j})$ variables. If, however, a functional relation is established between the dependent and independent variables x^λ and x^j respectively, and dx^λ is established as a perfect differential, then it becomes apparent that the "dependent" variable $x^{\lambda j}$ takes on the familiar form of a partial derivative

$$x^{\lambda j} = \partial_{x^j} x^\lambda \equiv \frac{\partial x^\lambda}{\partial x^j} . \qquad (7)$$

With this in mind we turn our attention to this "dependent" variable ($x^{\lambda j}$) which, by equation (6), can in general only be said to be the ratio of infinitesimal displacements $\frac{dx^\lambda}{dx^j}$ when $dx^{j'} = 0$, ($j \neq j' = 1, \ldots, n$) and ask how this variable transforms. More specifically, we are asking, "What is the relation between the ξ^j's and the $\xi^{\lambda j}$'s of the transformation with generator

$$U' = \xi^j \partial_{x^j} + \xi^{\lambda j} \partial_{x^{\lambda j}} \quad . \qquad (8)$$

In considering the general transformation of equation (6) under $T(\delta\alpha)$, we have, according to equations (1) and (2):

$$\begin{aligned} x^{\lambda j} &\to\to\to \bar{x}^{\lambda j} = x^{\lambda j} + \delta\alpha\xi^{\lambda j} \\ dx^\lambda &\to\to\to d\bar{x}^\lambda = dx^\lambda + \delta\alpha\partial_{x^i}\xi^\lambda dx^i \qquad \text{(sum over i)} \qquad (9) \\ dx^j &\to\to\to d\bar{x}^j = dx^j + \delta\alpha\partial_{x^i}\xi^j dx^i . \end{aligned}$$

Requiring that $T(\delta\alpha)$ leave the Pfaffian form invariant, we have

$$dx^\lambda - \sum_j x^{\lambda j} dx^j = 0 \to\to\to d\bar{x}^\lambda - \sum_j \bar{x}^{\lambda j} d\bar{x}^j \Big|_{dx^\lambda = \sum_j x^{\lambda j} dx^j} = 0 \qquad (10)$$

where

$$\begin{aligned} d\bar{x}^\lambda - \sum_j \bar{x}^{\lambda j} d\bar{x}^j \Big| &= \sum_i (dx^\lambda + \delta\alpha\partial_{x^i}\xi^\lambda dx^i) - \sum_{ij}(x^{\lambda j} + \delta\alpha\xi^{\lambda j})(dx^j + \delta\alpha\partial_{x^i}\xi^j dx^i) \Big| \\ &= \sum_i (dx^\lambda + \delta\alpha\partial_{x^i}\xi^\lambda dx^i) - \sum_{ij}(x^{\lambda j} dx^j + \delta\alpha x^{\lambda j}\partial_{x^i}\xi^j dx^i + \delta\alpha dx^j \xi^{\lambda j} \\ &\qquad + (\delta\alpha)^2 \xi^{\lambda j}\partial_{x^i}\xi^j dx^i) \Big| \qquad (11) \\ &= \sum_{ij}[dx^\lambda + \delta\alpha\partial_{x^i}\xi^\lambda dx^i - dx^\lambda - \delta\alpha x^{\lambda j}\partial_{x^i}\xi^j dx^i - \delta\alpha dx^j \xi^{\lambda j}] \Big| \\ &= \delta\alpha\sum_{ij}[\partial_{x^i}\xi^\lambda dx^i - x^{\lambda j}\partial_{x^i}\xi^j dx^i - \xi^{\lambda j} dx^j] \Big| \end{aligned}$$

In terms of the original (untransformed) variables $dx^\lambda, x^{\lambda j}, dx^j$ then, the transformation takes the form.

$$dx^\lambda - \sum_j x^{\lambda j}dx^j = 0 \rightarrow\rightarrow\rightarrow \delta\alpha\sum_{ij}[\partial_{x^i}\xi^\lambda dx^i - x^{\lambda j}\partial_{x^i}\xi^j dx^i - \xi^{\lambda j}dx^j]\Big|_{(6)} = 0. \qquad (12)$$

Separating the x^i variables into dependent (x^k; $k>n$) and independent (x^ℓ; $\ell\leq n$) variables, the right-hand side of (12) becomes

$$\sum_{jk\ell}[\partial_{x^\ell}\xi^\lambda dx^\ell + \partial_{x^k}\xi^\lambda dx^k - x^{\lambda j}(\partial_{x^k}\xi^j dx^k + \partial_{x^\ell}\xi^j dx^\ell) - \xi^{\lambda j}dx^j]\Big| = 0 \qquad (13)$$

$$\sum_{jk\ell}[\partial_{x^\ell}\xi^\lambda dx^\ell + \partial_{x^k}\xi^\lambda dx^k - x^{\lambda j}(\partial_{x^k}\xi^j dx^k + \partial_{x^\ell}\xi^j dx^\ell) - \xi^{\lambda\ell}dx^\ell]\Big| = 0.$$

From (6), we have

$$dx^k = \sum_\ell x^{k\ell}dx^\ell. \qquad (14)$$

Substituting this into equation (13)

$$\begin{aligned}&\sum_{jk\ell}[\partial_{x^\ell}\xi^\lambda dx^\ell + \partial_{x^k}\xi^\lambda x^{k\ell}dx^\ell - x^{\lambda j}(\partial_{x^k}\xi^j x^{k\ell}dx^\ell + \partial_{x^\ell}\xi^j dx^\ell) - \xi^{\lambda\ell}dx^\ell]\Big| = 0\\ &\sum_{jk\ell}[\partial_{x^\ell}\xi^\lambda + \partial_{x^k}\xi^\lambda x^{k\ell} - x^{\lambda j}x^{k\ell}\partial_{x^k}\xi^j - x^{\lambda j}\partial_{x^\ell}\xi^j - \xi^{\lambda\ell}]dx^\ell = 0\ .\end{aligned} \qquad (15)$$

Enforcing the independence of dx^ℓ for all ℓ, we find that the expression within the parenthesis must vanish for each value of ℓ:

$$\sum_{jk}[\partial_{x^\ell}\xi^\lambda + \partial_{x^k}\xi^\lambda x^{k\ell} - x^{\lambda j}x^{k\ell}\partial_{x^k}\xi^j - x^{\lambda j}\partial_{x^\ell}\xi^j - \xi^{\lambda\ell}] = 0 \qquad (16)$$

or, solving for $\xi^{\lambda\ell}$,

$$\xi^{\lambda\ell} = \sum_{jk}[\partial_{x^\ell}\xi^\lambda + \partial_{x^k}\xi^\lambda x^{k\ell} - x^{\lambda j}x^{k\ell}\partial_{x^k}\xi^j - x^{\lambda j}\partial_{x^\ell}\xi^j]. \qquad (17)$$

In terms of λ and j, we have for each value of λ and j

$$\begin{aligned}\xi^{\lambda j} &= \sum_{k\ell}[\partial_{x^j}\xi^\lambda + \partial_{x^k}\xi^\lambda x^{kj} - x^{\lambda\ell}x^{kj}\partial_{x^k}\xi^\ell - x^{\lambda\ell}\partial_{x^j}\xi^\ell]\\ &= \sum_k[\partial_{x^j}\xi^\lambda + \partial_{x^k}\xi^\lambda x^{kj} - \sum_\ell x^{\lambda\ell}(\partial_{x^j}\xi^\ell + x^{kj}\partial_{x^k}\xi^\ell)]\ .\end{aligned} \qquad (18)$$

Defining an operator

$$D_j \equiv \partial_{x^j} + x^{kj}\partial_{x^k} \tag{19}$$

(18) becomes

$$\xi^{\lambda j} = \sum_k [D_j(\xi^\lambda) - \sum_\ell x^{k\ell} D_j(\xi^\ell)]. \tag{20}$$

Thus, the generators of contact transformations take on the form

$$U = \xi^j \partial_{x^j} + \xi^\lambda \partial_{x^\lambda} + \xi^{\lambda j}\partial_{x^{\lambda j}} \tag{21}$$

where the variables $x^j, x^\lambda, x^{\lambda j}$ are defined by the Pfaffian relation (6) and $\xi^{\lambda j}$ satisfies (20). For the restricted case in which ξ^j, ξ^λ and $\xi^{\lambda j}$ are functions of x^j and x^λ only, one refers to U above as being the first extension of the generator of a point transformation. The first extension is usually designated as $\hat{U}$ where

$$\hat{U} = U + \xi^{\lambda j}\partial_{x^{\lambda j}}. \tag{22}$$

2-5. One-Parameter Lie Groups Admitted by Ordinary Differential Equations

As shown in the last section, derivatives can be considered as special cases of ratios of infinitesimal displacements (when $dx^{j'} = 0$). In this section, we will consider the invariance properties of ordinary differential equations. This essentially means that we shall be dealing with Pfaffians of one independent variable (n=1). We shall suppose there can exist a functional relation between the "dependent" and "independent" variables for s "dependent" variables, i.e.

$$x^\lambda = f(x^1) \qquad \lambda = 2,\ldots,s \,. \tag{23}$$

Then the variables $x^{\lambda 1}$ are defined as total derivatives

$$x^{\lambda 1} \equiv \frac{dx^\lambda}{dx^1} \equiv x^\lambda_1 \,. \tag{24}$$

Extending this notation to the second and higher derivatives, we have

$$x^{\lambda 11} \equiv \frac{d^2x^\lambda}{d(x^1)^2} \equiv x^\lambda_{11}; \quad x^{\lambda 111} \equiv \frac{d^3x^\lambda}{d(x^1)^3} \equiv x^\lambda_{111}; \quad \ldots\ldots \tag{25}$$

We are now in a position to use Lie's theory on continuous groups of transformations in application to our studies of differential equations. We shall consider differential equations that can be reduced to "normal" form, giving a system of first order ordinary differential equations

$$\frac{dx_{m-1}}{dx^1} = x^\lambda_m = f^{\lambda m}(x^1, x^\lambda, x^\lambda_1, \ldots, x^\lambda_{m-1}) \qquad \text{for each } \lambda, m \tag{26}$$

where $x^\lambda, \ldots, x^\lambda_m$ are defined in equation (25). Then according to (21), the generator of the transformation takes on the form

$$U = \xi^1 \partial_{x^1} + \xi^\lambda \partial_{x^\lambda} + \xi^{\lambda 1} \partial_{x^{\lambda 1}} + \xi^{\lambda 11} \partial_{x^{\lambda 11}} + + + \tag{27}$$

and the invariance condition becomes

$$
\begin{aligned}
&U(x^{\lambda}_{m} - f^{\lambda m})\big|_{x^{\lambda}_{m}=f^{\lambda m}} = 0 \\
&\xi^{\lambda m} - (\xi^{1} f^{\lambda m}_{1} + \xi^{\lambda} f^{\lambda m}_{\lambda} + \xi^{\lambda j} f^{\lambda m}_{\lambda j})\big|_{x^{\lambda}_{m}=f^{\lambda m}} = 0 \\
&\text{where } f^{\lambda m}_{i} \equiv \partial_{x^{i}} f^{\lambda m} .
\end{aligned}
\tag{28}
$$

From equation (20), (28) becomes

$$
\sum_{k=2}^{\infty} [D_{m}(\xi^{\lambda}) - x^{k1} D_{m}(\xi^{1})] - \{\xi^{1} f^{\lambda m}_{1} + \xi^{\lambda} f^{\lambda m}_{\lambda} + \sum_{k'=2}^{\infty} [D_{j}(\xi^{\lambda}) - x^{k'1} D_{j}(\xi^{1})]\}\big|_{(26)} = 0
$$

$$
\begin{aligned}
&\sum_{k}^{\infty} [\xi^{\lambda}_{m} + x^{km} \xi^{1}_{k} - x^{k1}(\xi^{1}_{m} + x^{km} \xi^{1}_{k})] - \{\xi^{1} f^{\lambda m}_{1} + \xi^{\lambda} f^{\lambda m}_{\lambda} + \\
&\qquad \sum_{k'} [\xi^{\lambda}_{j} + x^{k'j} \xi^{\lambda}_{k'} - x^{k'1}(\xi^{1}_{j} + x^{k'j} \xi^{1}_{k'})]\}\big|_{(26)} = 0.
\end{aligned}
\tag{29}
$$

Changing dummy indices where k'=k and making the substitution

$$
x^{ki} = x^{k}_{i} = f^{ki} \tag{30}
$$

as expressed by the condition, we have

$$
\begin{aligned}
&\sum_{k} \{\xi^{\lambda}_{m} + f^{km} \xi^{1}_{k} - f^{k1}(\xi^{1}_{m} + f^{km} \xi^{1}_{k}) - \xi^{1} f^{\lambda m}_{1} - \xi^{\lambda} f^{\lambda m} \\
&\qquad - \sum_{j=1}^{m} [\xi^{\lambda}_{j} + f^{kj} \xi^{\lambda}_{k} - f^{k1}(\xi^{1}_{j} + f^{kj} \xi^{1}_{k})]\}\big| = 0.
\end{aligned}
\tag{31}
$$

2-6. Many-Parameter Groups

The previous discussions have focused on the requirements for one-parameter groups of invariance transformations. Together with the Lie commutator which defines the group multiplication, these one-parameter groups provide the framework for a Lie algebra -- an algebra of a many-parameter group. These groups may be of either finite or infinite dimension. If these groups are of infinite dimension, a subset of these may usually be found to close. Through a study of the commutation relations and its corresponding structure constants, much information can be obtained about the abstract group and its composition function (see section 2-1). Furthermore, the parameter space can then be used to fully characterize the global Lie group.

III. Invariance Properties of Hamiltonian Equations

3-1. Canonical Transformations and Constants of the Motion

Canonical transformations can be formulated in Lie theory as those continuous groups of transformations which leave the Poisson bracket relations invariant, i.e.

$$\{q^i,q^j\} = \{p^i,p^j\} = 0 \qquad i,j,k = 1,\ldots,n \tag{1}$$
$$\{p^i,q^j\} = \delta^{i,j}$$
$$\text{where } \{F,G\} = \frac{\partial F}{\partial p^k}\frac{\partial G}{\partial q^k} - \frac{\partial F}{\partial q^k}\frac{\partial G}{\partial p^k}\ .$$

Thus, under the infinitesimal transformation $T(\delta\alpha)$ with operator $e^{\delta\alpha U}$, we have, to first order in $\delta\alpha$,

$$q^i \to\to\to \bar{q}^i = q^i + \delta\alpha Q^i \tag{2}$$
$$p^i \to\to\to \bar{p}^i = p^i + \delta\alpha P^i$$
$$\text{where } U = Q^i\partial_{q^i} + P^i\partial_{p^i}\ .$$

We require for invariance

$$U\{q^i,q^j\}\big|_{\{q^i,q^j\}=0} = 0 \tag{3a}$$
$$U\{p^i,p^j\}\big|_{\{p^i,p^j\}=0} = 0 \tag{3b}$$
$$U[\{p^i,q^j\} - \delta^{i,j}]\big|_{\{p^i,q^j\}=\delta^{ij}} = 0 \tag{3c}$$

or

$$\{Uq^i,q^j\} + \{q^i,Uq^j\}\big| = 0$$
$$\{Q^i,q^j\} + \{q^i,Q^j\}\big| = 0 \tag{4a}$$
$$\partial_{p^j}Q^i - \partial_{p^i}Q^j = 0\ ;$$

$$\{Up^i, p^j\} + \{p^i, Up^j\}| = 0$$
$$\{P^i, p^j\} + \{p^i, P^j\}| = 0 \qquad (4b)$$
$$-\partial_{q^j} P^i + \partial_{q^i} P^j = 0 \ ;$$

$$\{Up^i, q^j\} + \{p^i, Uq^j\}| = 0$$
$$\{P^i, q^j\} + \{p^i, Q^j\}| = 0 \qquad (4c)$$
$$\partial_{p^j} P^i + \partial_{q^i} Q^j = 0 \ .$$

Each canonical transformation is derivable from a "characteristic function" $G(q,p)$ such that

$$Q^i = \partial_{p^i} G$$
$$P^i = -\partial_{q^i} G \ . \qquad (5)$$

In terms of G, U takes the form

$$U = \{G\cdot\ \} = \partial_{p^i} G \partial_{q^i} - \partial_{q^i} G \partial_{p^i} \ . \qquad (6)$$

Note that for the special case where the characteristic function is the Hamiltonian of the dynamical system, we have for the generator (V) of the canonical transformation

$$V = \{H\cdot\ \} = \partial_{p^i} H \partial_{q^i} - \partial_{q^i} H \partial_{p^i} \qquad (7)$$
$$= \dot{q}^i \partial_{q^i} + \dot{p}^i \partial_{p^i}$$
$$\text{where } \dot{q}^i = \frac{dq^i}{dt} \ ; \quad \dot{p}^i = \frac{dp^i}{dt} \ .$$

The group operator associated with the transformation becomes

$$T(\delta t) = e^{\delta t V} = \exp[\delta t(\dot{q}^i \partial_{q^i} + \dot{p}^i \partial_{p^i})] \qquad (8)$$

where the time (t) takes on the role as the parameter of the continuous transformation. Operating on p^i and q^i, we have

$$e^{\delta tV}q^i = \bar{q}^i = q^i + \delta t(\dot{q}^i\partial_{q^i} + \dot{p}^i\partial_{p^i})q^i$$
$$= q^i + \delta t\dot{q}^i \tag{9a}$$

$$e^{\delta tV}p^i = \bar{p}^i = p^i + \delta t(\dot{q}^i\partial_{q^i} + \dot{p}^i\partial_{p^i})\, p^i$$
$$= p^i + \delta t\dot{p}^i \,. \tag{9b}$$

Thus, it can be seen that the effect of the transformation is to move the points (q^i,p^i) along its trajectory in phase space. For this reason, the operator is termed the "evolution operator", and its generator V, the "evolution generator". Furthermore, an invariant function of the evolution transformation can be seen to be a "first integral" or "constant of the motion". The requirement for a constant of the motion $\kappa(q^i,p^i)$ then, is

$$e^{\delta tV}\kappa(q^i,p^i) = \bar{\kappa}(\bar{q}^i,\bar{p}^i) = \kappa(q^i,p^i) \tag{10}$$

or, more specifically, for a given manifold upon which our "flow" in phase space is defined, we may require invariance of the function on that particular manifold. Then (10) becomes

$$e^{\delta tV}\kappa(q^i,p^i)\big|_{W=0} = \kappa(q^i,p^i)\big|_{W=0} \tag{10'}$$

or

$$\kappa(q^i,p^i) + \delta tV\kappa(q^i,p^i)\big| = \kappa(q^i,p^i)\big|$$
$$0 = V\kappa(q^i,p^i)\big|. \tag{11}$$

This can be seen to be in full agreement with the classical theory of Hamiltonian mechanics, where (11) can be rewritten as

$$\{H\cdot\ \}\kappa(q^i,p^i)\big|_{W=0} = 0$$
$$\{H,\kappa\}\big| = \frac{d\kappa}{dt}\Big| = 0 \,. \tag{12}$$

3-2. Continuous Group Transformations Admitted by Hamilton's Equations

When $\dot{p}_i = \{H, q_i\}$ and $\dot{q}_i = \{H, p_i\}$ are Hamilton's equations, they define a flow characterized by the Hamiltonian $H(q^i, p^i)$ (the characteristic function of the evolution generator) in the phase space. In order to find the groups of continuous contact transformations which are admitted by the flow, we must require that $\hat{U}$ annihilate the equations of motion and the Hamiltonian which defines the space (see fig.1)

$$\hat{U}[\dot{q}^i - \{H, q^i\}]\big|_{\dot{q}_i = \{H, q^i\}} = 0 \tag{13a}$$

$$\hat{U}[\dot{p}^i - \{H, p^i\}]\big|_{\dot{p}^i = \{H, p^i\}} = 0 \tag{13b}$$

$$UH(q^i, p^i)\big|_{W=0} = 0 \tag{13c}$$

where $\hat{U} = Q^i \partial_{q^i} + P^i \partial_{p^i} + \hat{Q}^i \partial_{q^i} + \hat{P}^i \partial_{p^i}$;

and we have for (13) in terms of Q^i, P^i

$$\begin{aligned}
&\hat{Q}^i - \hat{U}(\partial_{p^i} H)\big| = 0 \\
&D_t Q^i - Q^k \partial_{q^k} \partial_{p^i}(H) - P^k \partial_{p^k} \partial_{p^i}(H)\big| = 0 \\
&\dot{q}^k \partial_{q^k} Q^i + \dot{p}^k \partial_{p^k} Q^i - \partial_{p^i}(Q^k \partial_{q^k} H + P^k \partial_{p^k} H) + \partial_{p^i} Q^k \partial_{q^k} H + \partial_{p^i} Q^k \partial_{p^k} H\big| = 0 \\
&(\partial_{q^k} Q^i + \partial_{p^i} P^k)\partial_{p^k} H - (\partial_{p^k} Q^i - \partial_{p^i} Q^k)\partial_{q^k} H - \partial_{p^i}(UH)\big| = 0
\end{aligned} \tag{14a}$$

$$\begin{aligned}
&\hat{P}^i + \hat{U}(\partial_{q^i} H)\big| = 0 \\
&D_t P^i + Q^k \partial_{q^k} \partial_{q^i}(H) + P^k \partial_{p^k} \partial_{q^i}(H)\big| = 0 \\
&\dot{q}^k \partial_{q^k} P^i + \dot{p}^k \partial_{p^k} P^i + \partial_{q^i}(Q^k \partial_{q^k} H + P^k \partial_{p^k} H) - \partial_{q^i} Q^k \partial_{q^k} H - \partial_{q^i} P^k \partial_{p^k} H\big| = 0 \\
&(\partial_{q^k} P^i - \partial_{q^i} P^k)\partial_{p^k} H - (\partial_{p^k} P^i + \partial_{q^i} Q^k)\partial_{q^k} H + \partial_{q^i}(UH)\big| = 0
\end{aligned} \tag{14b}$$

$$Q^k \partial_{q^k} H + P^k \partial_{p^k} H\big| = 0. \tag{14c}$$

Thus, the conditions for the most general continuous groups of contact transformations admitted by Hamilton's equations on the manifold defined by W=0 are given by (14).

To find the canonical transformations, a subset of the most general contact transformations admitted by Hamilton's equations, we must enforce the invariance of the Poisson bracket relations on equation (14) as given by equation (4). Thus, we have for the invariance conditions of the continuous groups of canonical transformations:

$$(\partial_{q^k}Q^i + \partial_{p^i}P^k)\partial_{p^k}H - (\partial_{p^k}Q^i - \partial_{p^i}Q^k)\partial_{q^k}H - \partial_{p^i}(UH)\big|_{\dot{q}^i=\partial_{p^i}H} = 0 \quad (15a)$$

$$(\partial_{q^k}P^i - \partial_{q^i}P^k)\partial_{p^k}H - (\partial_{p^k}P^i + \partial_{q^i}Q^k)\partial_{q^k}H + \partial_{q^i}(UH)\big|_{\dot{p}^i=-\partial_{q^i}H} = 0 \quad (15b)$$

$$UH\big|_{W=0} = Q^k\partial_{q^k}H + P^k\partial_{p^k}H\big|_{W=0} = 0 \quad (15c)$$

$$\partial_{p^k}Q^i - \partial_{p^i}Q^k\big|_{\{q^i,q^k\}=0} = 0 \quad (15d)$$

$$-\partial_{q^k}P^i + \partial_{q^i}P^k\big|_{\{p^i,p^k\}=0} = 0 \quad (15e)$$

$$\partial_{p^k}P^i + \partial_{q^i}Q^k\big|_{\{p^i,q^k\}=\delta^{ik}} = 0\,. \quad (15f)$$

The effect of the canonical requirements (d-e) is to require that each term in the parenthesis in equations (a-c) vanish independently. More explicitly, they establish a canonical relationship between equations (15a) and (15b), and consequently, they express an independence of the canonical variables $\dot{q}^k$ and $\dot{p}^k$ as can be seen if we rewrite these as

$$(\partial_{q^k}Q^i + \partial_{p^i}P^k)\dot{q}^k + (\partial_{p^k}Q^i - \partial_{p^i}Q^k)\dot{p}^k - \partial_{p^i}(UH)\big| = 0 \quad (16a)$$

$$(\partial_{q^k}P^i - \partial_{q^i}P^k)\dot{q}^k + (\partial_{p^k}P^i + \partial_{q^i}Q^k)\dot{p}^k + \partial_{q^i}(UH)\big| = 0\,. \quad (16b)$$

Furthermore, at first glance, it appears as though the last terms in the equations above vanish trivially as a consequence of equation (15c); however, as was shown by Wulfman and Sumi,* these terms are physically significant as they express a stability condition for the system. It follows from this, that

*Wulfman, Sumi, "New, Locally Stable, Symmetries of Keplerian Systems", Atomic Scattering Theory, edited by J. Nuttall, pp. 197.

THEOREM:

The generator of a canonical or M-canonical (where M defines the manifold of flow) transformation that leaves invariant Hamilton's equations of motion on M is locally stable about M.

Thus we have, for a transformation to be canonical and leave invariant Hamilton's equations:

$$\partial_{q^k} Q^i + \partial_{p^i} P^k \Big|_{\{q^i,H\};\{p^i,H\}} = 0 \tag{17a}$$

$$\partial_{q^k} P^i - \partial_{q^i} P^k \Big|_{\{q^i,H\};\{p^i,H\}} = 0 \tag{17b}$$

$$\partial_{p^k} Q^i - \partial_{p^i} Q^k \Big|_{\{q^i,H\};\{p^i,H\}} = 0 \tag{17c}$$

$$\partial_{q^i}(UH) \Big|_{\{q^i,H\};\{p^i,H\}} = 0 \tag{17d}$$

$$\partial_{p^i}(UH) \Big|_{\{q^i,H\};\{p^i,H\}} = 0 \tag{17e}$$

$$UH\Big|_{W=0} = 0 \; . \tag{17f}$$

3-3. Dynamical Degeneracy and Dynamical Groups

To obtain the many-parameter Lie groups which are admitted by a Hamiltonian system of given energy E = constant (dynamical degeneracy group), the transformations must be required to leave invariant the manifold of constant energy $H(q^i,p^i) = E$. We define a new function we shall call the "extended Hamiltonian"

$$W(q^i,p^i,E,t) = H(q^i,p^i,t) - E. \tag{18}$$

Then the manifold of constant energy is defined by

$$W(q^i,p^i,t) = 0 \tag{19}$$

for a fixed value of E. The conditions for invariance are obtained from (14) by replacing the Hamiltonian with the "extended Hamiltonian" and requiring the annihilation of the expression only on the manifold $W \doteq 0$. Then (14) becomes

$$(\partial_{q^k}Q^i + \partial_{p^i}P^k)\partial_{p^k}W - (\partial_{p^k}Q^i - \partial_{p^i}Q^k)\partial_{q^k}W - \partial_{p^i}(UW)\big|_{\{q^i,W\}=0;W=0} = 0 \tag{20a}$$

$$(\partial_{q^k}P^k - \partial_{q^i}P^k)\partial_{p^k}W - (\partial_{p^k}P^i + \partial_{q^i}Q^k)\partial_{q^k}W + \partial_{q^i}(UW)\big|_{\{p^i,W\}=0;W=0} = 0 \tag{20b}$$

$$UW\big|_{W=0} = 0 \tag{20c}$$

and similarly for the canonical transformations of equation (17).

For the dynamical groups of continuous transformations, equation (20) must be modified to include the energy (E) and time (t) as dynamical variables, and a new parameter (τ) introduced to parameterize the evolution group. Then in equations (20), one may simply redefine the phase space to encompass the time and energy (termed the "extended phase space") and allow the indices to run from 1 to $n+1$, where

$$q^{n+1} \equiv t;\ p^{n+1} \equiv E. \tag{21}$$

IV. Program DETERMININGEQNS

DETERMININGEQNS is a computer program written by the author in Pascal. This program will approximate, through a power series expansion in n variables, the generators $U(z^1,\ldots,z^n)$ of the contact transformations that leave invariant a given system of autonomous ordinary differential equations as well as arbitrary functions of the n variables expressed in a power series expansion. As such, DETERMININGEQNS has the flexibility to handle either Hamiltonian or non-Hamiltonian systems, with or without conditions of constraint. Furthermore, it is capable of obtaining such generators which are invariance generators only upon evaluation on the manifolds M^i of constraint, i.e.

$$U(z^1,\ldots,z^n)[\dot{z}^i - g(z^1,\ldots,z^n)]|_{M^k} = 0$$

and

$$U(z^1,\ldots,z^n)\ F^j(z^1,\ldots,z^n)|_{M^k} = 0$$

for $i = 1,\ldots,n$; $j = 0,\ldots,f$; $k = 1,\ldots,m$.

4-1. Systematic Derivation of the Determining Equations

Given a system of ordinary differential equations

$$\frac{dx^i}{d\omega} = f^i(x^1,\ldots,x^n,\omega); \qquad i = 1,\ldots,n \tag{1}$$

where ω is the parameter of the evolution group operator, we seek to find the generators U of continuous groups of contact transformations which leave this system of ordinary differential equations invariant. In order to simplify the notation, we introduce the new variables

$$\dot{x}^i = \frac{dx^i}{d\omega}. \tag{2}$$

Then (1) becomes

$$\dot{x}^i = f^i(x^1,\ldots,x^n,\ ). \tag{3}$$

Since U is the generator of contact transformations, U takes the form

$$U = \Omega(\omega,x\ ,\ldots,x^n)\partial_\omega + \xi^j(\omega,x^1,\ldots,x^n)\partial_{x^j} \tag{4}$$

and its extension is

$$\hat{U} = U + \hat{\xi}^j(\omega,x^1,\ldots,x^n)\partial_{\dot{x}^j} \tag{5}$$

where

$$\hat{\xi}^j = D(\xi^j) - \dot{x}^j D(\Omega); \qquad D = \partial_\omega + \dot{x}^\ell \partial_{x^\ell}. \tag{6}$$

For invariance, we require

$$\hat{U}[\dot{x}^i - f^i(\omega,x^1,\ldots,x^n)]\big|_{\dot{x}^\ell = f^\ell} = 0 \tag{7}$$

or

$$\hat{\xi}^i - \Omega\partial_\omega f^i - \xi^j\partial_{x^j} f^i \Big| = 0 \tag{8}$$

$$\partial_\omega \xi^i + x^j\partial_{x^j}\xi^i - \dot{x}^i(\partial_\omega\Omega + x^j\partial_{x^j}\Omega) - \Omega\partial_\omega f^i - \xi^j\partial_{x^j} f^i \Big| = 0 \tag{9}$$

However, for our purposes, we shall be interested only in autonomous ordinary differential equations and we will not consider transformations of our parameter ω. This restriction implies

$$\Omega = 0;\ f = f(x^1,\ldots,x^n);\ \xi = \xi(x^1,\ldots,x^n) \tag{10}$$

and our determining equations become

$$\dot{x}^j\partial_{x^j}\xi^i - \xi^j\partial_{x^j} f^i \Big| = 0 \tag{11}$$

that is,

$$f^j\partial_{x^j}\xi^i - \xi^j\partial_{x^j} f^i = 0. \tag{12}$$

If we define our evolution generator

$$V = \dot{x}^j\partial_{x^i} = f^j\partial_{x^j} \tag{13}$$

then, from (4), (10), and (13), equation (12) becomes

$$[U,V] = 0. \tag{14}$$

Thus, the invariance generators are simply those generators which commute with the evolution operator.

DETERMININGEQNS utilizes equations (12). The generators U are approximated by a power series expansion

$$U = \xi^i\partial_{x^i}\ ;\quad \xi^i = a^i_{j^1\ldots j^n}(x^1)^{j^1}\cdot(x^2)^{j^2}\cdots(x^n)^{j^n} \tag{15}$$

such that the determining equations become

$$f^j \partial_{x^j}[a^i_{k^1\ldots k^n}(x^1)^{k^1}\cdots(x^n)^{k^n}] - [a^j_{k^1\ldots k^n}(x^1)^{k^1}\cdots(x^n)^{k^n}]\partial_{x^j}f^i = 0. \quad (16)$$

The functions f^i can also be written as a power series

$$f^i = c^i_{j^1\ldots j^n}(x^1)^{j^1}\cdots(x^n)^{j^n} \quad (17)$$

Substituting (17) into (16), we have

$$c^j_{\ell^1\ldots\ell^n}(x^1)^{\ell^1}\cdots(x^n)^{\ell^n}\partial_{x^j}[a^i_{k^1\ldots k^n}(x^1)^{k^1}\cdots(x^n)^{k^n}]$$
$$- a^j_{k^1\ldots k^n}(x^1)^{k^1}\cdots(x^n)^{k^n}\partial_{x^j}[c^i_{\ell^1\ldots\ell^n}(x^1)^{\ell^1}\cdots(x^n)^{\ell^n}] = 0 \quad (18)$$

$$c^j_{\ell^1\ldots\ell^n}\cdot a^i_{k^1\ldots k^n}\cdot k^j\cdot(x^1)^{k^1+\ell^1}\cdots(x^j)^{k^j+\ell^j-1}\cdots(x^n)^{k^n+\ell^n}$$
$$- c^i_{\ell^1\ldots\ell^n}\cdot a^j_{k^1\ldots k^n}\cdot\ell^j\cdot(x^1)^{k^1+\ell^1}\cdots(x^j)^{k^j+\ell^j-1}\cdots(x^n)^{k^n+\ell^n} = 0. \quad (19)$$

Collecting terms, we have:

$$(k^j c^j_{\ell^1\ldots\ell^n}a^i_{k^1\ldots k^n} - \ell^j c^i_{\ell^1\ldots\ell^n}a^j_{k^1\ldots k^n})(x^1)^{k^1+\ell^1}\cdots(x^j)^{k^j+\ell^j-1}$$
$$\cdots(x^n)^{k^n+\ell^n} = 0\ . \quad (20)$$

Changing indices, let $p^s = k^s + \ell^s$, then (20) becomes

$$[k^j c^j_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^i_{k^1\ldots k^n} - (p^j - k^j)\cdot c^i_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^j_{k^1\ldots k^n}$$
$$\times [(x^1)^{p^1}\cdots(x^j)^{p^j-1}\cdots(x^n)^{p^n}] = 0 \quad (21)$$

therefore, require

$$k^j c^j_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^i_{k^1\ldots k^n} - (p^j - k^j)\cdot c^i_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^j_{k^1\ldots k^n} = 0. \quad (22)$$

Thus, we have a system of linear equations in the variables $a^i_{k^1\ldots k^n}$ which we solve using a standard linear equation solving program.

4-2.1. Invariance of Differential Equations Defined on a Manifold

Incorporated in the program is an option to impose further restrictions on the generators by requiring that the transformations they generate leave certain manifolds of constraint invariant. Thus, for the manifolds defined by $W^i(\vec{x}) = 0$, one may require

$$UW^i = 0 \tag{23}$$

where

$$W^i = b^i_{j^1 \ldots j^n}(x^1)^{j^1} \cdots (x^n)^{j^n} \tag{24}$$

and (23) becomes

$$a^k_{\ell^1 \ldots \ell^n} \cdot (x^1)^{\ell^1} \cdots (x^n)^{\ell^n} \partial_{x^k} \cdot [b^i_{j^1 \ldots j^n}(x^1)^{j^1} \cdots (x^n)^{j^n}] = 0 \, . \tag{25}$$

Collecting terms,

$$a^k_{\ell^1 \ldots \ell^n}(x^1)^{\ell^1} \cdots (x^n)^{\ell^n} (j^k b^i_{j^1 \ldots j^n})(x^1)^{j^1} \cdots (x^k)^{j^k - 1} \cdots (x^n)^{j^n} = 0 \tag{26}$$

$$j^k b^i_{j^1 \ldots j^n} \cdot a^k_{\ell^1 \ldots \ell^n} \cdot (x^1)^{\ell^1 + j^1} \cdots (x^k)^{\ell^k + j^k - 1} \cdots (x^n)^{\ell^n + j^n} = 0. \tag{27}$$

Changing indices, let $\ell^s + j^s = p^s$; $\ell^s = p^s - j^s$

$$j^k b^i_{j^1 \ldots j^n} \cdot a^k_{(p^1 - j^1) \ldots (p^n - j^n)} \cdot (x^1)^{p^1} \cdots (x^k)^{p^1 - 1} \cdots (x^n)^{p^n} = 0. \tag{28}$$

Therefore, require

$$j^k b^i_{j^1 \ldots j^n} \cdot a^k_{(p^1 - j^1) \ldots (p^n - j^n)} = 0 \, . \tag{29}$$

To find a larger group of generators which leave a given system invariant, we can generalize our definition of invariance such that it is subject to the evaluation on the manifold, i.e., require

$$[U,V]\big|_{W^i=0} = 0 \tag{30}$$

and

$$UW\big|_{W^i=0} = 0 \ . \tag{31}$$

This "conditional" invariance is a further generalization on equations (21) and (28). Here, we make the substitution of variables as dictated by $W = 0$. Thus, we have for any non-zero $c^i_{r^1 \ldots r^n}$

$$c^i_{j^1 \ldots j^n}(x^1)^{j^1}\cdots(x^n)^{j^n} = 0 \tag{32}$$

$$c^i_{r^1 \ldots r^n}(x^1)^{r^1}\cdots(x^n)^{r^n} = -\,c^i_{j^{1\prime}\ldots j^{n\prime}}(x^1)^{j^{1\prime}}\cdots(x^n)^{j^{n\prime}} \tag{33}$$

$$(x^1)^{r^1}\cdots(x^n)^{r^n} = -\frac{c^i_{j^{1\prime}\ldots j^{n\prime}}}{c^i_{r^1 \ldots r^n}}(x^1)^{j^{1\prime}}\cdots(x^n)^{j^{n\prime}} \ . \tag{34}$$

Define $\alpha(j,k,a)$ and $\beta(j,k,a)$ such that

$$\alpha^i_{p^1\ldots p^j-1\ldots p^n} \equiv k^j c^j_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^i_{k^1\ldots k^n} - (p^j-k^j) \times c^i_{(p^1-k^1)\ldots(p^n-k^n)}\cdot a^j_{k^1\ldots k^n} \tag{35}$$

$$\beta^i_{p^1\ldots p^j-1\ldots p^n} \equiv j^k b^i_{j^1\ldots j^n}\cdot a^k_{(p^1-j^1)\ldots(p^n-j^n)} \ ; \tag{36}$$

Then equations (21) and (28) can be rewritten

$$\alpha^i_{p^1\ldots p^j-1\ldots p^n}\cdot(x\ )^{p^1}\cdots(x^j)^{p^j-1}\cdots(x^n)^{p^n} = 0 \tag{37}$$

$$\beta^i_{p^1\ldots p^j-1\ldots p^n}\cdot(x^1)^{p^1}\cdots(x^j)^{p^j-1}\cdots(x^n)^{p^n} = 0. \tag{38}$$

Substituting (34) into (37) and (38), we have:

$$\left[-\alpha^i_{r^1\ldots r^n}\times\frac{c^i_{\ell^1\ldots\ell^n}}{c^i_{r^1\ldots r^n}} + \alpha^i_{\ell^1\ldots\ell^n}\right]\cdot(x^1)^{\ell^1}\cdots(x^n)^{\ell^n} = 0 \tag{39}$$

$$\left[-\beta^i_{r^1\ldots r^n}\times\frac{c^i_{\ell^1\ldots\ell^n}}{c^i_{r^1\ldots r^n}} + \beta^i_{\ell^1\ldots\ell^n}\right]\cdot(x^1)^{\ell^1}\cdots(x^n)^{\ell^n} = 0 \tag{40}$$

Thus, the full system of equations which determine the generators which leave the system and a given manifold of constraint invariant, subject to the evaluation on the manifold, are given in equations (39) and (40).

4-3. Technical Aspects of DETERMININGEQNS

One goal of this research was to develop a reasonably efficient computer program which could obtain (set up) and solve the determining equations for a given set of ordinary differential equations. This essentially meant that the program must use equation (11) of the previous section to explicitly derive and solve the determining equations.

It was apparent then, that what was needed was a symbol manipulation program which could perform the basic operations of addition, subtraction, multiplication, and differentiation of polynomials. In addition, this program must also be capable of collecting terms on common variables.

It must be recognized that equation (16) contains three different types of data structures as can be seen in equation (18):

(i) real number coefficients--- $c^j_{\ell_1 \ldots \ell_n}$

(ii) variable coefficients--- $a^i_{k_1 \ldots k_n}$

(iii) dynamical variables--- x_i.

We were thus faced with the problem of finding a computer data structure which could satisfactorily manipulate these coefficients and variables as cohesive units (which we will label as "terms") in an algebraic expression. Such a data structure was found in PASCAL. This language was very easy to learn and contained data structures called "records." The primary advantage of the record structure is the efficiency with which it can be manipulated.

4-3.1 Representation of Polynomials

The terms of a polynomial are represented by records. These records are divided into 4 "fields" as represented in the diagram below:

I	II	III	IV	→ → → → →	I'	II'	III'	IV'	→ → → • • •

I: real coefficient field $(c^i_{\ell_1 \ldots \ell_n})$

II: variable coefficient field $(a^i_{\ell_1 \ldots \ell_n})$

III: dynamical variable field $(x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n})$

IV: pointer address field.

The first three fields identify the coefficients and dynamical variables of the term. The fourth field "points" to the next term in the polynomial (i.e., contains the address of the term which is to follow). In this way, an ordered chain of records represents an ordered polynomial. As an example, the polynomial

$$g(x_1,x_2,x_3,x_4) = 5a^1_{0001}x_1x_2x_3x_4 + 2a^2_{1010}x_1 - 3a^4_{0010}x_1x_2 \qquad (1)$$

would be represented as the string of records:

1					2					3				
5.0	a^1_{0001}	$x_1^2x_2^3x_3x_4$	2	→	2.0	a^2_{1010}	x_1	3	→	-3.0	a^4_{0010}	$x_1^2x_2^2$	▨	(2)

It will be noted that the variables in these terms can be uniquely identified by specifying their indices. The polynomial can then be represented by numbers such as

5.0	1.0001	2311		→	2.0	2.1010	1000		→	-3.0	4.0010	2200	▨	(3)

Here, the variable a^i_{jklm} is represented by a real number whose integral digit represents the superscript (i) of "a" and whose decimal digits represent the subscripts $(k_1k_2k_3k_4)$ of "a". The dynamical variable

$x_1^{j_1}x_2^{j_2}x_3^{j_3}x_4^{j_4}$ is represented by an integral number of 4 digits. The i^{th} digit of the integer specifies the exponent of the corresponding i^{th} variable x_i, i.e.

$$a^{i}_{k_1k_2k_3k_4} = i.k_1k_2k_3k_4$$
$$x_1^{j_1}x_2^{j_2}x_3^{j_3}x_4^{j_4} = j_1j_2j_3j_4 \qquad (3')$$

With this representation we are now in a position to manipulate large ~~polynomials in a compact scheme which requires relatively little space~~ (4 words per term).

4-3.2 <u>Algorithms of Operations</u>

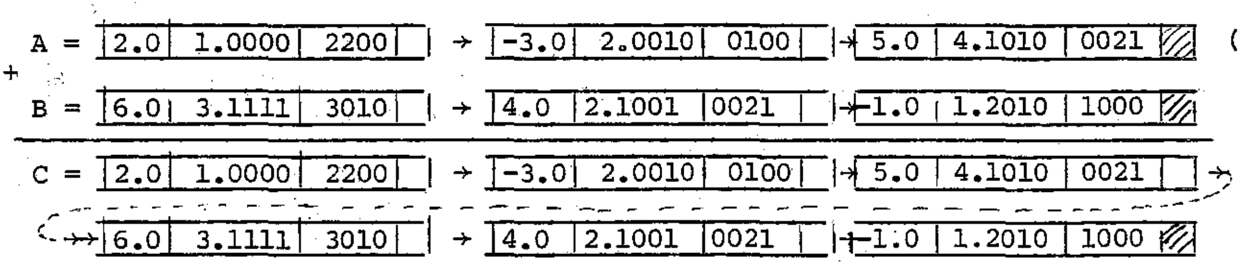
(i) Addition of Polynomials

In our representation, the addition of two polynomials is trivial. It simply requires the redirecting of the pointer of the last term in the first polynomial such that it "points" to the first term of the second polynomial. For example, consider the addition of A + B = C where A,B,C, are polynomials

$$A = 2a^1_{0000}x_1^2x_2^2 - 3a^2_{0010}x_2 + 5a^4_{1010}x_3^2x_4 \qquad (4)$$

$$B = 6a^3_{1111}x_1^3x_3 + 4a^2_{1001}x_3^2x_4 - a^1_{2010}x_1$$

In our representation, we have:



(5)

To complete the addition, we must collect terms on common dynamical variables. This procedure of "collecting terms" is simply a sorting procedure which sorts the polynomial, compares for "like" variables coefficients in terms of "like" dynamical variables, and adds the real coefficient if such "common terms" are found. Thus, in the last example, C becomes:

C = | 5.0 | 4.1010 | 0021 | | → | 4.0 | 2.1001 | 0021 | | → | -3.0 | 2.0010 | 0100 | | → (6)

→ → | -1.0 | 1.2010 | 1000 | | → | 2.0 | 1.0000 | 2200 | | → | 6.0 | 3.1111 | 3010 | ▨

$$C = (5a^4_{1010} + 4a^2_{1001})x_3^2x_4 - 3a^2_{0010}x_2 - a^1_{2010}x_1 + 2a^1_{0000}x_1^2x_2^2 + 6a^3_{1111}x_1^3x_3$$

(ii) Subtraction of Polynomials

The operation of subtraction requires that the coefficients of the second polynomial be multiplied by (-1) before addition.

(iii) Multiplication of Polynomials

To multiply two polynomials (A × B) we follow the standard multiplication scheme, taking each term in polynomial A and multiplying it by each term in polynomial B, and finally, collecting terms.

In our representation, the multiplication of two terms requires that the coefficients be multiplied and the exponents of the variables be added. Note in equation (1-16), that we shall always be multiplying coefficients of the f^i and ξ^i expansions. This presents no problem in our representation scheme since the terms of the f^i expansion contain no "variable coefficients" $a^i_{k_1 \ldots k_n}$, then the variable coefficients of the ξ^i expansion are always multiplied by unity. For example, consider the two polynomials f^1, ξ^3

$$f^1 = 2x_1^2x_2x_4 - 16x_3^3x_4 + 8x_2x_3^2x_4 \qquad (7)$$

$$\times \quad \xi^3 = 8a^2_{1001}x_1x_4 - 3a^1_{1111}x_2x_4 + 4a^2_{0010}x_1x_3$$

$$f^1 \cdot \xi^3 = 16a^2_{1001}x_1^3x_2x_4^2 - 6a^1_{1111}x_1^2x_2^2x_4^2 + 8a^2_{0010}x_1^3x_2x_3x_4$$

$$- 128a^2_{1001}x_1x_3^3x_4^2 - 48a^1_{1111}x_2x_3^3x_4^2 - 64a^2_{0010}x_1x_3^4x_4$$

$$+ 64a^2_{1001}x_1x_2x_3^2x_4^2 - 24a^1_{1111}x_2^2x_3^2x_4^2 + 32a^2_{0010}x_1x_2x_3^2x_4$$

In our representation, we have:

f^1 = [2.0 | 2101 |] → [-16.0 | 0031 |] → [8.0 | 0121 | /] (8)

ξ^3 = [8.0 | 2.1001 | 1001 |] → [-3.0 | 1.1111 | 0101 |] → [4.0 | 2.0010 | 1010 | /]

f = [16.0 | 2.1001 | 3102 |] → [-6.0 | 1.1111 | 2202 |] → [8.0 | 2.0010 | 3111 |] →

→ [-128.0 | 2.1001 | 1032 |] → [48.0 | 1.1111 | 0234 |] → [-64.0 | 2.0010 | 1041 |] →

→ [64.0 | 2.1001 | 1122 |] → [-24.0 | 1.1111 | 0222 |] → [32.0 | 2.0010 | 1121 | /]

(iv) Derivatives of Polynomials

Taking the derivative of polynomials is straightforward. It simply requires one to "find" the exponent of the variable x_i which we are taking the derivative with respect to, multiplying the real coefficient by that exponent, and lowering that exponent by one. Thus we have from the example above

$$\frac{d}{dx_2}\xi^3 = -3a^1_{1111}x_4$$

= [-3.0 | 1.1111 | 0001 | /]

(v) Sorting of Polynomials

The sorting method used to collect terms was that of the RADIX sort. Initially, it was thought that the sorting procedure required a sort on several "keys" (the number of keys being equal to the dimension of the problem). Since then, however, it became apparent that all that was needed was a sorting procedure which would simply recognize and collect common terms. This part of the program has not yet been simplified. At present the program essentially sorts the polynomials on the exponent field of the dynamical variables such that the polynomial is put in order from smallest exponent number to the largest exponent number. For example

$x_1^2 + x_2^2 - x_3^4 x_1$ = [1.0 | 2000 |] → [1.0 | 0200 |] → [1.0 | 1040 | /]

becomes

[1.0 | 0200 |] → [-1.0 | 1040 |] → [1.0 | 2000 | ///] $= x_2^2 - x_1 x_3^4 + x_1^2$

A modification of the sorting procedure could reduce the processing time slightly, however the amount of time saved would probably be insignificant considering the size of the polynomials sorted.

4-3.3 Linear Equation Solver

Once the determining equations have been obtained, the linear equations are set into a matrix. The size of the matrix is limited by the computer system. On the Burrough's B6700, the matrix size is limited to 1023 x 1023. The linear equations are solved using a subroutine called MFGR which is part of an IBM package of matrix manipulating techniques. This subroutine was translated from Fortran to Pascal and modified to handle two-dimensional arrays in order to allow matrices of much larger dimensions than the program was initially intended for.

This linear equation solver is presently the slow step in the DETERMININGEQNS program. It was recognized by the author only recently that a more efficient program could be written which would take into account the fact that the matrices dealt with in the program are "sparse" matrices. A method to deal with sparse matrices was found in Fundamentals of Data Structures by Horowitz and Sahni. Thus, the subroutine MFGR could be considerably modified such that the searching time required in MFGR could be greatly reduced. The only handicap seen by the author is the sizes of the matrix allowed in this case. It seems that according to this "revised" method, the number of non-zero matrix elements will be limited to the number of rows or columns allowed in a two-dimensional array peculiar to the computer system used (therefore on the B6700, we would be limited to 1,023 non-zero elements in our matrix).

4-3.4 Truncation Scheme

Consider an n-dimensional problem where

$$U = \xi^{i}\partial/\partial x^{i} \qquad i = 1\ldots n \tag{1}$$

and ξ^i are approximated by an expansion

$$\xi^{i} = a^{i}_{j_1\ldots j_n}(x^1)^{j_1}\ldots(x^n)^{j_n} \qquad \text{for } j_1 + \;\; + \ldots + j_n = 0\ldots \tag{2}$$

where $a^{i}_{j_1\ldots j_n}$ are arbitrarily unknown coefficients. Also, we have a given evolution generator (V)

$$V = f^{i}\partial/\partial x^{i} \qquad i = 1\ldots n \tag{3}$$

where the f^i are known functions

$$f^{i} = b^{i}_{j_1\ldots j_n}(x^1)^{j_1}\ldots(x^n)^{j_n} \qquad j_1 + j_2 + \ldots + j_n = s_0\ldots m_v.$$

Thus our task is to make an approximation for U such that

$$[U,V] = UV - VU = 0 \tag{5}$$

$$= \xi^{i}\partial/\partial x^{i} f^{k}\partial/\partial x^{k} - f^{k}\partial/\partial x^{k}\xi^{i}\partial/\partial x^{i} \qquad k = 1\ldots n$$

or

$$0 = [a^{i}_{j_1\ldots j_n}(x\;)^{j_1}\ldots(x^n)^{j_n}]\partial/\partial x^{i}[b^{k}_{\ell_1\ldots\;\;n}(\;\;)^{\ell_1}\ldots(x^n)^{\ell_n}]\partial/\partial x^{k} \tag{6}$$

$$- [b^{k}_{\ell_1\ldots\ell_n}(x^1)^{\ell_1}\ldots(x^n)^{\ell_n}]\partial/\partial x^{k}[a^{i}_{j_1\ldots j}(x^1)^{j_1}\ldots(x^n)^{j_n}]\partial/\partial$$

for $i,k = 1\ldots n$

$j_1 + \ldots + j_n = 0\ldots\;\; _u$

$\ell_1 + \ldots + \ell_n = s_0.\;\; m_v$

$$= (a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots\ell_n})(\ell_i)[(x^1)^{j_1}\ldots(x^n)^{j_n}][(x^1)^{\ell_1}\ldots(x^i)^{\ell_i-1}\ldots(x^n)^{\ell_n}]\partial/\partial x^k$$

$$- (a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots n})(j_k)[(x^1)^{\ell_1}\ldots(x^n)^{\ell_n}][(x^1)^{j_1}\ldots(x^k)^{j_k-1}\ldots(x^n)^{j_n}]\partial/\partial x^i$$

$$= (\ell_i)(a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots\ell_n})[(x^1)^{j_1+\ell_1}\ldots(x^i)^{j_i+\ell_i-1}\ldots(x^n)^{j_n+\ell_n}]\partial/\partial x^k$$

$$- (j_k)(a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots\ell_n})[(x^1)^{j_1+\ell_1}\ldots(x^k)^{j_k+\ell_k-1}\ldots(x^n)^{j_n+\ell_n}]\partial/\partial x^i.$$

Changing indices in the second term, we have

$$= [(\ell_i)(a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots\ell_n}) - (j_i)(a^k_{j_1\ldots j_n} \cdot b^i_{\ell_1\ldots\ell_n})] \qquad (7)$$

$$\times [(x^1)^{j_1+\ell_1}\ldots(x^i)^{j_i+\ell_i-1}\ldots(x^n)^{j_n+\ell_n}]\partial/\partial x^k.$$

Let $m_i = j_i + \ell_i$, then

$$0 = [(\ell_i)(a^i_{j_1\ldots j_n} \cdot b^k_{\ell_1\ldots\ell_n}) - (j_i)(a^k_{j_1\ldots j_n} b^i_{\ell_1\ldots\ell_n})]$$

$$[(x^1)^{m_1}\ldots(x^i)^{m_i-1}\ldots(x^n)^{m_n}]\partial/\partial x^k. \qquad (8)$$

From this, we have for the total power in each term (m) is

$$m = \sum_{i=1}^{n} m_i - 1 \qquad (9)$$

$$= j_1 + \ell_1 + \ldots + j_i + \ell_i + \ldots + j_n + \ell_n - 1$$

$$= (j_1 + \ldots + j_n) + (\ell_1 + \ldots + \ell_n) - 1$$

and the maximum total power in the expansion (m_t) is

$$m_t = (j_1 + \ldots + j_n)_{max} + (\ell_1 + \ldots + \ell_n)_{max} - 1$$

$$= m_u + m_v - 1. \qquad (10)$$

The minimum power in the expansion (m_0) is

$$m_0 = s_0 - 1 \qquad (11)$$

Thus the range of m is

$$m = m_0 \ldots m_t = (s_0 - 1) \ldots (m_u + m_v - 1). \tag{12}$$

If we let

$$p_k = j_1 + \ldots + j_n; \quad r_k = \ell_1 + \ldots + \ell_n - 1$$
$$p_i = j_1 + \ldots + j_n - 1; \quad r_i = \ell_1 + \ldots + \ell_n \tag{13}$$

careful examination of (6) reveals that if we were to truncate our equations on t, where t is greater than the sum of the maximum value of p_k and the lowest value of r_k (likewise for p_i and r_i), that is, if

$$t > p_{k_{max}} + r_{k_{min}} \tag{14}$$

then we will introduce an error into our equations, in that we will not have a sufficient "power" in some terms, as if we were neglecting the lower orders of the f^i expansion (likewise for the ξ^i expansion if we are considering the $p_{i_{min}}$ and $r_{i_{max}}$ terms). Thus, we note that we must have

$$t \leq p_{k_{max}} + r_{k_{min}} = p_{i_{min}} + r_{i_{max}}$$
or
$$t \leq m_u + s_0 - 1 = m_u - 1 + s_0 . \tag{15}$$

4-3.5 Instructions for DETERMININGEQNS

A flow chart for the use of DETERMININGEQNS is given in Appendix II. In this section we will briefly explain the meanings of the questions asked in the program and the format in which the questions must be answered. Unless the format is strictly adhered to, many complications could arise. The flow chart should be referred to to help understand the following explanations.

The first question (1) asked in the program refers to the number of dynamical variables in the problem. Thus, if we were dealing with, for example, the planar Kepler problem, we would have two degrees of freedom. Hamilton's equations, however, involve four dynamical variables. The dimension of the problem then, is four. The dimension should be entered as an integer (i.e. no decimal numbers allowed).

The second question (2) asks whether the user has a θ expansion ($U = \theta^i \partial_{x^i}$; U: the generator of the general transformation) already on file (D1) which he would like to use, or whether he would like to have a new θ expansion to work with. This option was incorporated in order to save the user some time if he were running similar problems successively. The user must answer either "Y" or "N" for "Yes" or "No", respectively. An answer of "Yes" or "No" would result in some complications. This restriction must be followed throughout the program in response to any Yes or No questions. The only character acceptable other than "Y" or "N" is the blank character " " to which the computer will not respond.

Question (3) assumes that the user wants to change his θ expansion. It asks whether the user would like to enter his own θ expansion (in a specified format to be discussed later) or if he wants the computer to systematically enter a θ expansion from zeroth order to some given order in each variable or to some total order in all variables.

eg. 2^{nd} Total Order

$$\theta = a_{000} + a_{100}x_1 + a_{010}x_2 + a_{001}x_3 + a_{200}x_1^2 + a_{020}x_2^2 + a_{002}x_3^2$$
$$+ a_{101}x_1x_3 + a_{110}x_1x_2 + a_{011}x_2x_3$$

or

2^{nd} Order in Each Variable

$$\theta = a_{000} + a_{100}x_1 + a_{010}x_2 + a_{001}x_3 + a_{200}x_1^2 + a_{020}x_2^2 + a_{002}x_3^2$$
$$+ a_{101}x_1x_3 + a_{110}x_1x_2 + a_{011}x_2x_3 + a_{201}x_1^2x_3 + a_{210}x_1^2x_2 + a_{120}x_1x_2^2$$
$$+ a_{021}x_2^2x_3 + a_{102}x_1x_3^2 + a_{012}x_2x_3^2 + a_{220}x_1^2x_2^2 + a_{202}x_1^2x_3^2 + a_{022}x_2^2x_3^2$$
$$+ a_{222}x_1^2x_2^2x_3^2$$

The user must answer "Y" or "N".

Question (4) assumes the user would like to enter his own θ expansion. Note that the θ expansion must be entered as an n-digit integer for an n-dimensional problem as explained in section II. Each "term" of the expansion must be separated by a blank.

eg. As in the last example for "2^{nd} Total Order", the computer will print

"ENTER THE THETA EXPANSION:"

The user must then enter the expansion as

000 100 010 001 200 020 002 101 110 011 -1

Note that in order to signal the computer that you have entered the last term of the expansion, the last entry must be a negative integer.

Question (5) assumes that the user wants the computer to automatically enter a θ expansion for him. The question then, is whether he wants the θ expansion truncated at a total power in all variables as explained in question (3). If a total maximum power is desired, the user must reply "Y". If a maximum power in each variable is desired, the user must reply "N". No other option is allowed.

Question (6) asks for the maximum power referred to in question (5). The maximum power must be a positive integer less than 10.

Question (7) asks if the user would like to enter some new functions (f^i) as defined in section 4-1, equation (1) on the systematic derivation of the determining equations, or if the function (f^i) are already on file (D2,D3) in the same manner as the θ expansion.

Question (8) assumes the user wants to enter new functions (f^i). The functions must be entered as in the representation specified in section III (Technical Aspects of DETERMININGEQNS), equation (3'). The real coefficient, then the dynamical variable must be entered (as a pair).

<u>eg</u>. Consider a polynomial

$$f^1 = 3x_1^2 - 2x_3 + 25.4x_1^2x_2$$

This must be entered as

"ENTER FUNCTION F(1):"

3.0 200

-2.0 001

25.4 210

5E10 -1

Note that the last two entries must be a real number $\leq$ 5E10 and a negative integer, respectively.

Question (9) asks whether there are any mainfolds which the user would also like to be left invariant. The operator again answers "Y" or "N".

Question (10) assumes that the user wants some manifolds left invariant. The question is, "HOW MANY MANIFOLDS DO YOU WANT LEFT INVARIANT?".

The reply must be an integer.

Questions (11), (12) and (13) are repeated m times, where m is the number of manifold restrictions requested in question (10). In question (11) the user must specify what the maximum power or "order" of each manifold expression is. That is, "WHAT IS THE ORDER OF W(I)?". Question (12) asks whether or not the restrictions should be subjected to the conditions of the manifold (i.e. do you want $UW = 0$ or $UW|_{W=0} = 0$, where U is the invariance generator). If the user wants $UW|_{W=0} = 0$, then he must answer "Y".

Question (13) is simply a directive to the user telling him to enter the equation for the i^{th} manifold W^i in the same manner as the functions f^i were entered in question (7), however, with the additional requirements that the user enter the highest order term first. In the case where there are two or more terms that could be considered as the "highest order", either will suffice.

Question (14) asks what ultimate order of approximation to the generator U the user wants. Note that the order of the θ expansion asked for in question (6) is not generally the order of approximation to the generator U one seeks. The relation is described in section 4-3.4 on the "Truncation Scheme".

The final question (15) is an option for the user to see the determining equations. This option is especially useful in troubleshooting problems that may arise in the equations. The user again must reply "Y" or "N".

4-3.6. The Limit Cycle - An Example Problem

The limit cycle serves as a good example to show how DETERMININGEQNS may be used to approximate the generators admitted by a system of autonomous ordinary differential equations. In polar coordinates, the system of equations

$$\frac{dr}{dt} = r(1 - r^2); \qquad \frac{d\phi}{dt} = 1 \tag{1}$$

have a limit cycle solution. Setting

$$\begin{aligned} x_1 &= r; & x_2 &= \phi \\ f^1 &= r(1 - r^2); & f^2 &= 1 \end{aligned} \tag{2}$$

we may obtain a 4th order approximation to the generators $U^i = \Theta^i \partial/\partial x_i$ by truncating our Θ expansion on $R_u = 4$ and $T_u = R_u + S_0 - 1 = 3$ (see section 4-3.4, "Truncation Scheme"). The input/output is shown on the following page; from which we obtain the following 10 independent generators:

$$\begin{aligned}
U_4^1 &= -\frac{1}{3}\left(1 + x_2 + \frac{1}{2}x_2^2 - 3x_1^2x_2 + \frac{1}{6}x_2^3 + \frac{1}{24}x_2^4\right)\partial/\partial x_1 \\
&= -\frac{1}{3}\left(1 + \phi + \frac{1}{2}\phi^2 - 3r^2\phi + \frac{1}{6}\phi^3 + \frac{1}{24}\phi^4\right)\partial/\partial r \\
U_4^2 &= -x_1\left(1 - x_2 + \frac{1}{2}x_2^2 + \frac{1}{3}x_1^2 - \frac{1}{6}x_2^3\right)\partial/\partial x_2 \\
&= -r\left(1 - \phi + \frac{1}{2}\phi^2 + \frac{1}{3}r^2 - \frac{1}{6}\phi^3\right)\partial/\partial\phi \\
U_4^3 &= -\frac{1}{3}\left(1 + x_2 - 3x_1^2 + \frac{1}{2}x_2^2 + \frac{1}{3}x_2^3 - \frac{3}{2}x_1^2x_2^2 + \frac{1}{24}x_2^4\right)\partial/\partial x_1 \\
&= -\frac{1}{3}\left(1 + \phi - 3r^2 + \frac{1}{2}\phi^2 + \frac{1}{3}\phi^3 - \frac{3}{2}r^2\phi^2 + \frac{1}{24}\phi^4\right)\partial/\partial r \\
U_4^4 &= \partial/\partial x_2 = \partial/\partial\phi \\
U_4^5 &= x_1^3\left(-\frac{1}{2} + x_2\right)\partial/\partial x_1 = \frac{1}{2}r^3(-1 + 2\phi)\partial/\partial r
\end{aligned} \tag{3}$$

****

THE LIMIT CYCLE - AN EXAMPLE PROBLEM

****

```
R DETEQNS7
#RUNNING 6414
WHAT IS THE DIMENSION OF THE PROBLEM?
#?
2
DO YOU WISH TO CHANGE YOUR THETA EXPANSION?
ANSWER Y OR N FOR YES OR NO RESPECTIVELY.
Y
DO YOU WANT TO ENTER YOUR OWN THETA EXPANSION?
  N
DO YOU WANT A TOTAL MAXPWR TRUNCATION?  IF NOT,
WE WILL TRUNCATE AT A MAXIMUM POWER OF EACH
VARIABLE.
Y
WHAT IS THE TOTAL MAXIMUM POWER OF YOUR EXPANSION?
4
#6414 (A015006)D1 REMOVED ON SWAT PK065 .
         0         1         2         3         4        10        11        12
        13        20        21        22        30        31        40
DO YOU WISH TO CHANGE YOUR FUNCTIONS F[I]?
Y
ENTER COEFFICIENT AND EXPONENTS OF EACH TERM OF F[I]
AS FOLLOWS:
         COEFF                EXPONENT

ENTER THE EXPONENT AS AN INTEGER, EACH DIGIT REPRESENT-
ING THE POWER OF AN X-VARIABLE
TERMINATE WITH A COEFF-FIELD OF 5E10 AND NEGATIVE EXP
ENTER TERMS OF FUNCTION F[ 1]:
1.0 10
-1.0 30
5E10 -1
ENTER TERMS OF FUNCTION F[ 2]:
1.0 00
5E10 -1
#6414 (A015006)D2 REMOVED ON SWAT PK065 .
#6414 (A015006)D3 REMOVED ON SWAT PK065 .
ECHO DATA:
THE FUNCTION F [ 1] IS:
         1          10
        -1          30
THE FUNCTION F [ 2] IS:
         1           0
DO YOU WANT TO IMPOSE A MANIFOLD RESTRICTION?
N
```

```
REPLY IS:N
WHAT TOTAL MAXIMUM POWER DO YOU WISH TO TRUNCATE ON?
3
DO YOU WANT THE DETERMINING EQUATIONS PRINTED OUT?
N
REPLY IS:N
THE SIZE OF THE MATRIX IS:  20 X  30=       600
#ET=2:21.7 PT=5.8 IO=4.2

R MFGRDETEQNS7
#RUNNING 6418
WHAT IS THE DIMENSION OF THE PROBLEM?
#?
2
IS THIS A NEW PROBLEM?
Y
THE SIZE OF THE MATRIX IS:  20X  30=       600
HOW MANY ITERATIONS DO YOU WANT?
30
THE RANK IS:        20

THE GENERATORS ARE:


U[   1] = (A 1.21) +  -0.013888888889(A 1.04) +  -0.055555555556(A 1.03) +
 -0.333333333334(A    1) +  -0.333333333334(A 1.01) +  -0.166666666668(A 1.02
) +

U[   2] = (A 2.11) +  0.166666666666(A 2.13) +  -0.333333333334(A  2.3) +  -0.5
(A 2.12) +     -1(A  2.1) +

U[   3] = (A  1.2) +  -0.013888888889(A 1.04) +  -0.055555555556(A 1.03) +
 -0.333333333334(A    1) +  -0.333333333334(A 1.01) +  -0.166666666668(A 1.02
) +   0.5(A 1.22) +

U[   4] = (A    2) +

U[   5] = (A 1.31) +  -0.5(A  1.3) +

U[   6] = (A  1.4) +

U[   7] = (A 2.22) +   0.5(A  2.2) +    -1(A 2.21) +

U[   8] = (A  1.1) +    -1(A  1.3) +

U[   9] = (A 2.31) +  -0.333333333334(A  2.3) +

U[  10] = (A  2.4) +

#6418 (A015006)GEN REMOVED ON SWAT PK065 .
DO YOU WANT TO SEE THE MATRIX?
```

$$U_4^6 = x_1^4\partial/\partial x_1 = r^4\partial/\partial r$$

$$U_4^7 = \frac{1}{2}x_1^2(1 - 2x_2 + x_2^2)\partial/\partial x_2 = \frac{1}{2}r^2(1 - 2\phi + \phi^2)\partial/\partial\phi$$

$$U_4^8 = (x_1 - x_1^3)\partial/\partial x_1 = (r - r^3)\partial/\partial r$$

$$U_4^9 = -\frac{1}{3}x_1^3(1 - x_2)\partial/\partial x_2 = -\frac{1}{3}r^3(1 - \phi)\partial/\partial\phi$$

$$U_4^{10} = x_1^4\partial/\partial x_2 = r^4\partial/\partial\phi$$

Note that the evolution generator (V) is given by

$$V = U_4^8 + U_4^4. \quad (4)$$

Wulfman [31] has shown that the most general form of the generators admitted by this system is given by

$$U = \xi^r\partial/\partial r + \xi^\phi\partial/\partial\phi \quad (5)$$

where ξ^r is an arbitrary linear combination of the functions ($\xi^r = k_\alpha\xi_\alpha^r$)

$$\xi_\alpha^r = r^{\alpha+1}(r^2 - 1)^{1-\frac{1}{2}\alpha}\exp(-\alpha\phi), \quad (6a)$$

and ξ^ϕ is an arbitrary linear combination of the functions ($\xi^\phi = k_\beta\xi_\beta^\phi$)

$$\xi_\beta^\phi = r^\beta(r^2 - 1)^{-\frac{1}{2}\beta}\exp(-\beta\phi) \quad (6b)$$

if α and β are arbitrary parameters. Expanding these functions in Taylor series around the origin ($r = \phi = 0$), (6a) and (6b) become

$$\xi_{\bar{\alpha}}^r = (-1)^{3\bar{\alpha}+m+2}\cdot r^{2\bar{\alpha}+1}\cdot\frac{(2\bar{\alpha}\phi)^m}{m!}\cdot[1 + \frac{(1+\bar{\alpha})(1+\bar{\alpha}-1)(1+\bar{\alpha}-2)\cdots(1+\bar{\alpha}-n+1)}{n!}(-1)^n r^{2n}]$$

$$= (-1)^{3\bar{\alpha}+2} r^{2\bar{\alpha}+1} [1 + \frac{(1+\bar{\alpha})(1+\bar{\alpha}-1)\cdots(1+\bar{\alpha}-n+1)}{n!} (-1)^n r^{2n}]$$

$$+ (-1)^{3(\bar{\alpha}+1)} r^{2\bar{\alpha}+1} (2\bar{\alpha}\phi) [1 + \frac{(1+\bar{\alpha})\cdots(1+\bar{\alpha}-n+1)}{n!} (-1)^n r^{2n}] \quad (7a)$$

$$+ (-1)^{3\bar{\alpha}+4} r^{2\bar{\alpha}+1} 2\bar{\alpha}^2\phi^2 [1 + \frac{(1+\bar{\alpha})\cdots(1+\bar{\alpha}-n+1)}{n!} (-1)^n r^{2n}]$$

$$+ (-1)^{3\bar{\alpha}+5} r^{2\bar{\alpha}+1} \frac{4}{3} \bar{\alpha}^3\phi^3 [1 + \frac{(1+\bar{\alpha})\cdots(1+\bar{\alpha}-n+1)}{n!} (-1)^n r^{2n}]$$

$$+ \ldots + + +$$

$$\xi^{\phi}_{\beta} = (-1)^{\bar{\beta}+1} r^{-2\bar{\beta}} \frac{(2\bar{\beta}\phi)^m}{m!} [1 + \frac{\bar{\beta}(\bar{\beta}-1)\cdots(\bar{\beta}-n+1)}{n!} (-1)^n r^{2n}]$$

$$= (-1)^{\bar{\beta}+1} r^{-2\bar{\beta}} [1 + \frac{\bar{\beta}(\bar{\beta}-1)\cdots(\bar{\beta}-n+1)}{n!} (-1)^n r^{2n}]$$

$$+ (-1)^{\bar{\beta}+1} r^{-2\bar{\beta}} 2\bar{\beta}\phi [1 + \frac{\bar{\beta}(\bar{\beta}-1)\cdots(\bar{\beta}-n+1)}{n!} (-1)^n r^{2n}] \quad (7b)$$

$$+ (-1)^{\bar{\beta}+1} r^{-2\bar{\beta}} 2\bar{\beta}^2\phi^2 [1 + \frac{\bar{\beta}(\bar{\beta}-1)\cdots(\bar{\beta}-n+1)}{n!} (-1)^n r^{2n}]$$

$$+ (-1)^{\bar{\beta}+1} r^{-2\bar{\beta}} \frac{4}{3} \bar{\beta}^3\phi^3 [1 + \frac{\bar{\beta}(\bar{\beta}-1)\cdots(\bar{\beta}-n+1)}{n!} (-1)^n r^{2n}] + \ldots + +$$

where $\bar{\alpha} = -\frac{\alpha}{2}$ and $\bar{\beta} = -\frac{\beta}{2}$. The following correspondences between the computer-generated approximations (3) and those of equations (7) can be made:

$$U^1_4 \leftarrow\rightarrow -\frac{i}{3} [\xi^{r}_{\alpha} - O(r^2) - O(r^2\phi^2) - O(r^2\phi^3) - O(r^2\phi^4)] \partial/\partial r \; : \; \bar{\alpha} = -\tfrac{1}{2}; \; m=0,1,2,3,4; \; n = 1$$

$$U^2_4 \leftarrow\rightarrow i \; [\xi^{\phi}_{\beta} - O(r^2\phi) - O(r^2\phi^2) - O(r^2\phi^3)] \partial/\partial\phi \; : \; \bar{\beta}=-\tfrac{1}{2}; \; m=0,1,2,3; \; n=1$$

$$U^3_4 \leftarrow\rightarrow -\frac{i}{3} [\xi^{r}_{\alpha} - O(r^2\phi) - O(r^2\phi^3) - O(r^2\phi^4)] \partial/\partial r \; : \; \bar{\alpha}=-\tfrac{1}{2}; \; m=0,1,2,3,4; \; n=1$$

$$U^4_4 \leftarrow\rightarrow - \xi^{\phi}_{\beta} \partial/\partial\phi \; : \; \bar{\beta}=0; \; m=n=1$$

$$U^5_4 \leftarrow\rightarrow \xi^{r}_{\alpha} \partial/\partial r \; : \; \bar{\alpha}=1; \; m=0,1; \; n=0$$

$$U_4^6 \leftrightarrow -i\xi_{\bar{\alpha}}^{r}\ \partial/\partial r \qquad : \quad \bar{\alpha} = \frac{3}{2}\ ;\ m=n=0$$

$$U_4^7 \leftrightarrow \frac{1}{2}\ \xi_{\bar{\beta}}^{\phi}\ \partial/\partial\phi \qquad : \quad \bar{\beta} = -1\ ;\ m=0,1,2\ ;\ n=0$$

$$U_4^8 \leftrightarrow \xi_{\alpha}^{r}\partial/\partial r \qquad : \quad \alpha = 0 \qquad (8)$$

$$U_4^9 \leftrightarrow i\xi_{\bar{\beta}}^{\phi}\ \partial/\partial\phi \qquad : \quad \bar{\beta} = -\frac{3}{2}\ ;\ m=0,1\ ;\ n=0$$

$$U_4^{10} \leftrightarrow -\xi_{\bar{\beta}}^{\phi}\ \partial/\partial\phi \qquad : \quad \bar{\beta} = -2\ ;\ m=n=0$$

In the 5th order approximation, we find

$$U_5^1 = -\frac{1}{3}\left(1 + \phi - 3r^2 + \frac{1}{2}\phi^2 + \frac{1}{6}\phi^3 + \frac{1}{24}\phi^4 - \frac{3}{2}r^2\phi^2 + r^4 + \frac{1}{120}\phi^5\right)\partial/\partial r$$

$$U_5^2 = -\frac{1}{3}r^4(1 - 3\phi)\partial/\partial r$$

$$U_5^3 = \partial/\partial\phi$$

$$U_5^4 = -\frac{1}{3}\left(1 + \phi + \frac{1}{2}\phi^2 + \frac{1}{6}\phi^3 - 3r^2\phi + \frac{1}{24}\phi^4 - \frac{1}{6}r^2\phi^3 + \frac{1}{120}\phi^5\right)\partial/\partial r$$

$$U_5^5 = \frac{1}{2}r^2\left(1 - 2\phi + \frac{1}{2}r^2 + 2\phi^2 - \frac{4}{3}\phi^3\right)\partial/\partial\phi$$

$$U_5^6 = r^5\partial/\partial r \qquad (9)$$

$$U_5^7 = r\left(1 - \phi + \frac{1}{2}\phi^2 - \frac{4}{9}r^2 - \frac{1}{6}\phi^3 - \frac{1}{3}r^2\phi + \frac{1}{24}\phi^4\right)\partial/\partial\phi$$

$$U_5^8 = \frac{1}{2}r^3(1 - 2\phi + 2\phi^2)\partial/\partial r$$

$$U_5^9 = \frac{2}{9}r^3\left(1 - 3\phi + \frac{9}{2}\phi^2\right)\partial/\partial\phi$$

$$U_5^{10} = r(1 - r^2)\partial/\partial r$$

$$U_5^{11} = -\frac{1}{4}r^4(1 - 4\phi)\partial/\partial\phi$$

$$U_5^{12} = r^5\partial/\partial\phi$$

Comparing the generators obtained through the 4th and 5th order approximations, we have:

$$U_5^4 = U_4^1 - \frac{1}{6} r^2\phi^3\partial/\partial r + \frac{1}{20}\phi^5\partial/\partial r$$

$$-U_5^7 = U_4^2 - \frac{1}{9} r^2\partial/\partial\phi - \frac{1}{3} r^2\phi\partial/\partial\phi + \frac{1}{24}\phi^4\partial/\partial\phi$$

$$U_5^1 = U_4^3 + (r^4 + \frac{1}{120}\phi^5)\partial/\partial r$$

$$U_5^3 = U_4^4$$

$$U_5^8 = -U_4^5 + 2r^3\phi^2\partial/\partial\phi$$

$$U_5^2 = -\frac{1}{3}(U_4^6 - 3\phi r^4\partial/\partial r)$$

$$U_5^5 = U_4^7 + r^2(\frac{1}{2} r^2 + \phi^2 - \frac{4}{3}\phi^3)\partial/\partial\phi$$

$$U_5^{10} = U_4^8$$

$$U_5^9 = -\frac{2}{3}[U_4^9 + (-2r^3\phi + \frac{9}{2} r^3\phi^2)\partial/\partial\phi]$$

$$U_5^{11} = -\frac{1}{4}(U_4^{10} - 4r^4\phi\partial/\partial\phi).$$

```
R DETEQNS7
#RUNNING 6404
WHAT IS THE DIMENSION OF THE PROBLEM?
#?
2
DO YOU WISH TO CHANGE YOUR THETA EXPANSION?
ANSWER Y OR N FOR YES OR NO RESPECTIVELY.
Y
DO YOU WANT TO ENTER YOUR OWN THETA EXPANSION?
N
DO YOU WANT A TOTAL MAXPWR TRUNCATION?  IF NOT,
WE WILL TRUNCATE AT A MAXIMUM POWER OF EACH
VARIABLE.
Y
WHAT IS THE TOTAL MAXIMUM POWER OF YOUR EXPANSION?
5
#6405 (A015006)D1 REMOVED ON SWAT PK065 .
        0          1          2          3          4          5          10         11
        12         13         14         20         21         22         23         30
        31         32         40         41         50
DO YOU WISH TO CHANGE YOUR FUNCTIONS F[I]?
N
ECHO DATA:
THE FUNCTION F [ 1] IS:
        1          10
        -1         30
THE FUNCTION F [ 2] IS:
        1          0
DO YOU WANT TO IMPOSE A MANIFOLD RESTRICTION?
N
REPLY IS:N
WHAT TOTAL MAXIMUM POWER DO YOU WISH TO TRUNCATE ON?
4
DO YOU WANT THE DETERMINING EQUATIONS PRINTED OUT?
N
REPLY IS:N
THE SIZE OF THE MATRIX IS:  30 X  42=      1260
#ET=1:17.8 PT=6.2 IO=4.6


R MFGRDETEQNS7
#RUNNING 6408
WHAT IS THE DIMENSION OF THE PROBLEM?
#?
2
IS THIS A NEW PROBLEM?
Y
THE SIZE OF THE MATRIX IS:  30X  42=      1260
HOW MANY ITERATIONS DO YOU WANT?
42
```

```
THE RANK IS:         30

THE GENERATORS ARE:



U[   1] = (A  1.2) +  -0.00277777777779(A 1.05) +  -0.0138888888889(A 1.04) +
 -0.0555555555556(A 1.03) +  -0.333333333334(A    1) +  -0.333333333334(A 1.01
) +  -0.333333333334(A  1.4) +  -0.166666666668(A 1.02) +   0.5(A 1.22) +

U[   2] = (A 1.41) +  -0.333333333334(A  1.4) +

U[   3] = (A    2) +

U[   4] = (A 1.21) +  -0.00277777777779(A 1.05) +  -0.0138888888889(A 1.04) +
 -0.0555555555556(A 1.03) +  -0.333333333334(A    1) +  -0.333333333334(A 1.01
) +  0.166666666668(A 1.23) +  -0.166666666668(A 1.02) +

U[   5] = (A 2.22) +  0.25(A  2.4) +  -0.66666666666(A 2.23) +   0.5(A  2.2) +
   -1(A 2.21) +

U[   6] = (A  1.5) +

U[   7] = (A  2.1) +  0.0416666666665(A 2.14) +  -0.166666666666(A 2.13) +
 0.444444444443(A  2.3) +  -0.333333333334(A 2.31) +   0.5(A 2.12) +    -1(A
 2.11) +

U[   8] = (A 1.32) +   0.5(A  1.3) +    -1(A 1.31) +

U[   9] = (A 2.32) +  0.222222222223(A  2.3) +  -0.66666666666(A 2.31) +

U[  10] = (A  1.1) +    -1(A  1.3) +

U[  11] = (A 2.41) +  -0.25(A  2.4) +

U[  12] = (A  2.5) +

#6408 (A015006)GEN REMOVED ON SWAT PK065 .
DO YOU WANT TO SEE THE MATRIX?
N
#6408 (A015006)MAT REMOVED ON SWAT PK065 .
#ET=1:11.7 PT=6.6 IO=1.9
```

Making the following correspondences between the 4th and 5th order generators

$$\begin{array}{llll} U_4^1 \leftrightarrow U_5^4 & & U_4^7 \leftrightarrow U_5^5 & \\ U_4^2 \leftrightarrow U_5^7 & & U_4^8 \leftrightarrow U_5^{10} & \\ U_4^3 \leftrightarrow U_5^1 & & U_4^9 \leftrightarrow U_5^9 & \\ U_4^4 \leftrightarrow U_5^3 & & U_4^{10} \leftrightarrow U_5^{11} & \\ U_4^5 \leftrightarrow U_5^8 & & ? \leftrightarrow U_5^6 & \\ U_4^6 \leftrightarrow U_5^2 & & ? \leftrightarrow U_5^{12} & \end{array} \qquad (10)$$

we find upon higher and higher orders of approximation (verified up to 7th order) that these generators are all stable; and the coefficients settle down asymptotically to those calculated from equations (7).

Although a complete and thorough investigation of the stability of the computer-calculated generators has not been made (e.g. stability about various expansion points), the success of the approximations for the generators of the limit cycle optomistically suggests that such computer methods (in more sophisticated forms) may be successfully used to systematically uncover the symmetry properties of large physical systems. This success may at first seem trivial; however, a closer examination of the invariance transformations admitted by this system reveals a non-trivial feature. The trajectories of the limit cycle are not all diffeomorphic. There exists the limit cycle trajectory which admits only one functionally independent one-parameter compact group of transformations ($U = \partial/\partial\phi$). For this reason, it might be expected that higher and higher approximations to the generators would evidence some instability as the vector functions ξ_{α}^{r} and ξ_{β}^{ϕ} become asymptotically closer to 0 and 1, respectively.

V. Some One-Parameter Groups Admitted By The Two-Dimensional Anisotropic Harmonic Oscillator

The first approximation to many physical problems is that of the n-dimensional anisotropic harmonic oscillator. Such is the case, for example, (c.f. section 6-4) in the planar Kepler two-center problem. With this motivation, we shall now examine the group-theoretical properties of the 2D anisotropic harmonic oscillator. The Hamiltonian for the 2D AHO is

$$H = \tfrac{1}{2}[(p_1)^2 + k_1(q_1)^2 + (p_2)^2 + k_2(q_2)^2] \tag{1}$$

or, in action-angle variables with the force constant k_i (a variable parameter), the Hamiltonian can be expressed as

$$\begin{aligned} H &= \tfrac{1}{2}(\omega_1 J_1 + \omega_2 J_2) \\ &= \tfrac{1}{2}(H_1 + H_2) \\ H_1 &= \omega_1(J_1) \\ H_2 &= \omega_2(J_2) \end{aligned} \tag{2}$$

where $\omega_1 = k_1^{\frac{1}{2}}$ and $\omega_2 = k_2^{\frac{1}{2}}$.

In this four-dimensional phase space (we consider the energy as a parameter), we have three constants of the motion, namely

$$\begin{aligned} H &= H_1 + H_2 \\ D &= H_1 - H_2 \\ K &= \omega_2\psi_1 - \omega_1\psi_2 \end{aligned} \tag{3}$$

where ψ_1 and ψ_2 are the angle variables conjugate to J_1 and J_2, respectively. Physically, H, D, and K are interpreted as

(a) H, the total energy

(b) D, the energy difference between the two-dimensional oscillators

(c) K, an angular phase type variable between the two oscillators.

These constants of the motion may serve as kernels for the generators of canonical transformations. Following Sumi and Wulfman [33], we may define

$$\begin{aligned} x_1 &= K = \omega_2\psi_1 - \omega_1\psi_2 & y_1 &= \tfrac{1}{2}[-\frac{1}{\omega_2} J_1 + \frac{1}{\omega_1} J_2] \\ x_2 &= -(\frac{\psi_1}{\omega_1} + \frac{\psi_2}{\omega_2}) & y_2 &= W = (\omega_1 J_1 + \omega_2 J_2) - E \\ x_3 &= t & y_3 &= E \end{aligned} \tag{4}$$

where x_i and y_i are conjugate coordinate and momenta respectively. Then x_1, y_1, y_3 are constants of the motion. and for any analytic function $f(x_1, y_1, y_3)$, we have

$$U = \{f \;\bullet\} \tag{5}$$

where f is a constant of the motion and U is the generator of a canonical transformation. Furthermore, if we define

$$g^{k} = y_2^{k} \bullet x_2^{\ell}; \qquad k \geq 2, \quad \ell \geq 0 \tag{6}$$

then for $F = F(x_1, y_1, y_3, g^{k\ell})$

$$U = \{F \;\bullet\} \tag{7}$$

is the more general form of all M-canonical transformation generators of one-parameter degeneracy groups. It follows that these generators form a Lie Algebra

	$X_{i'}$	$Y_{i'}$	$X_{ij'}$	$Y_{ij'}$	$Z_{ij'}$	$P_{i'}$	$Q_{i'}$
X_i	0	0	0	$X_{i'}$	$X_{i'}$	$Z_{ij'}$	$X_{ij'}$
Y_i		0	$Y_{i'}$	0	$Y_{i'}$	$Y_{ij'}$	$Z_{ij'}$
X_{ij}			0	$Z_{ij'}$	$X_{ij'}$	*	*
Y_{ij}				0	$Y_{ij'}$	*	*
Z_{ij}					$Z_{ij'}$	$P_{i'}$	$Q_{i'}$
P_i						0	*
Q_i							0

(8)

where

$$
\begin{aligned}
X_i &= \partial/\partial x_i = \{y_i \cdot\} \\
Y_i &= \partial/\partial y_i = -\{x_i \cdot\} \\
X_{ij} &= y_i \partial/\partial x_j + y_j \partial/\partial x_i = \{y_i y_j \cdot\} \\
Y_{ij} &= x_i \partial/\partial y_j - y_j \partial/\partial y_i = -\{x_i x_j \cdot\} \\
Z_{ij} &= x_i \partial/\partial x_j - y_j \partial/\partial y_i = \{x_i y_j \cdot\} \\
P_i &= x\cdot y \partial/\partial y_i - x_i A = -\{(x\ y) x_i \cdot\} \\
Q_i &= x\cdot y \partial/\partial x_i + y_i A = \{(x\ y) y_i \cdot\}
\end{aligned}
\tag{9}
$$

Note, however, that the dimensions of our closed algebras (corresponding to those of Sumi and Wulfman) are smaller, in that the indices i,j run over a smaller range.

Our Hamiltonian can be rescaled such that

$$W = H - E = 0 \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \quad W = \frac{W}{\omega_1} = \frac{1}{\omega_1}(H - E) = \bar{H} - \bar{E} = 0 \tag{10}$$

and

$$\bar{H} = J_1 + \omega J_2 \tag{11}$$

where $\omega \equiv \frac{\omega_2}{\omega_1}$. Then our constants of the motion can be written with their corresponding generators as

$$\text{(a)} \quad J_1 \to \to \to U_{20} = \partial/\partial\psi_1$$

$$\text{(b)} \quad J_2 \to \to \to U_{41}\partial/\partial\dot{\psi}_2 \qquad (12)$$

$$\text{(c)} \quad \psi_2 - \omega\psi_1 \to \to U_{21} = -\partial/\partial J_2 + \omega\partial/\partial J_1 .$$

Dulock and McIntosh [13] have shown that the anisotropic harmonic oscillator has a Lie algebra which corresponds to that of the three dimensional rotation group. They obtain four constants of the motion labeled $\bar{H}$, $\bar{D}$, $\bar{L}$, and $\bar{K}$ (of which the first two correspond to our constants of the motion H, D). The set $\{\bar{K}, \bar{L}, \bar{D}\}$ have the commutation relations

	$\bar{K}$	$\bar{L}$	$\bar{D}$
$\bar{K}$	0	$2\omega D$	$-2\omega L$
$\bar{L}$	$-2\omega D$	0	$2\omega K$
$\bar{D}$	$2\omega L$	$-2\omega K$	0

.

Using programs DETERMININGEQNS and COMMUTATOR, we have found the following ten-parameter group of contact transformations which acts transitively on the manifold

$$U_{21} = -\omega\partial/\partial J_1 + \partial/\partial J_2$$

$$U_{51} = J_2\partial/\partial\psi_2$$

$$U_{53} = -J_2\partial/\partial J_1 + J_2\partial/\partial\omega + J_2\psi_1\partial/\partial\psi_2$$

$$U_{20} = \partial/\partial\psi_1 \qquad (13)$$

$$U_{41} = \partial/\partial\psi_2$$

$$U_{44} = -J_2\partial/\partial J_1 + \partial/\partial\omega + \psi_1\partial/\partial\psi_2$$

$$U_{49} = J_1\partial/\partial J_1 + \partial/\partial\omega + \omega\psi_1\partial/\partial\psi_2$$

$$U_{47} = \omega\partial/\partial\psi_2$$

$$U_{42} = J_1\partial/\partial J_1 + \omega\partial/\partial\omega + \psi_2\partial/\partial\psi_2$$

$$U_{46} = J_1\partial/\partial J_1 + J_2\partial/\partial J_2$$

and whose commutation relations are

	U_{21}	U_{51}	U_{53}	U_{20}	U_{41}	U_{44}	U_{49}	U_{47}	U_{42}	U_{46}
U_{21}	0	$-U_{41}$	$-U_{44}$	0	0	0	0	0	0	$-U_{21}$
U_{51}		0	0	0	0	0	0	0	$-U_{51}$	U_{51}
U_{53}			0	$-U_{51}$	0	0	$-U_{53}$	U_{51}	$-U_{53}$	U_{53}
U_{20}				0	0	U_{41}	U_{47}	0	0	0
U_{41}					0	0	0	0	$-U_{41}$	0
U_{44}						0	U_{44}	U_{41}	$-U_{44}$	0
U_{49}							0	U_{47}	0	0
U_{47}								0	0	0
U_{42}									0	0
U_{46}										0

from which we obtain the following subalgebras

$$A_0 = U_{21} + U_{46} + U_{20} + U_{41} + U_{44} + U_{49} + U_{47} + U_{42}$$

$$A_1 = U_{20} + U_{41} + U_{44} + U_{42}$$

$$A_2 = U_{20} + U_{41} + U_{44} + U_{49}$$

$$A_3 = U_{20} + U_{41} + U_{44} + U_{47} + U_{49}$$

$$A_4 = U_{20} + U_{41} + U_{44} + U_{42} + U_{49} \qquad (14)$$

$$A_5 = U_{20} + U_{41} + U_{44} + U_{42} + U_{47} + U_{49}$$

$$A_6 = U_{20} + U_{41} + U_{44}$$

$$A_7 = U_{44} + U_{49} \qquad A_9 = U_{21} + U_{46}$$

$$A_8 = U_{41} + U_{42} \qquad A_{10} = U_{47} + U_{49}$$

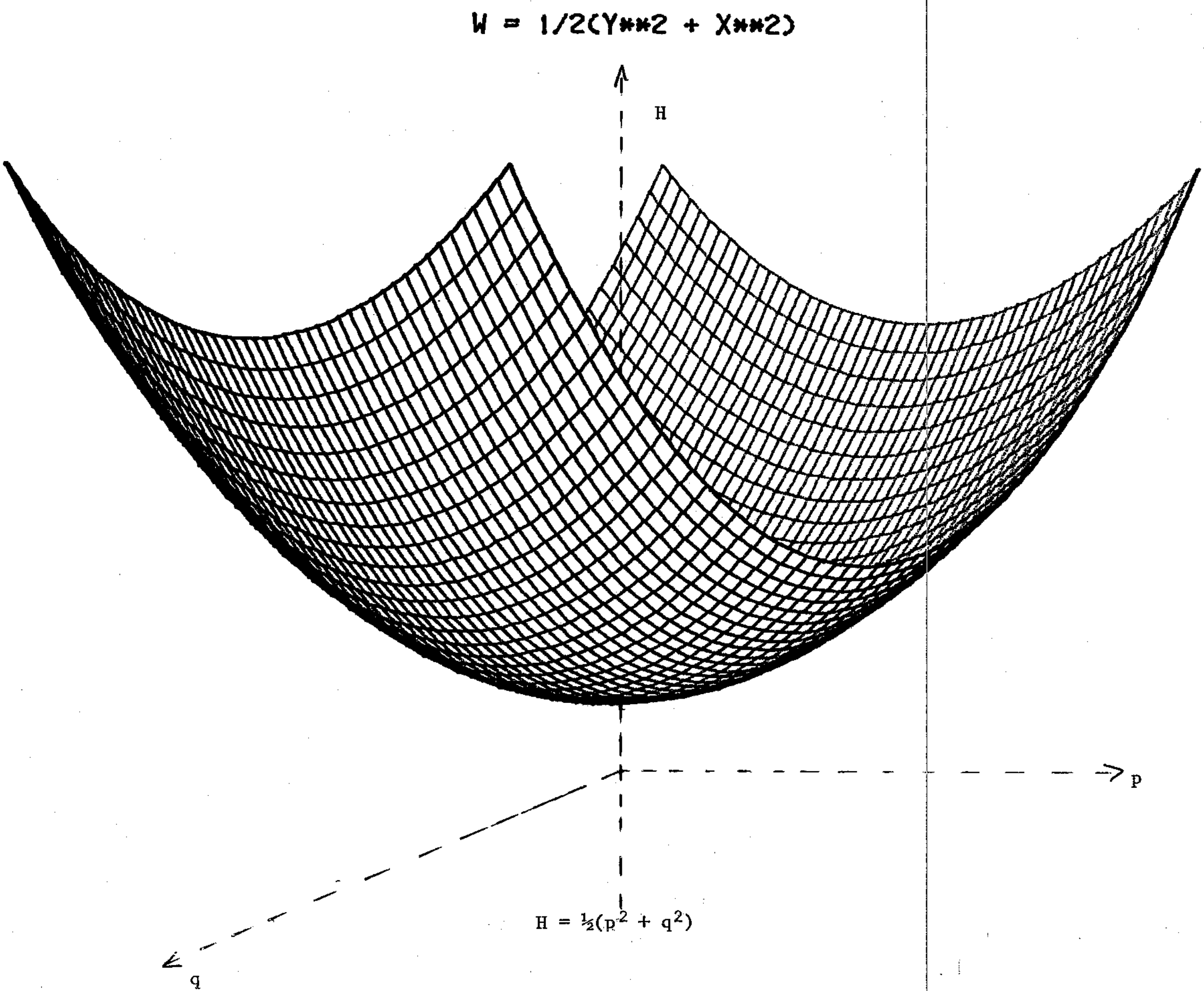
We find, however, that

$$\det|g_{\sigma\lambda}| = 0 \tag{15}$$

for the metric tensor $g_{\sigma\lambda}$ of the Lie algebra, where

$$g_{\sigma\lambda} = g_{\lambda\sigma} = c^{\tau}_{\sigma\rho}c^{\rho}_{\lambda\tau}$$
$$[U_{\sigma},U_{\rho}] = c^{\tau}_{\sigma\rho}U_{\tau}$$
$$[U_{\lambda},U_{\tau}] = c^{\rho}_{\lambda\tau}U_{\rho}$$

and thus we do not have a semi-simple Lie algebra. Note that A_6-A_{10} are the smallest non-trivial subalgebras. A_2 is the smallest semi-simple algebra that almost acts transitively (it acts transitively on the submanifold $J_1, \omega, \psi_1, \psi_2$). By adding A_9 we have a group that acts transitively on the manifold which, however, is not semi-simple.



$H = \frac{1}{2}(p^2 + q^2)$

VI. The Planar Two-Center Problem

Here we attempt to uncover the Lie groups admitted by the classical planar two-center problem. The techniques used are those discussed in the previous sections. Furthermore, it was found that a geometric analysis provides us with a tremendous visual aid which we use to guide our search.

The incentive for this group-theoretical analysis was provided by Fock [14] in his work on the hydrogen atom. Fock has shown in a very clever way that the accidental degeneracy of the hydrogen atom can be explained by the existence of a four-dimensional rotational symmetry in an "extended" momentum space. Through the use of group theory, it is possible to show in a straighforward manner how the four-dimensional rotation group SO(4) is admitted by the system.

In more complicated systems, exact solutions are often impossible. It becomes difficult in such systems then, to analytically explain or predict important physical phenomena. It is here, then that group theory can be used as a powerful tool to aid in uncovering important symmetry features of the system (such as conservation laws in classical systems or degeneracies in quantal systems). Once the group properties of a quantum mechanical system are known, approximate wave functions may be obtained through the use of the representation theory of groups. Shift operators may readily be found which then provides the researcher with all the tools needed to easily derive the full spectrum of quantum states for the system.

6-1. A Historical Survey of the Quantum-Mechanical Two-Center Problem

The problem of one electron moving in the field of two fixed nuclei (the restricted three-body problem) plays the same central role in molecular physics as the central potential problem plays in atomic physics. The hydrogen molecular ion serves as the simplest model for this quantum mechanical restricted three-body problem. Wolfgang Pauli was the first to treat this system in his Munich dissertation* In terms of the old quantum theory, he utilized the Bohr-Sommerfeld quantization condition

$$\oint p_i dq_i = 2\pi n_i \hbar \tag{1}$$

and obtained several molecular parameters for the H_2^+ system which proved to be quite unsatisfactory. Since the advent of modern quantum mechanics, many authors have successfully treated this problem.

The Hamiltonian for this system is

$$H = \frac{\vec{p}^2}{2m} - \left(\frac{Z^1}{r^1} + \frac{Z^2}{r^2}\right) \tag{2}$$

where r^1, r^2, are defined in figure 1. Thus, the Schroedinger equation becomes

$$\frac{-\hbar^2}{2m} \nabla^2\psi - \left(\frac{Z^1}{r^1} + \frac{Z^2}{r^2}\right)\psi = E\psi. \tag{3}$$

It was realized that this equation is separable in prolate spheroidal coordinates ξ^1, ξ^2, ϕ where

$$\xi^1 = \frac{r^1 + r^2}{R} \qquad (1 \leq \xi^1 \leq \infty) \tag{4a}$$

$$\xi^2 = \frac{r^1 + r^2}{R} \qquad (-1 \leq \xi^2 \leq 1) \tag{4b}$$

$$\phi = \phi \qquad (0 \leq \phi \leq 2\pi) . \tag{4c}$$

---

* Pauli, Ann. Phys. (Leipzig) 68, 177 (1922).

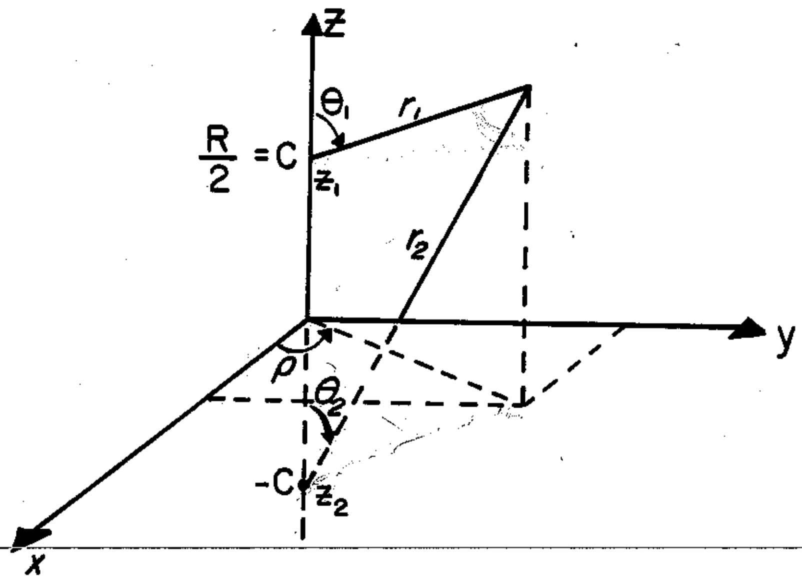


Fig. 1

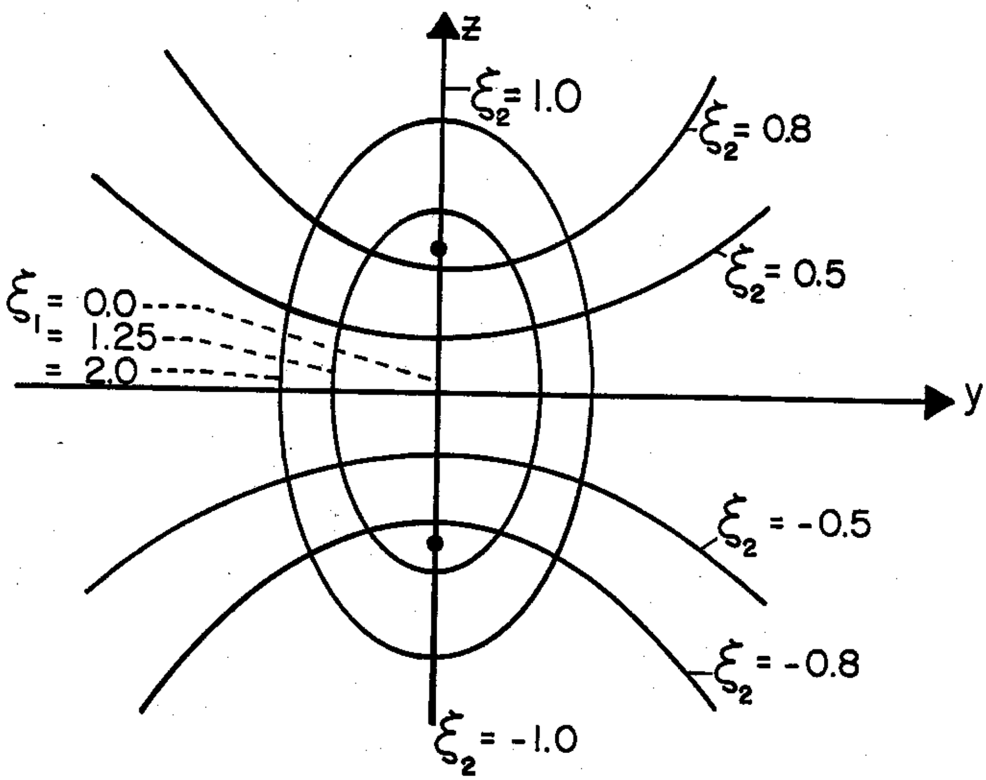


Fig. 2

Prolate Spheroidal Coordinates

The surfaces ξ^1 = constant, ξ^2 = constant are respectively ellipsoids and hyperboloids about the molecular axis. It is instructive to note at this point, that the prolate spheroidal coordinates reduce to the spherical polar system (r,θ,ϕ) in the limit in which $R = 0$ such that

$$r = \lim \tfrac{1}{2}\xi^1\xi^2 R, \qquad \cos\theta = \lim \xi^2.$$

In terms of these coordinates then, Schroedinger's equation is

$$\frac{4}{R^2[(\xi^1)^2 - (\xi^2)^2]}\{\frac{\partial}{\partial\xi^1}[(\xi^1)^2 - 1]\frac{\partial}{\partial\xi^1} + \frac{\partial}{\partial\xi^2}[1 - (\xi^2)^2]\frac{\partial}{\partial\xi^2} + [\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\frac{\partial^2}{\partial\phi^2}\}\psi + \frac{2m}{\hbar^2}[E - V]\psi = 0 \qquad (5)$$

where $V(\xi^1,\xi^2) = -\frac{2e^2}{R}[\frac{Z^1}{\xi^1 + \xi^2} + \frac{Z^2}{\xi^1 - \xi^2}]$. On multiplying through by $\tfrac{1}{2}R^2[(\xi^1)^2 - (\xi^2)^2]$ and expressing all units in terms of atomic units, the wave equation becomes

$$\{\frac{\partial}{\partial\xi^1}[(\xi^1)^2 - 1]\frac{\partial}{\partial\xi^1} + \frac{\partial}{\partial\xi^2}[1 - (\xi^2)^2]\frac{\partial}{\partial\xi^2} + [\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\frac{\partial^2}{\partial\phi^2} - p^2[(\xi^1)^2 - (\xi^2)^2] + R[Z^1 + Z^2]\xi^1 - R[Z^1 - Z^2]\xi^2\}\psi = 0 \qquad (6)$$

where $p^2 = -\tfrac{1}{2}ER^2$. Equation (6) is now separable if we assume our solution to be of the form $\psi(\xi^1,\xi^2,\phi) = X(\xi^1)Y(\xi^2)\Phi(\phi)$. Substituting this into (6), we get

$$Y(\xi^2)\Phi(\phi)\frac{\partial}{\partial\xi^1}[(\xi^1)^2 - 1]\frac{\partial}{\partial\xi^1}X(\xi^1) + X(\xi^1)\Phi(\phi)\frac{\partial}{\partial\xi^2}[1 - (\xi^2)^2]\frac{\partial}{\partial\xi^2}Y(\xi^2) + X(\xi^1)Y(\xi^2)[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\frac{\partial^2}{\partial\phi^2}\Phi(\phi) + \{-p^2[(\xi^1)^2 - (\xi^2)^2] + R[Z^1 + Z^2]\xi^1 - R[Z^1 - Z^2]\xi^2\}X(\xi^1)Y(\xi^2)\Phi(\phi) = 0 \qquad (7a)$$

dividing through by $\psi(\xi^1,\xi^2,\phi) = X(\xi^1)Y(\xi^2)\Phi(\phi)$, we get

$$\frac{1}{X(\xi^1)}\{\frac{\partial}{\partial\xi^1}[(\xi^1)^2 - 1]\frac{\partial}{\partial\xi^1}\}X(\xi^1) + \frac{1}{Y(\xi^2)}\{\frac{\partial}{\partial\xi^2}[1 - (\xi^2)^2]\frac{\partial}{\partial\xi^2}\}Y(\xi^2) + \frac{1}{\Phi(\phi)}[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\frac{\partial^2}{\partial\phi^2}\Phi(\phi) + \{-p^2[(\xi^1)^2 - (\xi^2)^2] + R[Z^1 + Z^2]\xi^1 - R[Z^1 - Z^2]\xi^2\} = 0 \qquad (7b)$$

Dividing through by $[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]$, (7b) becomes

$$\{X(\xi^1)[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\}^{-1}\{\frac{d}{d\xi^1}[(\xi^1)^2 - 1]\frac{d}{d\xi^1}\} X(\xi^1)$$
$$+ \{Y(\xi^2)[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]\}^{-1}\{\frac{d}{d\xi^2}[1 - (\xi^2)^2]\frac{d}{d\xi^2}\} Y(\xi^2) \quad (7c)$$
$$+ [\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}]^{-1} \{-p^2[(\xi^1)^2 - (\xi^2)^2] + R[Z^1 + Z^2]\xi^1 - R[Z^1 - Z^2]\xi^2\}$$
$$+ \frac{1}{\Phi(\phi)} \frac{d^2\Phi(\phi)}{d\phi^2} = 0.$$

We note that (7c) is separable in $\Phi(\phi)$ if we introduce the separation constant m^2, thus (7c) becomes

$$\frac{d^2\Phi(\phi)}{d\phi^2} = m^2\Phi(\phi) \quad (8a)$$

and

$$\frac{1}{X(\xi^1)} \{\frac{d}{d\xi^1}[(\xi^1)^2 - 1]\frac{d}{d\xi^1}\} X(\xi^1) + \frac{1}{Y(\xi^2)} \{\frac{d}{d\xi^2}[1 - (\xi^2)^2]\frac{d}{d\xi^2}\} Y(\xi^2)$$
$$+ \{-p^2[(\xi^1)^2 - (\xi^2)^2] + R[Z^1 + Z^2]\xi^1 - R[Z^1 - Z^2]\xi^2\} \quad (8b)$$
$$= m^2[\frac{1}{(\xi^1)^2 - 1} + \frac{1}{1 - (\xi^2)^2}].$$

In comparing (8a) to the separated Schroedinger equation of the hydrogen atom, we can see that this equation takes on the same form as that of the $\Phi(\phi)$ wave function for the hydrogen atom. Thus we have

$$\Phi(\phi) = e^{im\phi} \qquad m = 0, \pm 1, \pm 2, \ldots \quad (9)$$

Again, we can define a component of the angular momentum operator

$$\hat{L}_z = \frac{\hbar}{i} \frac{\partial}{\partial\phi} , \quad (10)$$

where, however, unlike the hydrogen atom, the choice of the z-axis is not arbitrary, but rather, is fixed by the orientation of the molecular axis as is seen in figure 1. The separation constant "m" can again be interpreted as the "magnetic quantum number" (remembering the restriction of the orientation of the molecule).

$$\hat{L}_z \Phi(\phi) = \frac{\hbar}{i} \frac{\partial}{\partial \phi} e^{im\phi} = m\hbar e^{im\phi} = m\hbar \Phi(\phi) \tag{11}$$

showing that the angular momentum along the molecular axis (z-axis) is quantized.

Equation (8b) can now be separated, introducing a second separation constant "A".

$$\left\{\frac{d}{d\xi^1}[(\xi^1)^2 - 1]\frac{d}{d\xi^1} - p^2(\xi^1)^2 + R[Z^1 + Z^2]\xi^1 - \frac{m}{(\xi^1)^2 - 1}\right\} X(\xi^1) = AX(\xi^1) \quad (12$$

$$\left\{\frac{d}{d\xi^2}[1 - (\xi^2)^2]\frac{d}{d\xi^2} + p^2(\xi^2)^2 - R[Z^1 - Z^2]\xi^2 - \frac{m}{1 - (\xi^2)^2}\right\} Y(\xi^2) = AY(\xi^2) \quad (12$$

It is precisely at this stage of the problem that the complications make themselves apparent. As can be seen in the ξ^1 and ξ^2 equations (12a) and (12b), each of the two equations involve the energy ($p^2 = -\frac{1}{2}ER^2$) and the separation constant "A". This makes the problem much more difficult to handle than the customary one in which each eigenvalue equation involves just one eigenvalue as is the case for the hydrogen atom.

Much work has been done by Wilson[30], Jaffe[20], and Baber and Hassé[4] in an attempt to find exact solutions to these equations. The result has been solutions in the form of (1) ascending power series in either odd or even powers of ξ^1 or ξ^2; (2) a series expansion in terms of associated Legendre polynomials; and (3) a power series solution in $(1 \pm \xi^1)$ or $(1 \pm \xi^2)$. Unfortunately, however, all three of these types of solutions lead to three-term recurrence relations for the coefficients of the expansion which require the evaluation of an infinite continued fraction to determine the allowed values of the separation constant "A" and allowed energy eigenvalues "E" as a function of the internuclear separation "R". The problem of evaluating this continued fraction and its associated recurrence relation proves to be a laborious task. Bates, Ledsham, and Stewart [5] have tabulated the sets of parameters needed to derive the wave functions for H_2^+.

Many approximation methods such as LCAO, UA (United Atom approximation), the variational method, and perturbation theory have been applied to the hydrogen molecular ion problem. They are most usefull for $R << 1$ or $R >> 1$. A quasiclassical description of the two-center problem by means of the WKB approximation has met with considerable success by employing the cannonically invariant Keller-Maslov quantization conditions. Work has been done in this area by Gershtein, Ponomarev, and Puzynina [16] and Strand and Reinhardt [27].

The classical two-center problem was shown by Euler [ ] to have a general integral of the motion. Erikson and Hill [15] showed this general integral to be of the form

$$\Omega = L' \cdot L'' + 2me^2c(Z^1\cos\theta^1 - Z^2\cos\theta^2) \tag{13}$$

in terms of the classical system, where L' and L" are the orbital angular momentum of the electron as referred to the two nuclei 1 and 2. The angles θ^1 and θ^2 are as defined in figure 1. The quantum analog of this constant of the motion was shown to be

$$\hat{\Omega} = \tfrac{1}{2}\hbar^{-2}(\hat{L}' \cdot \hat{L}'' + \hat{L}'' \cdot \hat{L}') + \gamma(Z^1\cos\theta^1 - Z^2\cos\theta^2) \tag{14}$$

where $\gamma \equiv 2me^2c/\hbar^2 = 2c/a_0$ and a_0 is the Bohr radius.

or

$$\hat{\Omega} = \hbar^{-2}\hat{L}^2 + c^2(\nabla^2 - \partial_z^2) + \gamma(Z^1\cos\theta^1 - Z^2\cos\theta^2). \quad (15)$$

$$= \frac{[1 - (\xi^2)^2]}{[(\xi^1)^2 - (\xi^2)^2]}\frac{\partial}{\partial\xi^1}\{[(\xi^1)^2 - 1]\frac{\partial}{\partial\xi^1}\}$$

$$- \frac{[(\xi^1)^2 - 1]}{[(\xi^1)^2 - (\xi^2)^2]}\frac{\partial}{\partial\xi^2}\{[1 - (\xi^2)^2]\frac{\partial}{\partial\xi^2}\}$$

$$+ [\frac{1}{(\xi^1)^2 - 1} - \frac{1}{1 - (\xi^2)^2}]\frac{\partial^2}{\partial\phi^2}$$

$$+ \gamma\frac{[Z^1 + Z^2]\xi^1[1 - (\xi^2)^2] + [Z^1 - Z^2]\xi^2[(\xi^2)^2 - 1]}{[(\xi^1)^2 - (\xi^2)^2]}.$$

It is seen that $\hat{H}, \hat{\Omega}$, and $\hat{L}_z \equiv \frac{\hbar}{i}\frac{\partial}{\partial\phi}$ form a complete set of commuting observables defining the stationary state of the system. Defining the operator

$$\hat{A} = -\hat{\Omega} - 2mc^2\hat{H}/h^2 \quad (16)$$

we find that

$$\hat{A}\psi = A\psi \quad (17)$$

where the eigenvalue "A" is seen to be the separation constant introduced in equations (12a) and (12b). Thus we can attribute the separablity of these equations to Ω. Furthermore, it now becomes apparent that the existence of this extra constant of the motion is also responsible for the observed crossing of energy levels of the electronic terms in H_2^+ which were supposedly forbidden by the Von Neumann - Wigner theorem.

The expression (16) for the operator $\hat{A}$ of the eigenvalue "A" is rather cumbersome. Coulson and Joseph [11] have given a much more aesthetically pleasing form for the operator $\hat{A}$

$$\hat{A} = \hat{L}^2 + c^2p_z^2 + \gamma z(\frac{Z^1}{r^1} - \frac{Z^2}{r^2}) \quad (18)$$

which closely resembles the form for the operator of the constant of the motion $\hat{\Omega}$. They have also shown that this extra constant of the motion persists for the arbitrary n-dimensional two-center problem of the more general Eddington potential (of which the harmonic oscillator is a particular case); however, does not exist for the multi-center problem.

6-2. General Formulation of the Classical Problem

The two-center problem was found to be separable in prolate spheroidal coordinates. The following is a conversion of the Hamiltonian function from cartesian to prolate spheroidal coordinates.

Defining the "extended Hamiltonian" $W \equiv H - E$, we have for the two-center Hamiltonian

$$W = \frac{1}{2m}(\vec{p}\cdot\vec{p}) - \left(\frac{Z^1}{r^1} + \frac{Z^2}{r^2}\right) + \frac{Z^1Z^2}{2c} - E. \tag{1}$$

In prolate spheroidal coordinates, we have

$$\xi^1 = \frac{r^1 + r^2}{2c}\ ; \quad \xi^2 = \frac{r^1 - r^2}{2c}\ ; \quad \xi^3 = \cos\phi \tag{2}$$

and the inverse transformation

$$r^1 = c(\xi^1 + \xi^2); \quad r^2 = c(\xi^1 - \xi^2); \quad \phi = \arccos\xi^3. \tag{3}$$

Thus, $V^1 \equiv -\left(\frac{Z^1}{r^1} + \frac{Z^2}{r^2}\right)$ can be rewritten in terms of prolate spheroidal coordinates

$$V^1 = -\frac{1}{c}\left[\frac{Z^1(\xi^1 - \xi^2) + Z^2(\xi^1 + \xi^2)}{(\xi^1)^2 - (\xi^2)^2}\right] \tag{4}$$

$$= -\frac{1}{c[(\xi^1)^2 - (\xi^2)^2]}[\xi^1(Z^1 + Z^2) - \xi^2(Z^1 - Z^2)] \tag{5}$$

and letting $\alpha^1 \equiv Z^1 + Z^2$, $\alpha^2 \equiv Z^1 - Z^2$, $Z^1Z^2 = \frac{(\alpha^1)^2 - (\alpha^2)^2}{4}$

$$V^1 = -\frac{1}{c[(\xi^1)^2 - (\xi^2)^2]}(\alpha^1\xi^1 - \alpha^2\xi^2). \tag{6}$$

and $V^2 \equiv \frac{Z^1Z^2}{2c}$ becomes

$$V^2 = \frac{(\alpha^1)^2 - (\alpha^2)^2}{8c}\ . \tag{7}$$

The conjugate momenta can be calculated from the kinetic energy

$$T = \tfrac{1}{2}m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) \tag{7}$$

where the transformation from cartesian to prolate spheroidal coordinates is

$$\begin{aligned} x &= c\xi^3[(\xi^1)^2 - 1]^{\frac{1}{2}}[1 - (\xi^2)^2]^{\frac{1}{2}} \\ y &= c[(\xi^1)^2 - 1]^{\frac{1}{2}}[1 - (\xi^2)^2]^{\frac{1}{2}}[1 - (\xi^3)^2]^{\frac{1}{2}} \\ z &= c\xi^1\xi^2. \end{aligned} \tag{8}$$

And

$$\begin{aligned} \dot{x} &= \frac{dx}{dt} = \frac{\partial x}{\partial \xi^1}\dot{\xi}^1 + \frac{\partial x}{\partial \xi^2}\dot{\xi}^2 + \frac{\partial x}{\partial \xi^3}\dot{\xi}^3 \\ &= \frac{c}{[(\xi^1)^2 - 1]^{\frac{1}{2}}[1 - (\xi^2)^2]^{\frac{1}{2}}}\{\xi^3\xi^1\dot{\xi}^1[1 - (\xi^2)^2] - \xi^3\xi^2\dot{\xi}^2[(\xi^1)^2 - 1] \\ &\qquad + \dot{\xi}^3[(\xi^1)^2 - 1][1 - (\xi^2)^2]\} \end{aligned} \tag{9}$$

$$\begin{aligned} \dot{y} &= \frac{dy}{dt} = \frac{\partial x}{\partial \xi^1}\dot{\xi}^1 + \frac{\partial x}{\partial \xi^2}\dot{\xi}^2 + \frac{\partial x}{\partial \xi^3}\dot{\xi}^3 \\ &= \frac{c}{[(\xi^1)^2 - 1]^{\frac{1}{2}}[1 - (\xi^2)^2]^{\frac{1}{2}}[1 - (\xi^3)^2]^{\frac{1}{2}}} \times \\ &\quad \{\xi^1\dot{\xi}^1[1 - (\xi^2)^2][1 - (\xi^3)^2] - \xi^2\dot{\xi}^2[(\xi^1)^2 - 1][1 - (\xi^3)^2] \\ &\qquad - \xi^3\dot{\xi}^3[(\xi^1)^2 - 1][1 - (\xi^2)^2]\} \end{aligned} \tag{10}$$

$$\begin{aligned} \dot{z} &= \frac{dz}{dt} = \frac{\partial z}{\partial \xi^1}\dot{\xi}^1 + \frac{\partial z}{\partial \xi^2}\dot{\xi}^2 \\ &= c[\xi^2\dot{\xi}^1 + \xi^1\dot{\xi}^2]\ . \end{aligned} \tag{11}$$

Thus,

$$\begin{aligned} v^2 &= \frac{c^2}{[(\xi^1)^2 - 1][1 - (\xi^2)^2]}\{\xi^3\xi^1\dot{\xi}^1[1 - (\xi^2)^2] - \xi^3\xi^2\dot{\xi}^2[(\xi^1)^2 - 1] \\ &\qquad + \dot{\xi}^3[(\xi^1)^2 - 1][1 - (\xi^2)^2]\}^2 \\ &+ \frac{c^2}{[(\xi^1)^2 - 1][1 - (\xi^2)^2][1 - (\xi^3)^2]}\{\xi^1\dot{\xi}^1[1 - (\xi^2)^2][1 - (\xi^3)^2] \\ &\qquad - \xi^2\dot{\xi}^2[(\xi^1)^2 - 1][1 - (\xi^3)^2] - \xi^3\dot{\xi}^3[(\xi^1)^2 - 1][1 - (\xi^2)^2]\}^2 \\ &+ c^2[\xi^2\dot{\xi}^1 + \xi^1\dot{\xi}^2]^2 \end{aligned} \tag{12}$$

or

$$\left(\frac{v}{c}\right)^2 = \frac{1}{AB}\,[\xi^3\xi^1\dot{\xi}^1 B - \xi^3\xi^2\dot{\xi}^2 A + \dot{\xi}^3 AB]^2 + \frac{1}{ABC}\,[\xi^1\dot{\xi}^1 BC - \xi^2\dot{\xi}^2 AC - \xi^3\dot{\xi}^3 AB]^2 \quad (13)$$

$$+ [\xi^2\dot{\xi}^1 + \xi^1\dot{\xi}^2]^2$$

$$= \frac{(\dot{\xi}^1)^2}{A}[A + B] + \frac{(\dot{\xi}^2)^2}{B}[A + B] + \frac{(\dot{\xi}^3)^2}{C}[AB] \quad (14)$$

where $A = [(\xi^1)^2 - 1]$; $B = [1 - (\xi^2)^2]$; $C = [1 - (\xi^3)^2]$.

Thus, we have for the conjugate momenta

$$p^i = \frac{T(\xi^i, \dot{\xi}^i)}{\dot{\xi}^i} = \tfrac{1}{2}m\,\frac{v^2(\xi^i, \dot{\xi}^i)}{\dot{\xi}^i} \quad (15)$$

$$p^1 = \tfrac{1}{2}mc^2\dot{\xi}^1\frac{[A + B]}{A} \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow (p^1)^2 = m^2c^4(\dot{\xi}^1)^2\frac{[A + B]^2}{4A^2} \quad (16)$$

$$p^2 = \tfrac{1}{2}mc^2\dot{\xi}^2\frac{[A + B]}{B} \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow (p^2)^2 = m^2c^4(\dot{\xi}^2)^2\frac{[A + B]^2}{4B^2} \quad (17)$$

$$p^3 = \tfrac{1}{2}mc^2\dot{\xi}^3\frac{AB}{C} \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow (p^3)^2 = m^2c^4(\dot{\xi}^3)^2\frac{[AB]^2}{4C^2} \quad (18)$$

and the kinetic energy in terms of prolate spheroidal coordinates and their conjugate momenta becomes

$$T = \frac{1}{2mc^2[(\xi^1)^2 - (\xi^2)^2]}\,\{[(\xi^1)^2 - 1](p^1)^2 + [1 - (\xi^2)^2](p^2)^2 \quad (19)$$

$$+ \frac{[1 - (\xi^3)^2][(\xi^1)^2 - (\xi^2)^2](p^3)^2}{[(\xi^1)^2 - 1][1 - (\xi^2)^2]}\}.$$

Employing a canonical transformation on p^3,

$$(\bar{p}^3)^2 = (p^3)^2[1 - (\xi^3)^2] \; ; \quad \bar{p}^3 = p^3[1 - (\xi^3)^2]^{\frac{1}{2}} \quad (20)$$

$$\bar{\xi}^3 = \frac{\xi^3}{[1 - (\xi^3)^2]^{\frac{1}{2}}} = \frac{\cos\phi}{[1 - \cos^2\phi]^{\frac{1}{2}}} = \cot\phi$$

then (19) becomes

$$T = \frac{1}{2mc^2[(\xi^1)^2 - (\xi^2)^2]}\,\{[(\xi^1)^2 - 1](p^1)^2 + [1 - (\xi^2)^2](p^2)^2 \quad (21)$$

$$+ \frac{[(\xi^1)^2 - (\xi^2)^2](\bar{p}^3)^2}{[(\xi^1)^2 - 1][1 - (\xi^2)^2]}\}.$$

Equations (16), (17), and (18) are then

$$p^1 = mc^2\dot{\xi}^1\frac{[(\xi^1)^2 - (\xi^2)^2]}{[(\xi^1)^2 - 1]} \tag{22}$$

$$p^2 = mc^2\dot{\xi}^2\frac{[(\xi^1)^2 - (\xi^2)^2]}{[1 - (\xi^2)^2]} \tag{23}$$

$$\bar{p} = -mc^2\dot{\phi}\frac{[(\xi^1)^2 - 1][1 - (\xi^2)^2]}{\sin\phi} \equiv p_\phi \tag{24}$$

and our Hamiltonian is

$$\begin{aligned} W &= H - E \\ &= \frac{1}{2mc^2[(\xi^1)^2 - (\xi^2)^2]}\{[(\xi^1)^2 - 1](p^1)^2 + [1 - (\xi^2)^2](p^2)^2 + \\ &\qquad \frac{[(\xi^1)^2 - (\xi^2)^2]}{[(\xi^1)^2 - 1][1 - (\xi^2)^2]}(\bar{p}^3)^2\} \\ &\quad - \frac{1}{c[(\xi^1)^2 - (\xi^2)^2]}[\alpha^1\xi^1 - \alpha^2\xi^2] + \frac{1}{8c}[(\alpha^1)^2 - (\alpha^2)^2] - E \qquad (25) \\ &= \frac{1}{[(\xi^1)^2 - (\xi^2)^2]}[H^1 + H^2] + \frac{1}{8c}[(\alpha^1)^2 - (\alpha^2)^2] - E \end{aligned}$$

where

$$H^1 = \frac{1}{2mc^2}\{[(\xi^1)^2 - 1](p^1)^2 + \frac{(\bar{p}^3)^2}{[(\xi^1)^2 - 1]} - \frac{\alpha^1\xi^1}{c}$$

$$H^2 = \frac{1}{2mc^2}\ [1 - (\xi^2)^2](p^2)^2 + \frac{(\bar{p}^3)^2}{[1 - (\xi^2)^2]} - \frac{\alpha^2\xi^2}{c}.$$

This Hamiltonian is undefined for $\xi^1 = \pm\xi^2$. Multiplying through by $c^2[(\xi^1)^2 - (\xi^2)^2]$ we obtain a new Hamiltonian

$$\begin{aligned} \bar{W} &= c^2[(\xi^1)^2 - (\xi^2)^2]W \qquad (26) \\ &= c^2[H^1 + H^2] + \frac{c}{8}[(\xi^1)^2 - (\xi^2)^2][(\alpha^1)^2 - (\alpha^2)^2] - Ec^2[(\xi^1)^2 - (\xi^2)^2] \\ &= \bar{W}^1 + \bar{W}^2 \end{aligned}$$

where

$$\bar{W}^1 = \frac{1}{2m}\{[(\xi^1)^2 - 1](p^1)^2 + \frac{(\bar{p}^3)^2}{[(\xi^1)^2 - 1]} + c(\xi^1)^2\frac{[(\alpha^1)^2 - (\alpha^2)^2]}{8} - Ec$$

$$- c\alpha^1\xi^1$$

$$\bar{W}^2 = \frac{1}{2m}\{[1 - (\xi^2)^2](p^2)^2 + \frac{(\bar{p}^3)^2}{[1 - (\xi^2)^2]} - c(\xi^2)^2 \frac{[(\alpha^1)^2 - (\alpha^2)^2]}{8} - Ec\} + c\alpha^2\xi^2.$$

which is well defined everywhere. Setting

$$\gamma \equiv c\{\frac{[(\alpha^1)^2 - (\alpha^2)^2]}{8} - Ec\} \tag{27}$$

then

$$\begin{aligned} \bar{W}^1 &= \frac{1}{2m}\{[(\xi^1)^2 - 1](p^1)^2 + \frac{(\bar{p}^3)^2}{[(\xi^1)^2 - 1]}\} + (\xi^1)^2\gamma - c\alpha^1\xi^1 \\ \bar{W}^2 &= \frac{1}{2m}\{[1 - (\xi^2)^2](p^2)^2 + \frac{(\bar{p}^3)^2}{[1 - (\xi^2)^2]}\} - (\xi^2)^2\gamma + c\alpha^2\xi^2. \end{aligned} \tag{28}$$

This can be expressed in terms of trigonometric and hyperbolic functions under the transformation

$$\begin{aligned} \xi^1 &\rightarrow\rightarrow\rightarrow \mu = \cosh^{-1}(\xi^1); & 0 \leq \mu < \infty \\ \xi^2 &\rightarrow\rightarrow\rightarrow \theta = \cos^{-1}(\xi^2); & 0 \leq \theta \leq \pi \end{aligned} \tag{29}$$

and our Hamiltonian functions become

$$\begin{aligned} \bar{W} &= \bar{W}_\mu + \bar{W}_\theta \\ \bar{W}_\mu &= \frac{1}{2m}[p_\mu^2 + \frac{p_\phi^2}{\sinh^2\mu}] + \gamma\cosh^2\mu - c\alpha^1\cosh\mu \\ \bar{W}_\theta &= \frac{1}{2m}[p_\theta^2 + \frac{p_\phi^2}{\sin^2\phi}] - \gamma\cos^2\theta + c\alpha^2\cos\theta. \end{aligned} \tag{30}$$

As any diffeomorphic system admits the same Lie algebra as the original system, we may then study the Lie algebras admitted by the diffeomorphic manifold defined by $\bar{W} = 0$. In this way we may uncover the group symmetries admitted by the flow defined by Hamilton's equations on the original manifold $W = 0$.

6-3. Topology

For the purpose of a group-theoretical analysis, the Hamiltonian of interest for the planar Kepler two-center problem ($p^3 = 0$) is given by (28) or (30) of the previous section as

$$\bar{W} = \bar{W}^1 + \bar{W}^2$$

$$\bar{W}^1 = \frac{1}{2m}[(\xi^1)^2 - 1](p^1)^2 + (\xi^1)^2\gamma - c\alpha^1\xi^1$$

$$\bar{W}^2 = \frac{1}{2m}[1 - (\xi^2)^2](p^2)^2 - (\xi^2)^2\gamma + c\alpha^2\xi^2 \qquad (1a)$$

for $1 \leq \xi^1 < \infty$; $-1 < \xi^2 < 1$

or, in terms of trigonometric and hyperbolic functions ($p_\phi = 0$)

$$\bar{W} = \bar{W}_\mu + \bar{W}_\theta$$

$$\bar{W}_\mu = \frac{1}{2m}p_\mu^2 + \gamma\cosh^2\mu - c\alpha^1\cosh\mu$$

$$\bar{W}_\theta = \frac{1}{2m}p_\theta^2 - \gamma\cos^2\theta + c\alpha^2\cos\theta \qquad (1b)$$

for $0 \leq \mu < \infty$; $0 \leq \theta \leq \pi$.

The effective potential then becomes

$$V^{eff} = (\xi^1)^2\gamma - c\alpha^1\xi^1 - (\xi^2)^2\gamma + c\alpha^2\xi^2$$

$$= \gamma[(\xi^1)^2 - (\xi^2)^2] - c[\alpha^1\xi^1 - \alpha^2\xi^2] \qquad (2a)$$

or

$$V^{eff} = \gamma\cosh^2\mu - c\alpha^1\cosh\mu - \gamma\cos^2\theta + c\alpha^2\cos\theta$$

$$= \gamma(\cosh^2\mu - \cos^2\theta) - c(\alpha^1\cosh\mu - \alpha^2\cos\theta). \qquad (2b)$$

It is instructive to examine the topology of these functions. Figures 6-1 to 6-7 show the 3D plots of $\bar{W}^1$ and $\bar{W}^2$ (m=c=1) where the nuclear repulsion term is omitted (i.e. $\gamma \rightarrow \gamma' = -Ec^2 = -E$). The special cases considered are the hydrogen molecular ion and the hydrogen atom. For H_2^+ ($\alpha^1 = Z^1 + Z^2 = 2$; $\alpha^2 = Z^1 - Z^2 = 0$) we have

for $E<0$ (bound)

$$\bar{W}^1 = \tfrac{1}{2}(p^1)^2[(\xi^1)^2 - 1] + (\xi^1)^2 - 2\xi^1 \tag{3a}$$

$$\bar{W}^2 = \tfrac{1}{2}(p^2)^2[1 - (\xi^2)^2] - (\xi^2) \tag{3b}$$

for $E>0$ (unbound)

$$\bar{W}^1 = \tfrac{1}{2}(p^1)^2[(\xi^1)^2 - 1] - (\xi^1)^2 - 2\xi^1 \tag{4a}$$

$$\bar{W}^2 = \tfrac{1}{2}(p^2)^2[1 - (\xi^2)^2] + (\xi^2)^2 \tag{4b}$$

and for the hydrogen atom ($Z^2 = 0 \rightarrow \alpha^1 = Z^1 = 1$; $\alpha^2 = Z^1 = 1$) we have

for $E<0$ (bound)

$$\bar{W} = \tfrac{1}{2}(p^1)^2[(\xi^1)^2 - 1] + (\xi^1)^2 - \xi^1 \tag{5a}$$

$$\bar{W} = \tfrac{1}{2}(p^2)^2[1 - (\xi^2)^2] - (\xi^2)^2 - \xi^2 \tag{5b}$$

and for $E>0$ (unbound)

$$\bar{W} = \tfrac{1}{2}(p^1)^2[(\xi^1)^2 - 1] - (\xi^1)^2 - \xi^1 \tag{6a}$$

$$\bar{W} = \tfrac{1}{2}(p^2)^2[1 - (\xi^2)^2] + (\xi^2)^2 - \xi^2. \tag{6b}$$

Equation (5a) and (6a) are not plotted, in that they are almost identical to (3a) and (4a) respectively.

The effective potential is plotted in figures 6-8 to 6-13 for various values of the parameters Z^1 and Z^2 which also encompasses the special cases of H_2^+ ($Z^1 = Z^2 = 1$) and the hydrogen atom ($Z^2 = 0$). In these following cases:

$E<0$ (bound)

$$V^{eff} = \cosh^2\mu - 2\cosh\mu - \cos^2\theta \quad ; \quad Z^1 = Z^2 = 1 \tag{7a}$$

$$V^{eff} = \cosh^2\mu - 3\cosh\mu - \cos^2\theta - \cos\theta \quad ; \quad Z^1 = 1,\ Z^2 = 2 \tag{7b}$$

$$V^{eff} = \cosh^2\mu - 3\cosh\mu - \cos^2\theta + \cos\theta \quad ; \quad Z^1 = 2,\ Z^2 = 1 \qquad (7c)$$

E>0 (unbound)

$$V^{eff} = -\cosh^2\mu - 2\cosh\mu + \cos^2\theta \quad ; \quad Z^1 = Z^2 = 1 \qquad (8a)$$

$$V^{eff} = -\cosh^2\mu - 3\cosh\mu + \cos^2\theta - \cos\theta \quad ; \quad Z^1 = 1,\ Z^2 = 2 \qquad (8b)$$

$$V^{eff} = -\cosh^2\mu - 3\cosh\mu + \cos^2\theta + \cos\theta \quad ; \quad Z^1 = 2,\ Z^2 = 1 \qquad (8c)$$

there seem to be no significant changes in the topology. All surfaces are diffeomorphic to the plane in 3-space. Furthermore, it can be seen that in the region of extrema, these surfaces resemble parabaloids. Since parabaloids are characteristic of the cross-sectional surfaces for the Hamiltonian of the two-dimensional harmonic oscillator ($H_{osc} = \frac{1}{2}[(p^1)^2 + (p^2)^2 + k^1(q^1)^2 + k^2(q^2)^2]$), these graphs suggest that, with a proper choice of origin, the harmonic oscillator may serve as a good approximation for the two-center problem.

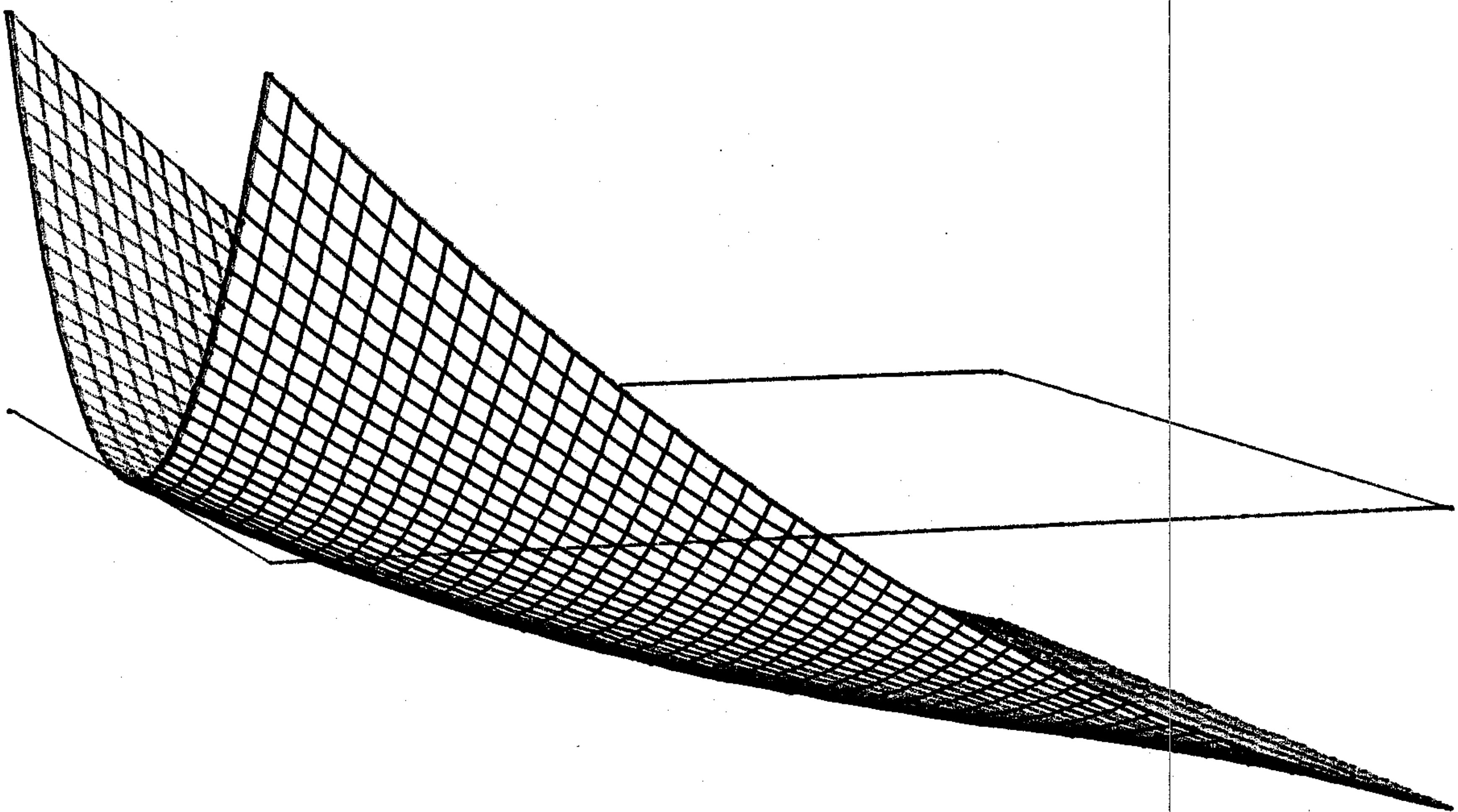


fig. 6-1a. H_2^+ (c.f. eqn. 3a)

$-1 < P_1 < 1$

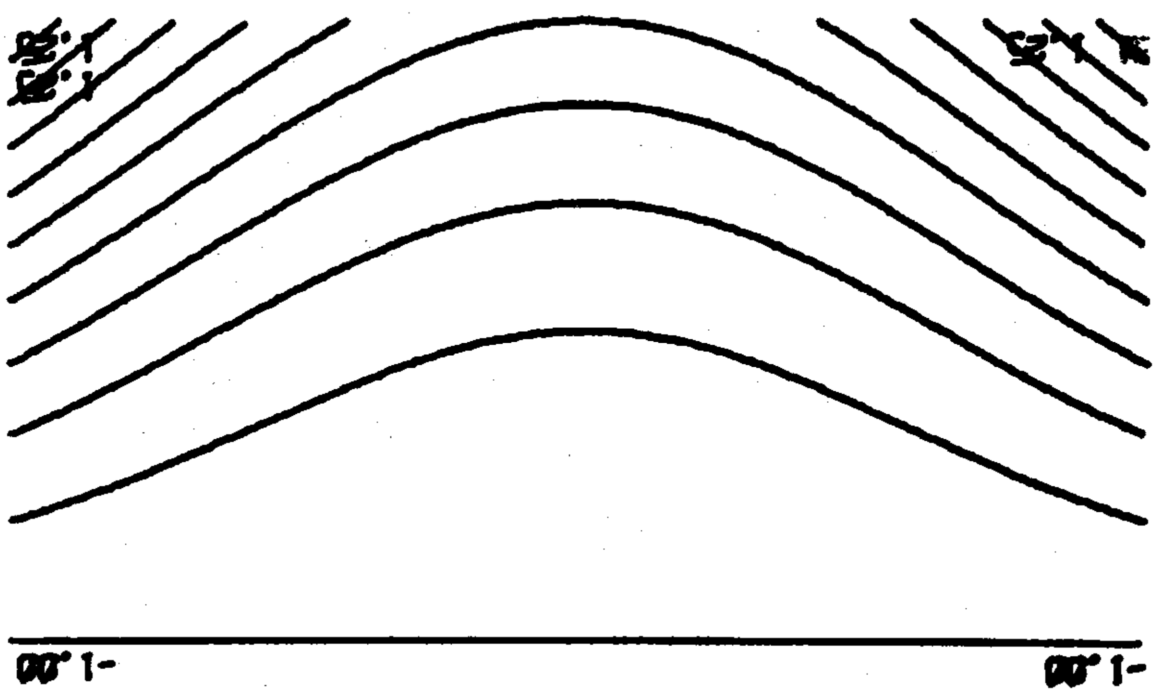


fig. 6-1b.

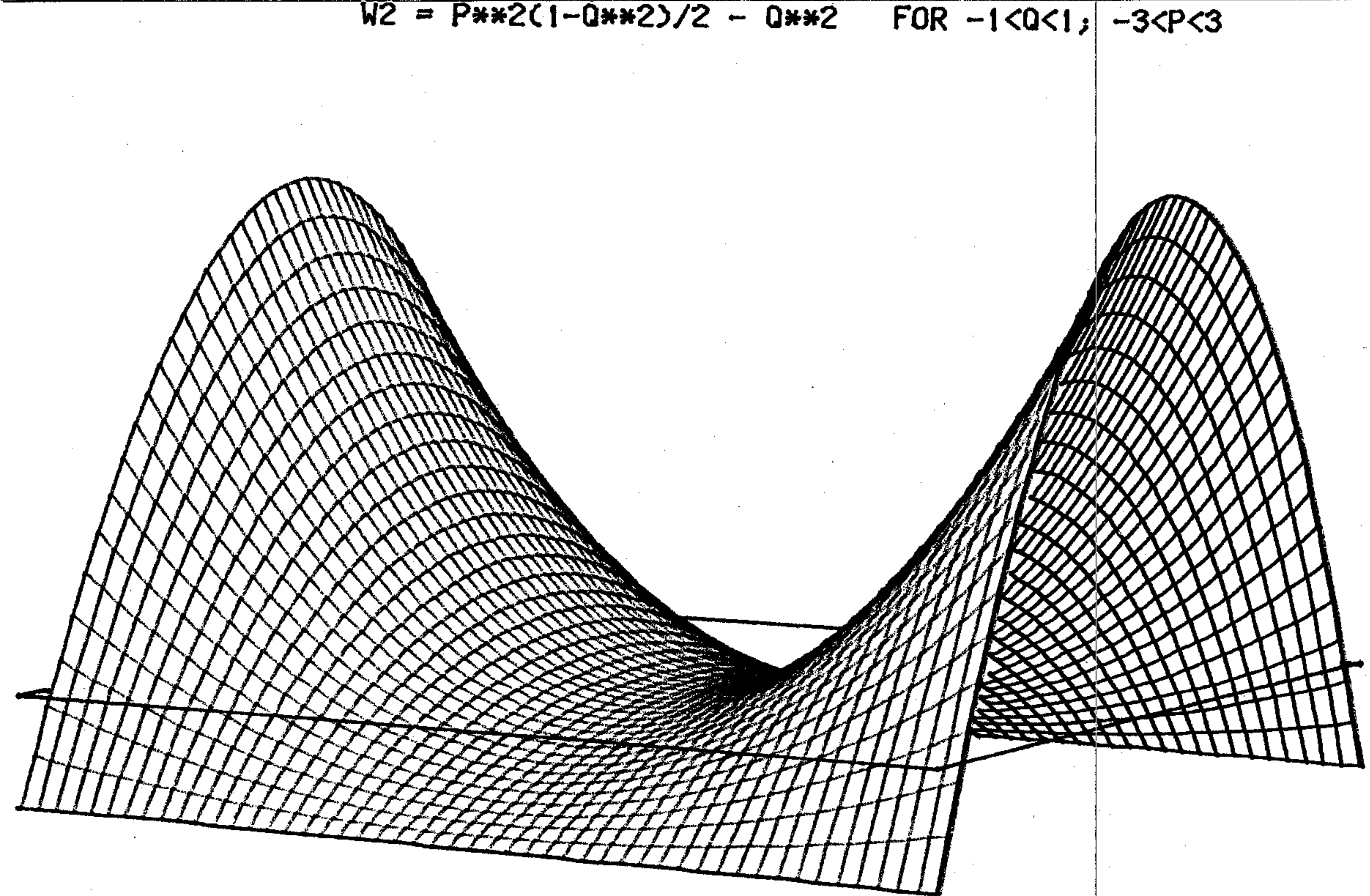


fig 6–2a. H_2^+ (c.f. eqn. 3b)

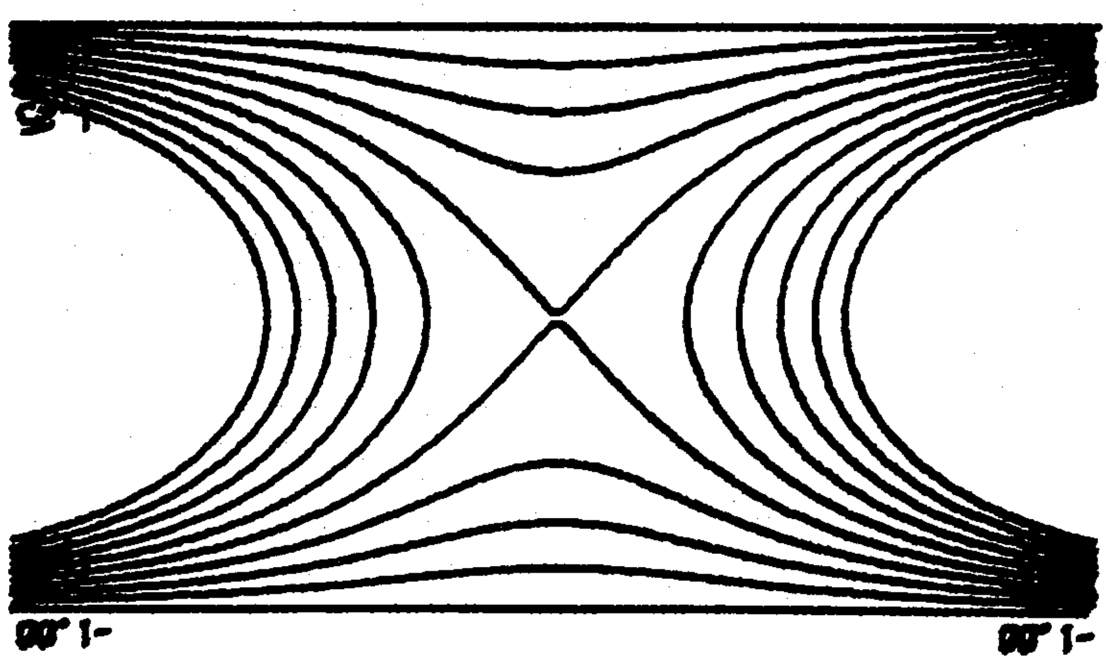


fig. 6-2b.

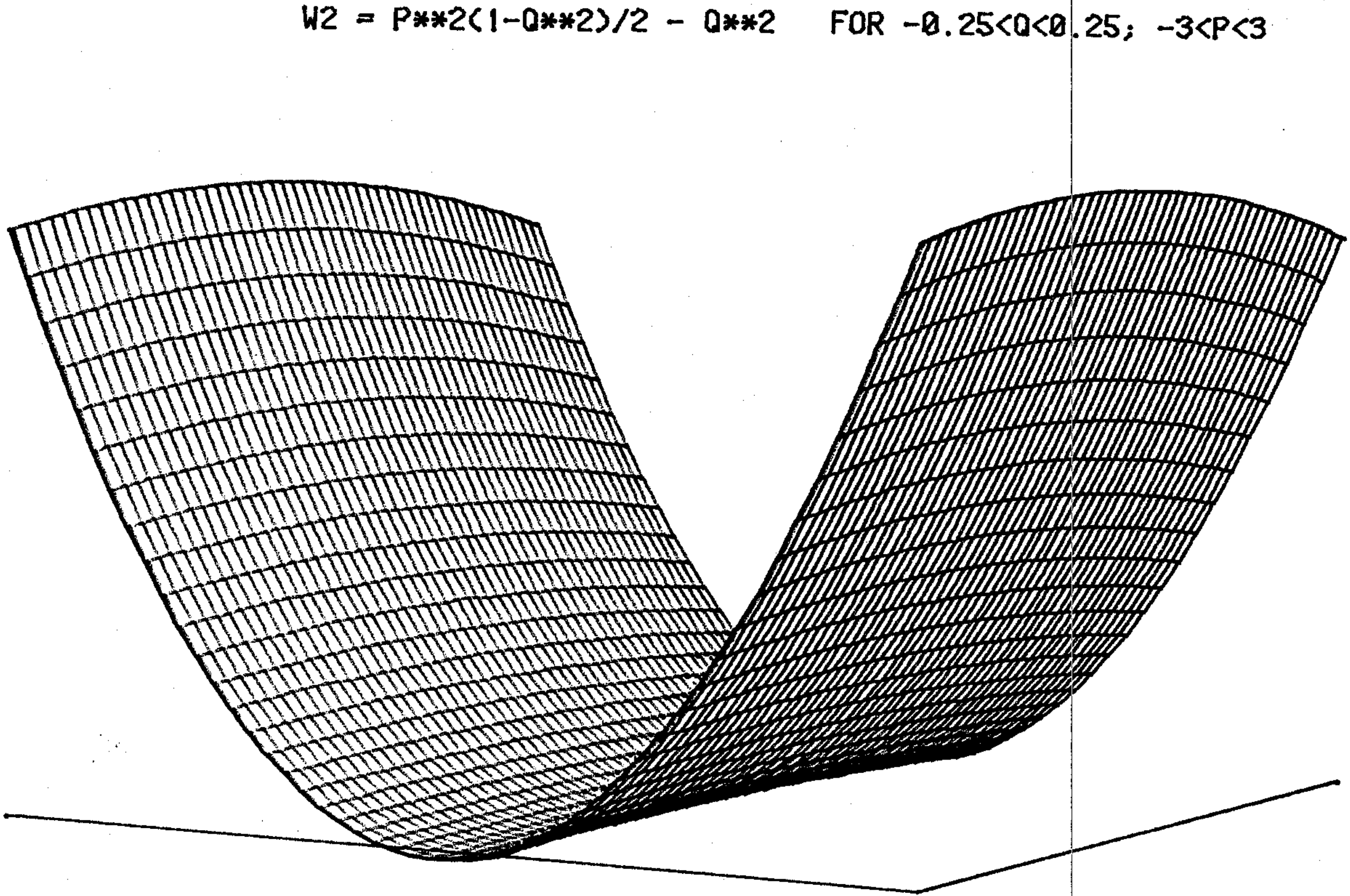


fig. 6-3a. H_2^+ (c.f. eqn. 3b)

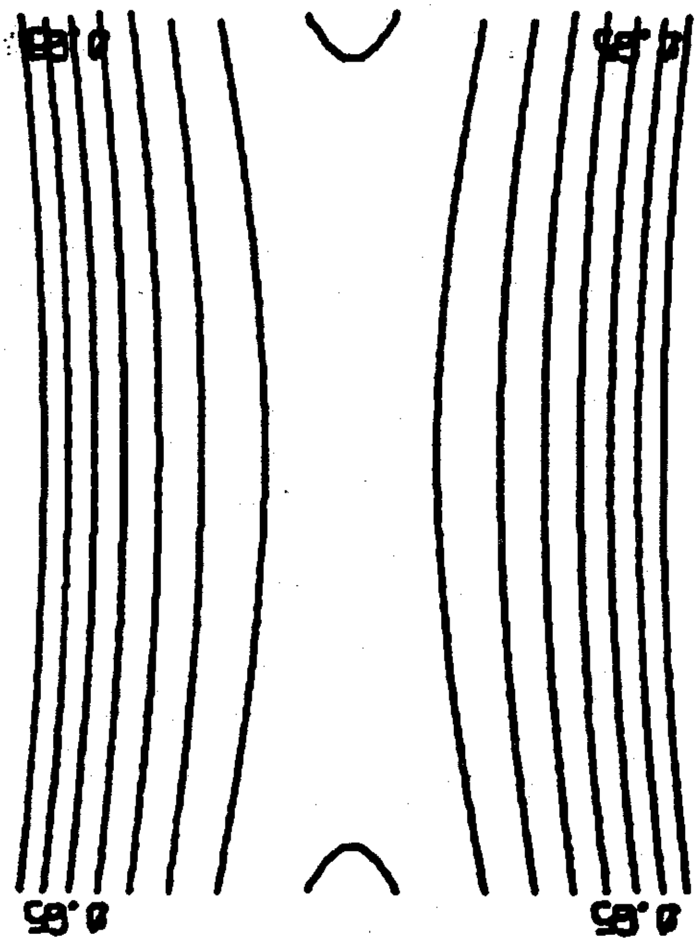


fig. 6-3b.

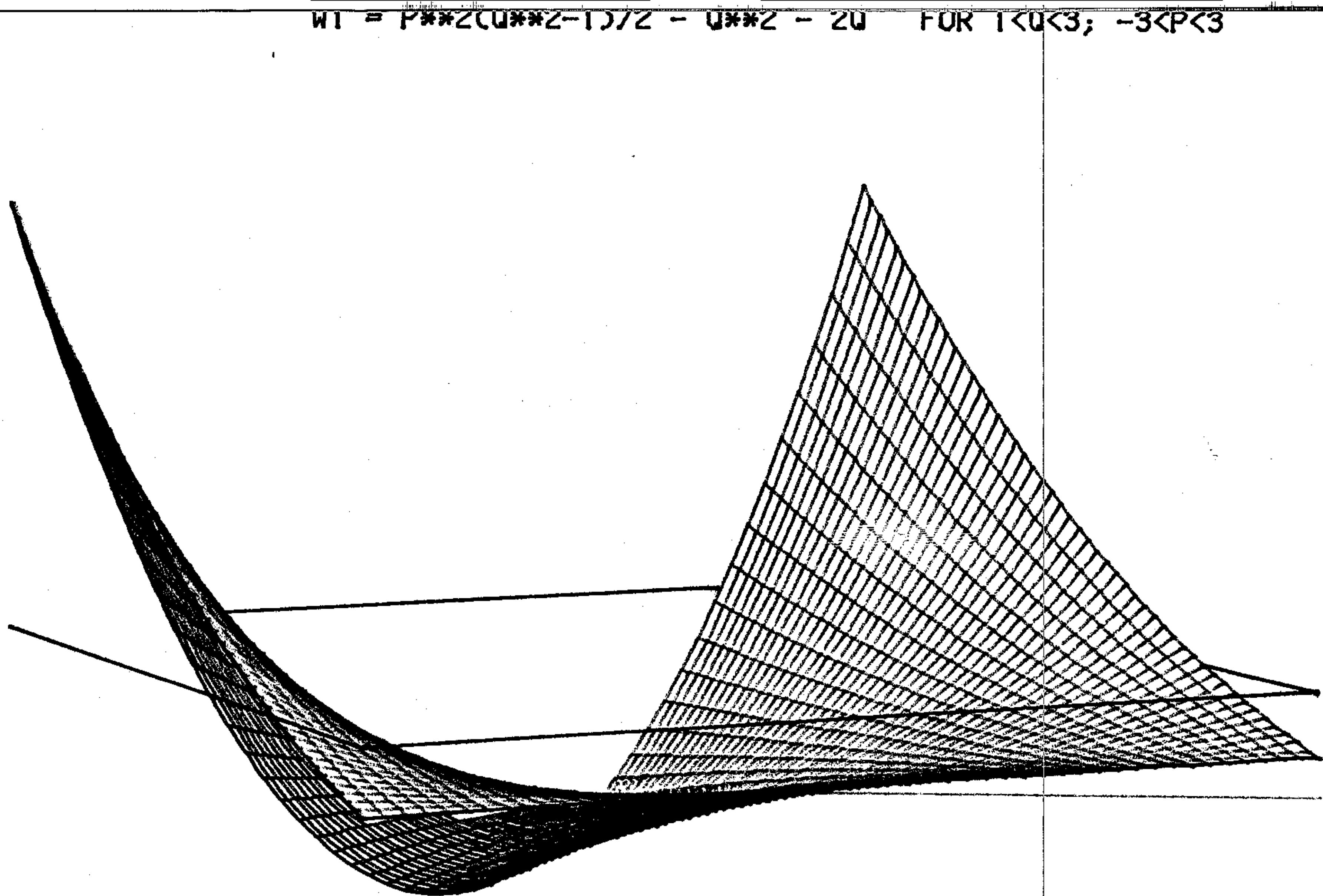


fig. 6-4a. H_2^+ (c.f. eqn. 4a)

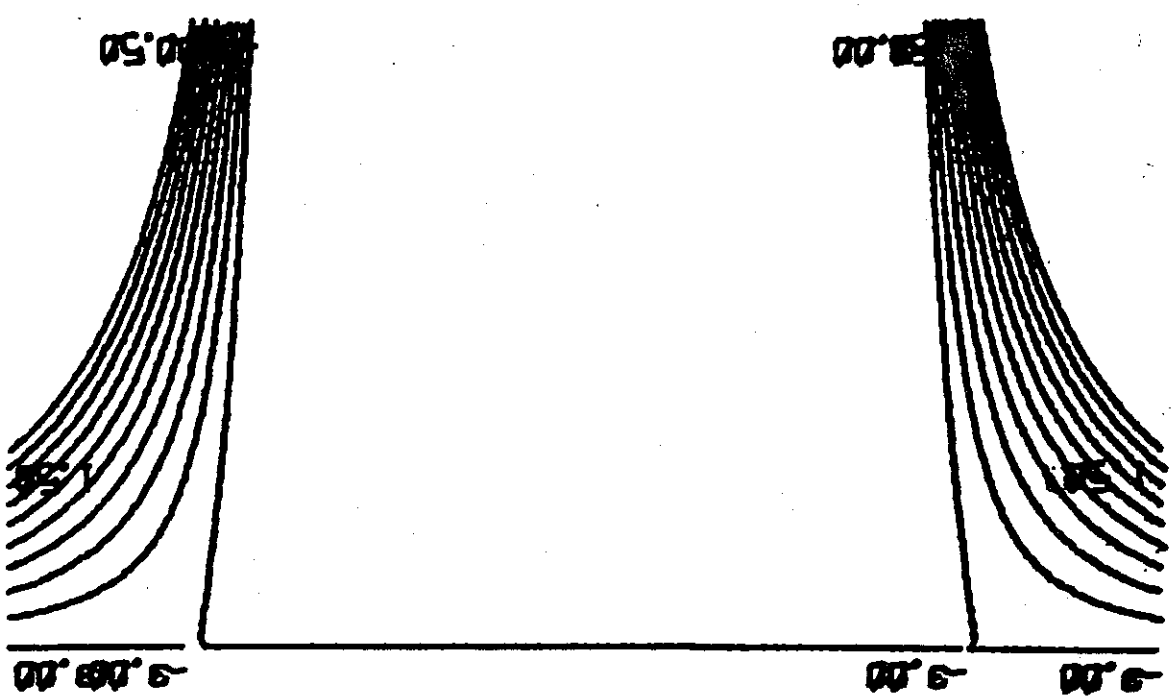


fig. 6-4b.

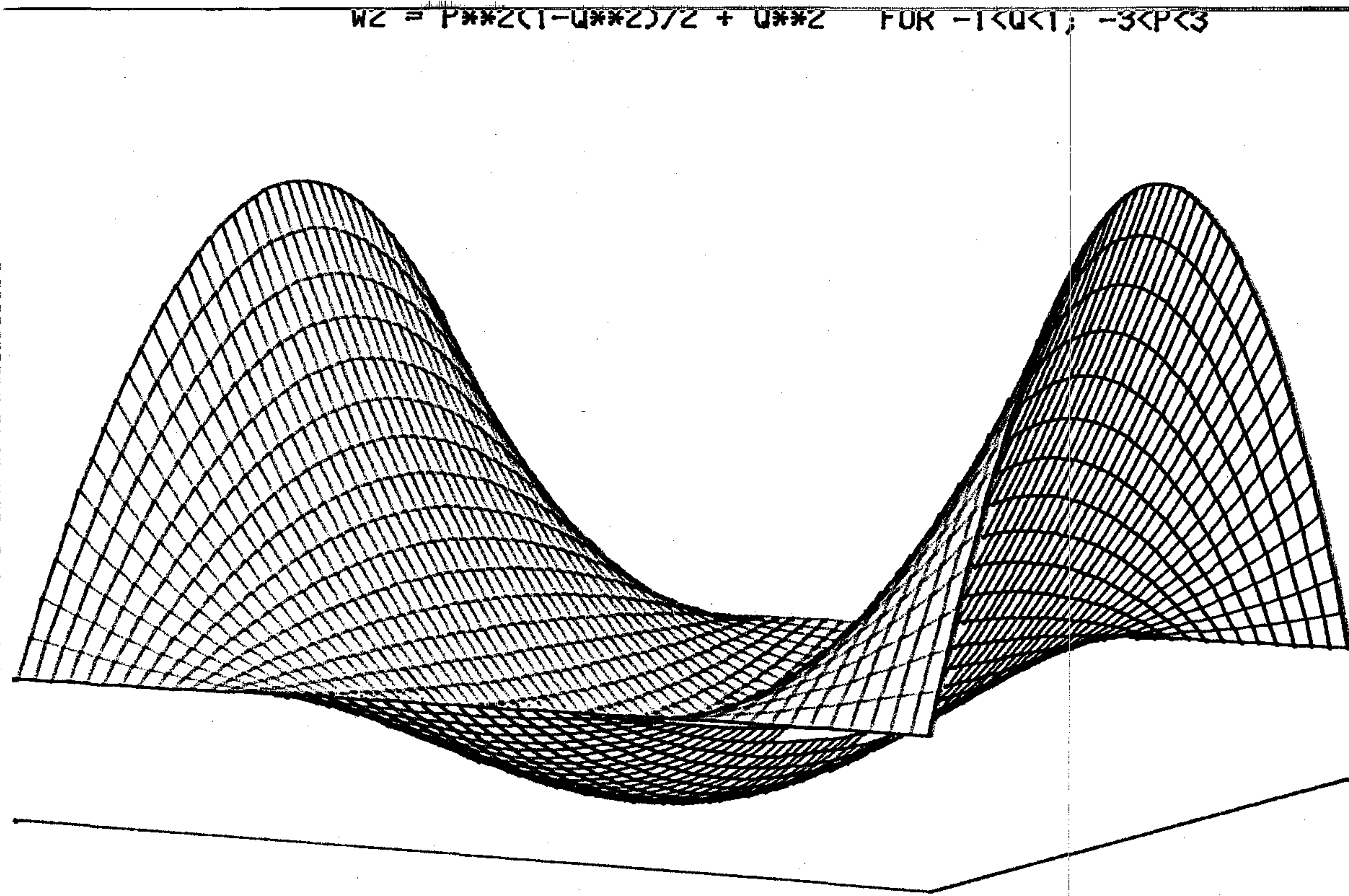


fig. 6-5a. H_2^+ (c.f. eqn. 4b)

$\omega_2 = \frac{1}{2} p_2 (1 - \xi_2) + \xi_2$

$-3 < p_2 < 3$

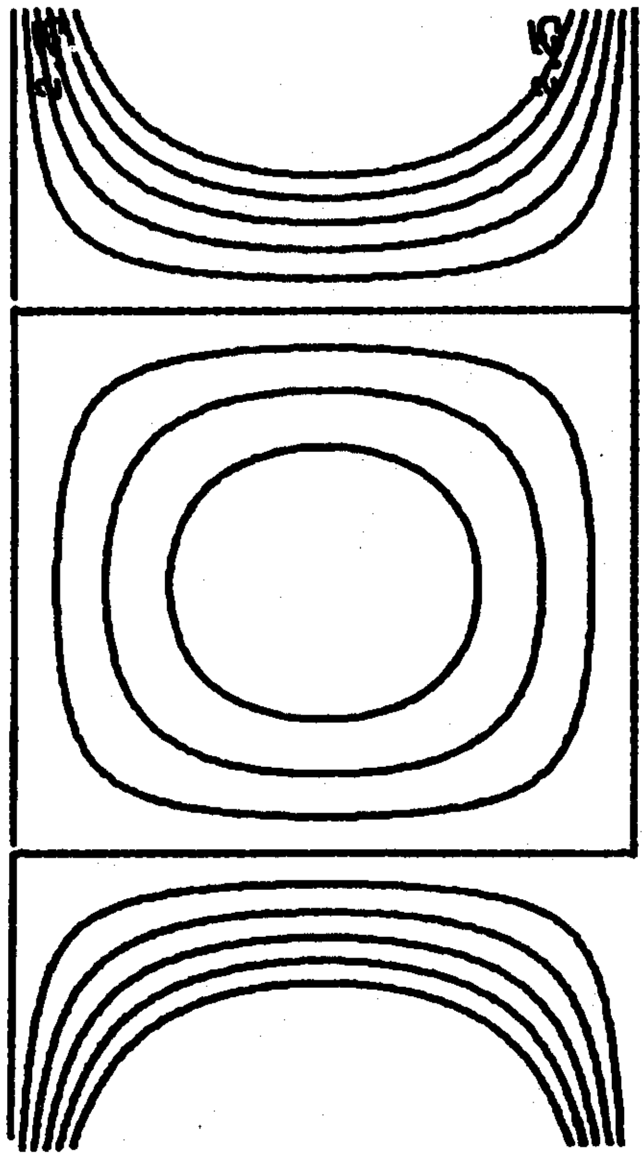


fig. 6-5b

W2 = P**2(1-Q**2)/2 - Q**2 - Q FOR

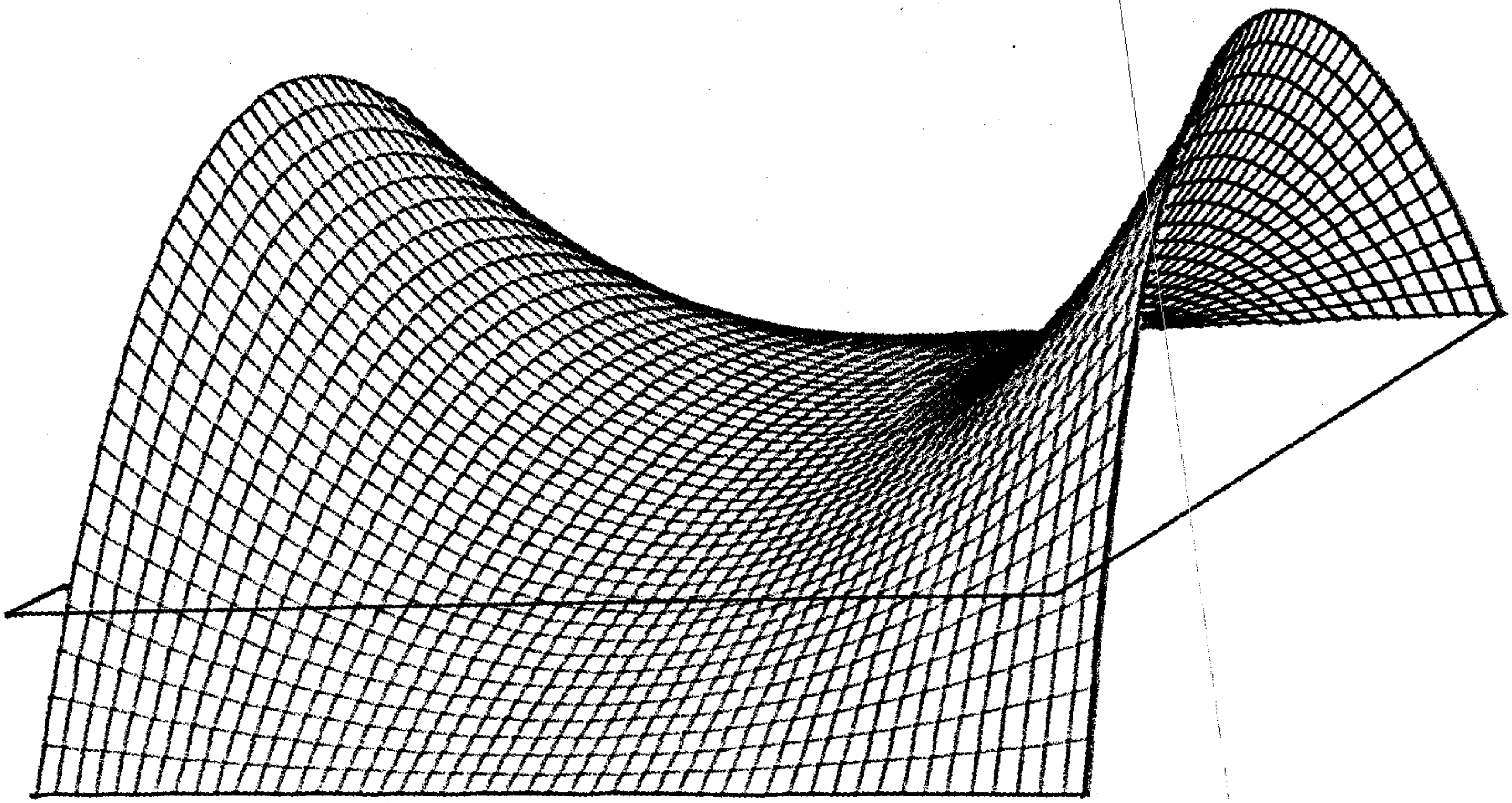


fig. 6-6a.
Hydrogen atom (c.f. eqn. 5b)

$-3 < p_2 < 3$

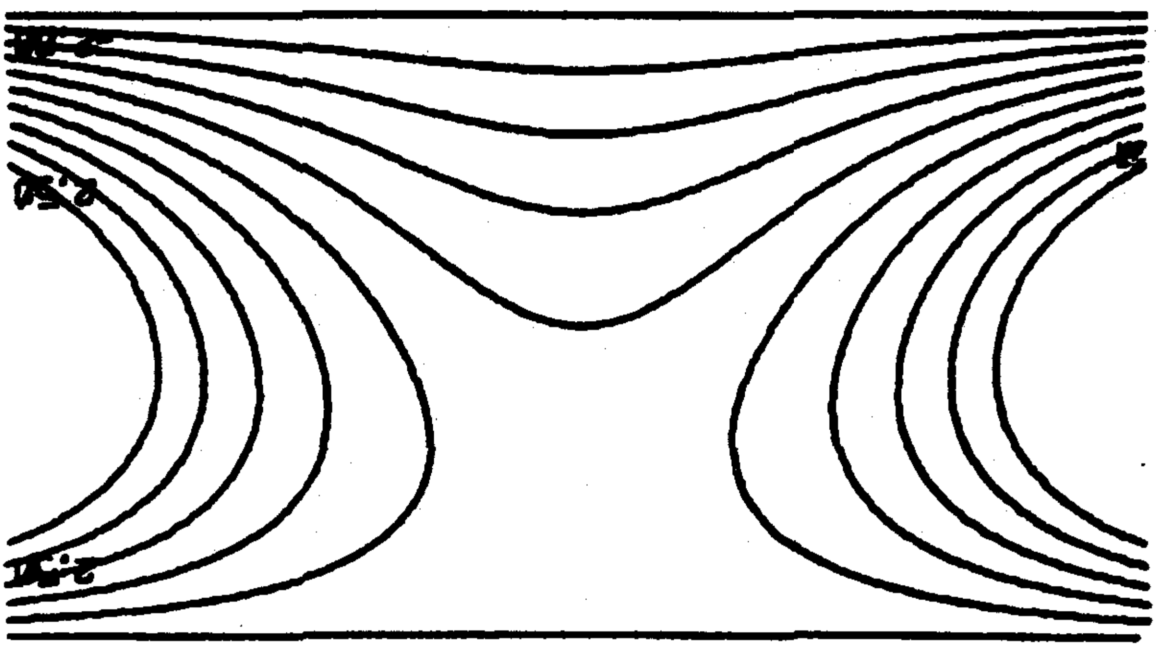


fig. 6-6b.

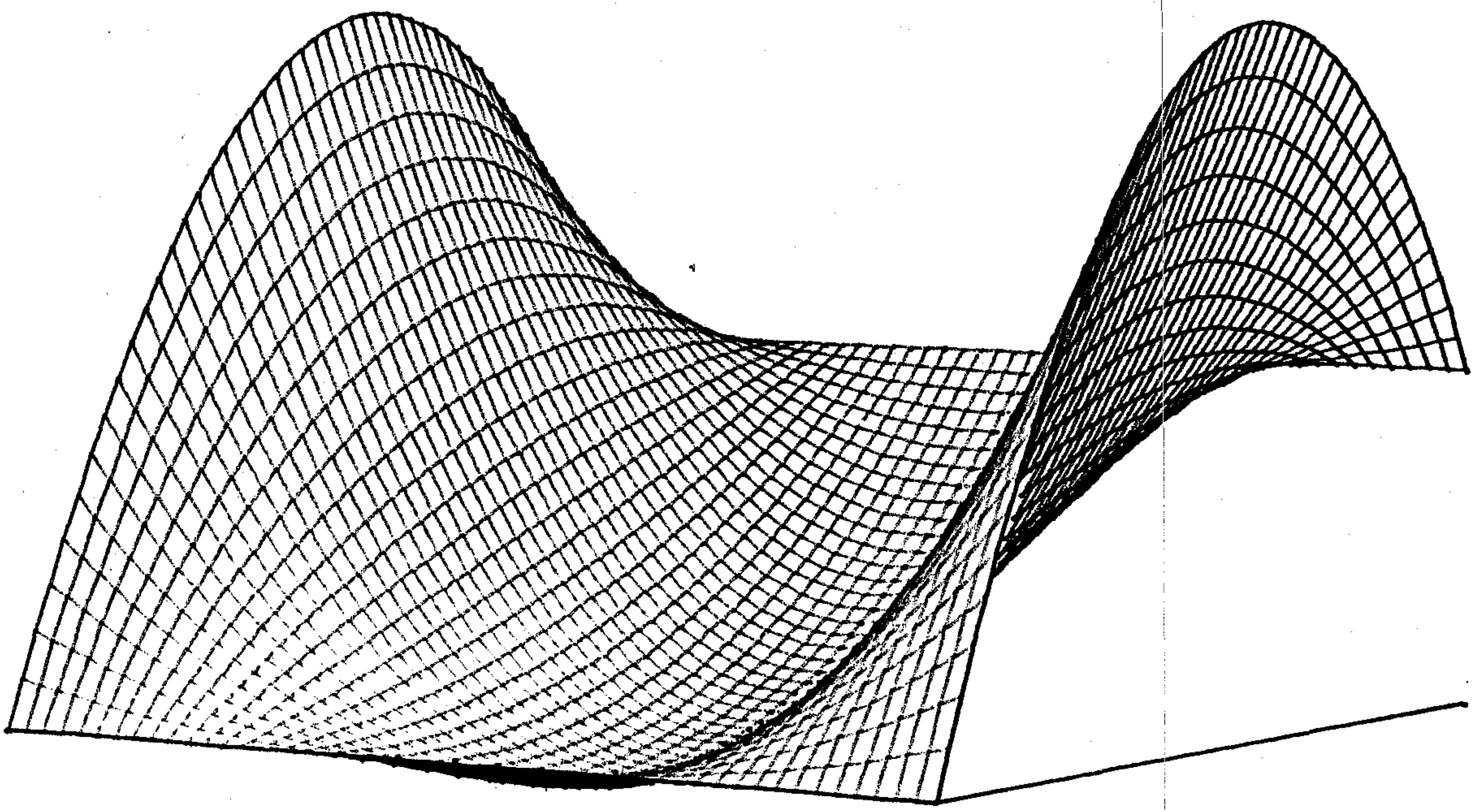


fig. 6-7a
Hydrogen atom (c.f. eqn 6b)

$w_2 = \frac{1}{2} p_2 (1-\xi_2) + \xi_2 - \xi_2$ for $-1 < \xi_2 < 1$

$-2 < p_2 < 2$

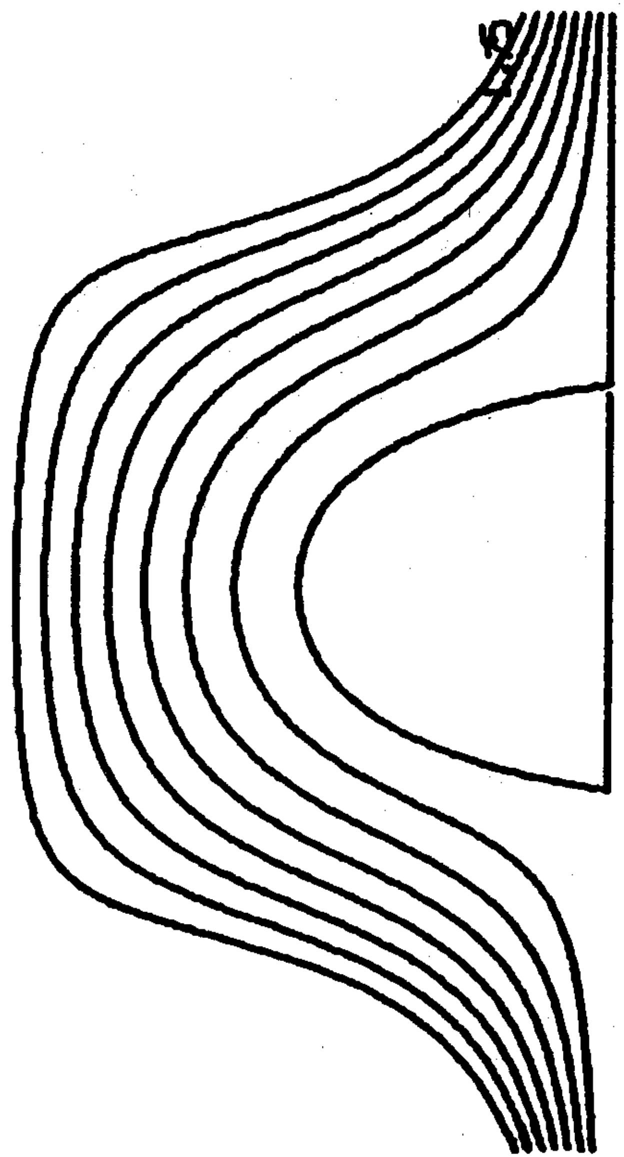
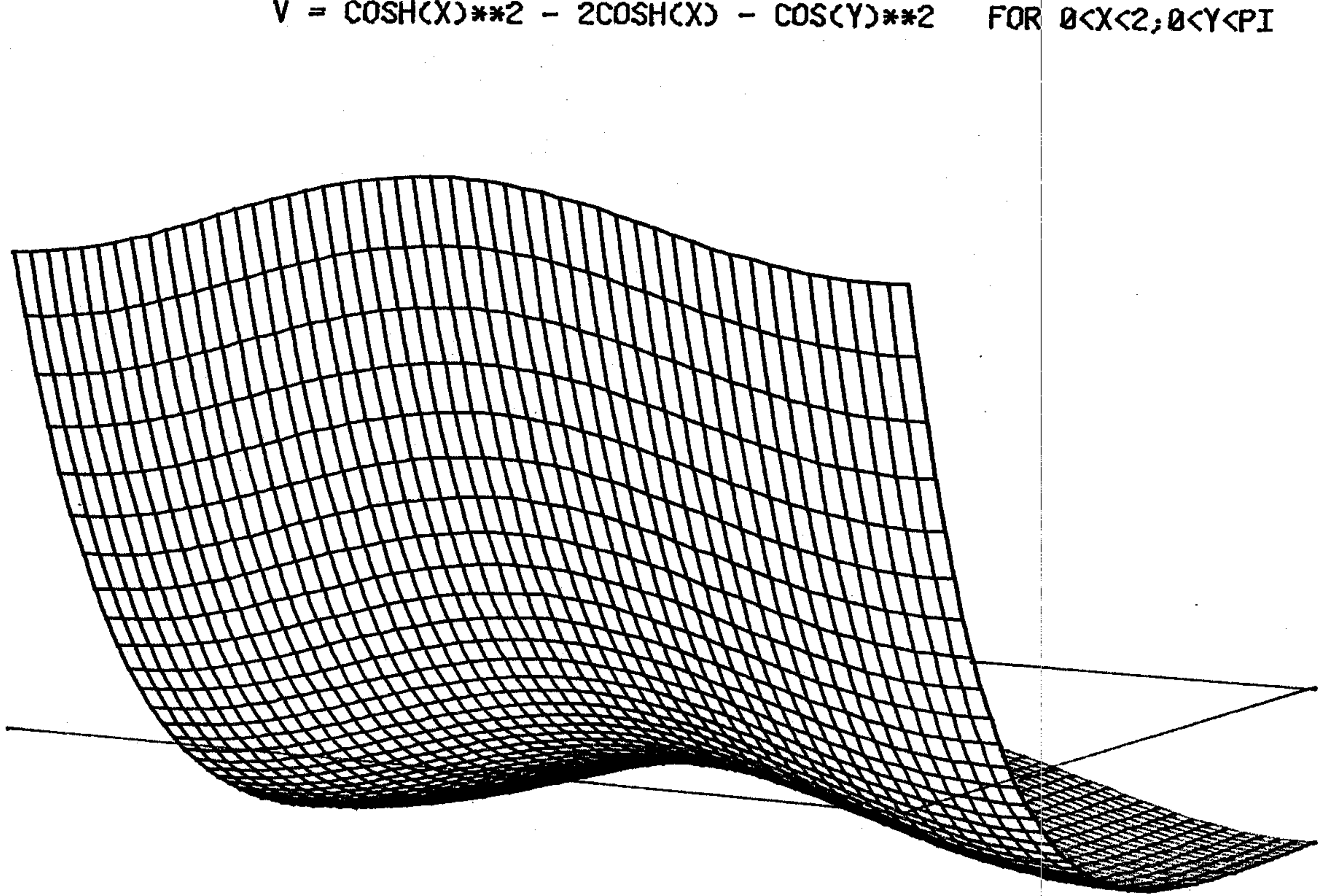


fig. 6-7b.



V^{eff} fig. 6-8a. (c.f. eqn 7a)

$0 < \theta < \pi$

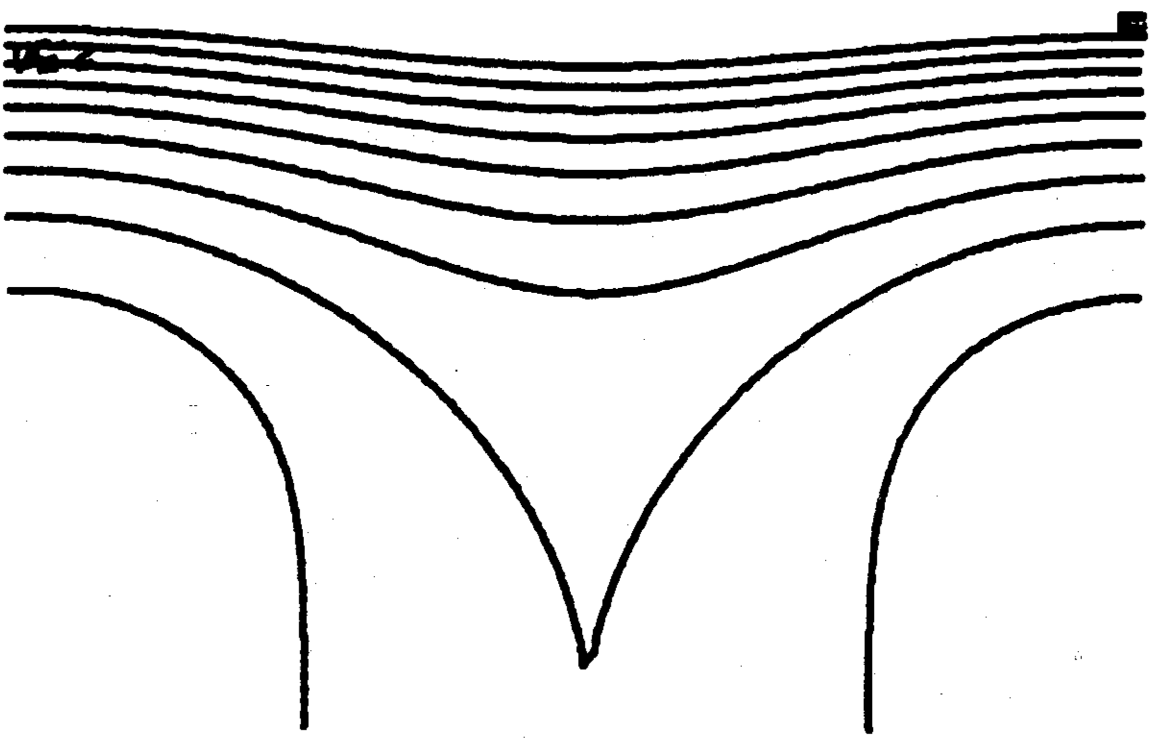


fig. 6-8b.

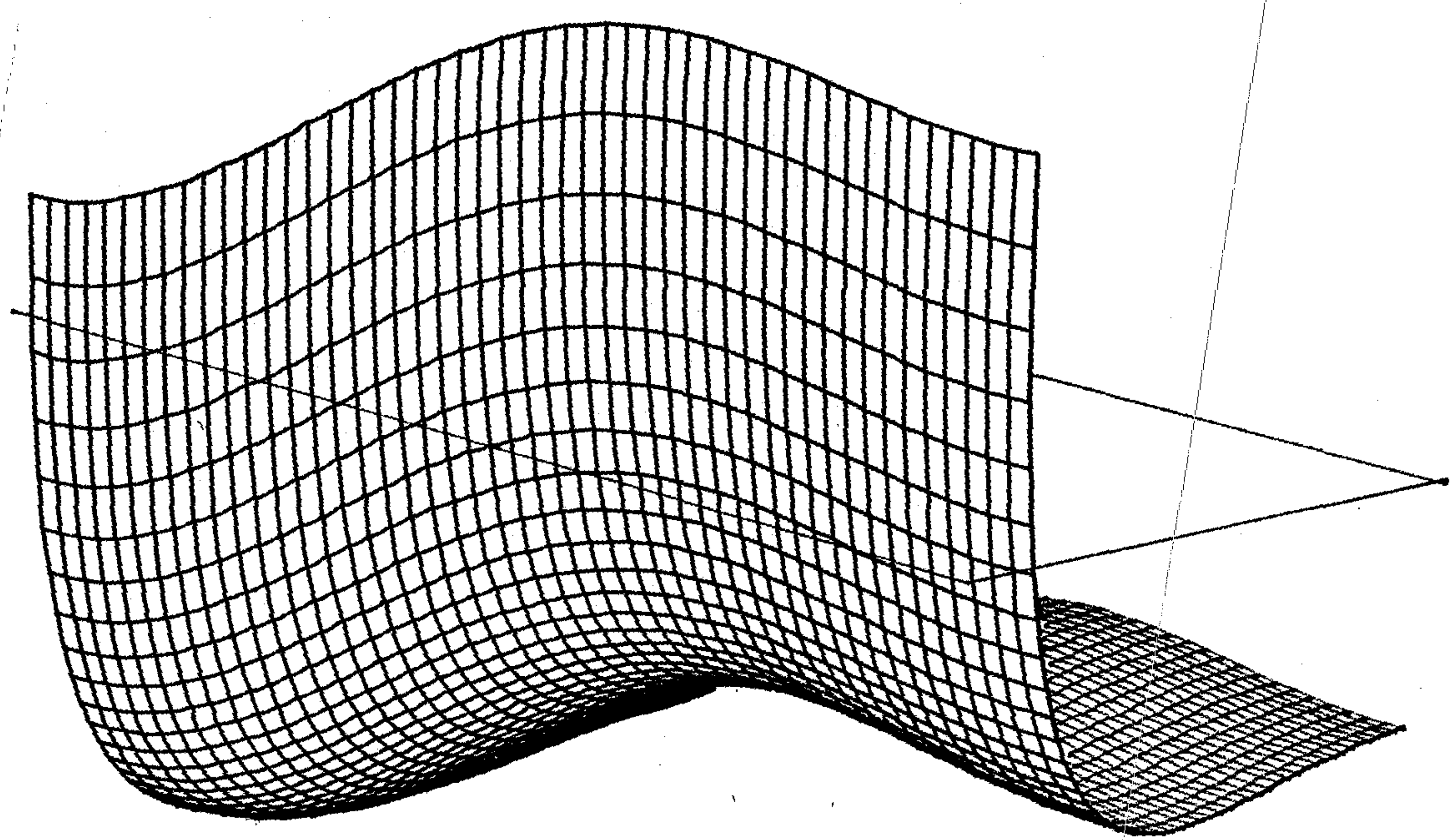


fig. 6-9a.
V^{eff} (c.f. eqn 7b)

$0 < \theta < \pi$

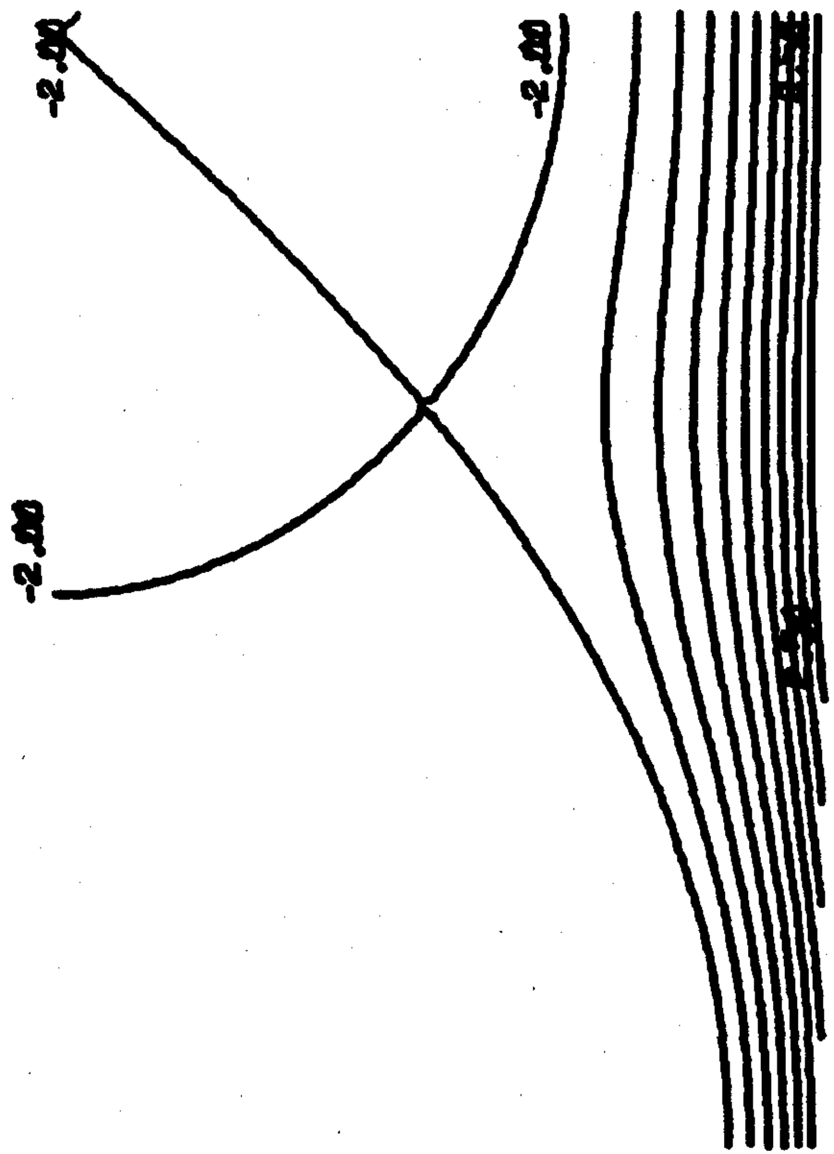


fig. 6-9b.

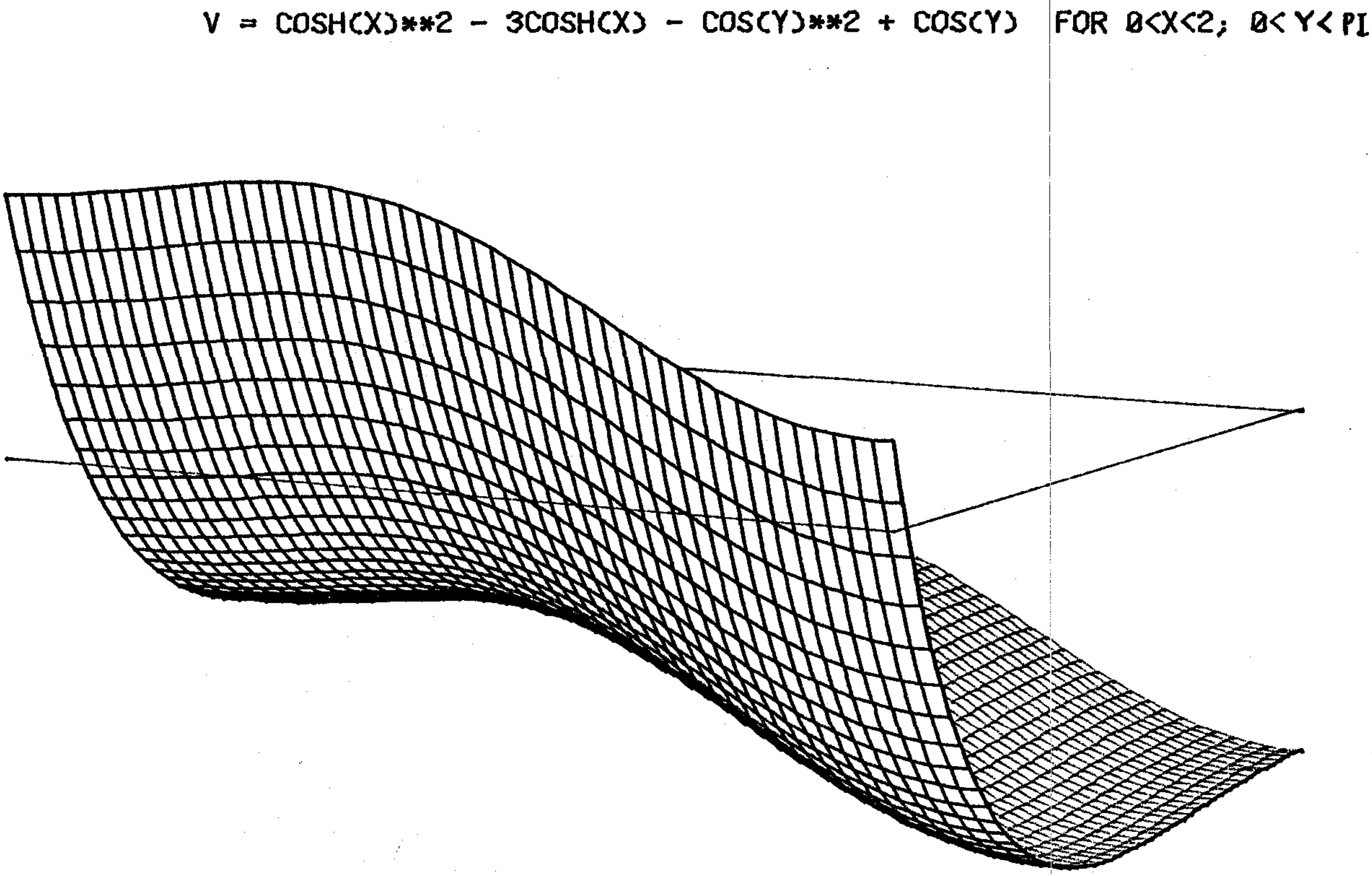


fig. 6-10a

V^{eff} (c.f. 7c)

for $0 < \mu < 2$

$0 < \theta < \pi$

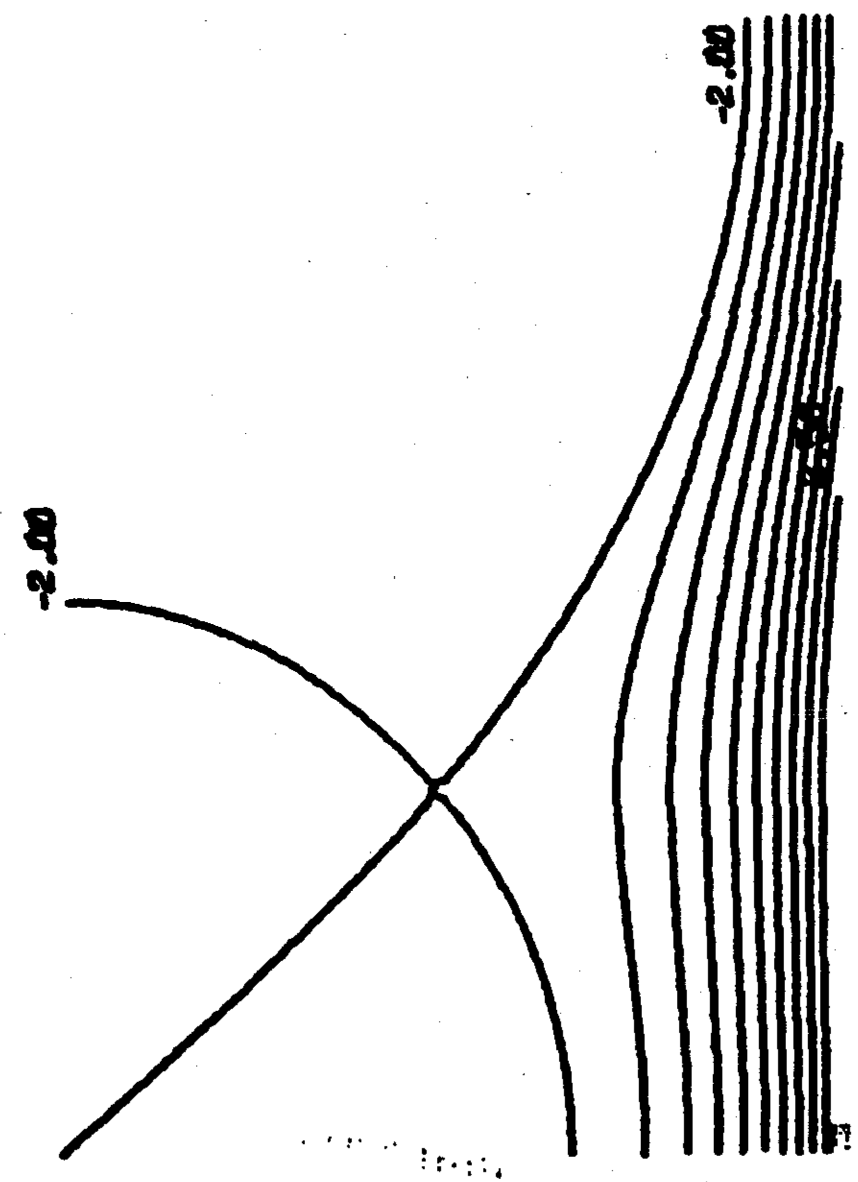


fig. 6-10b.

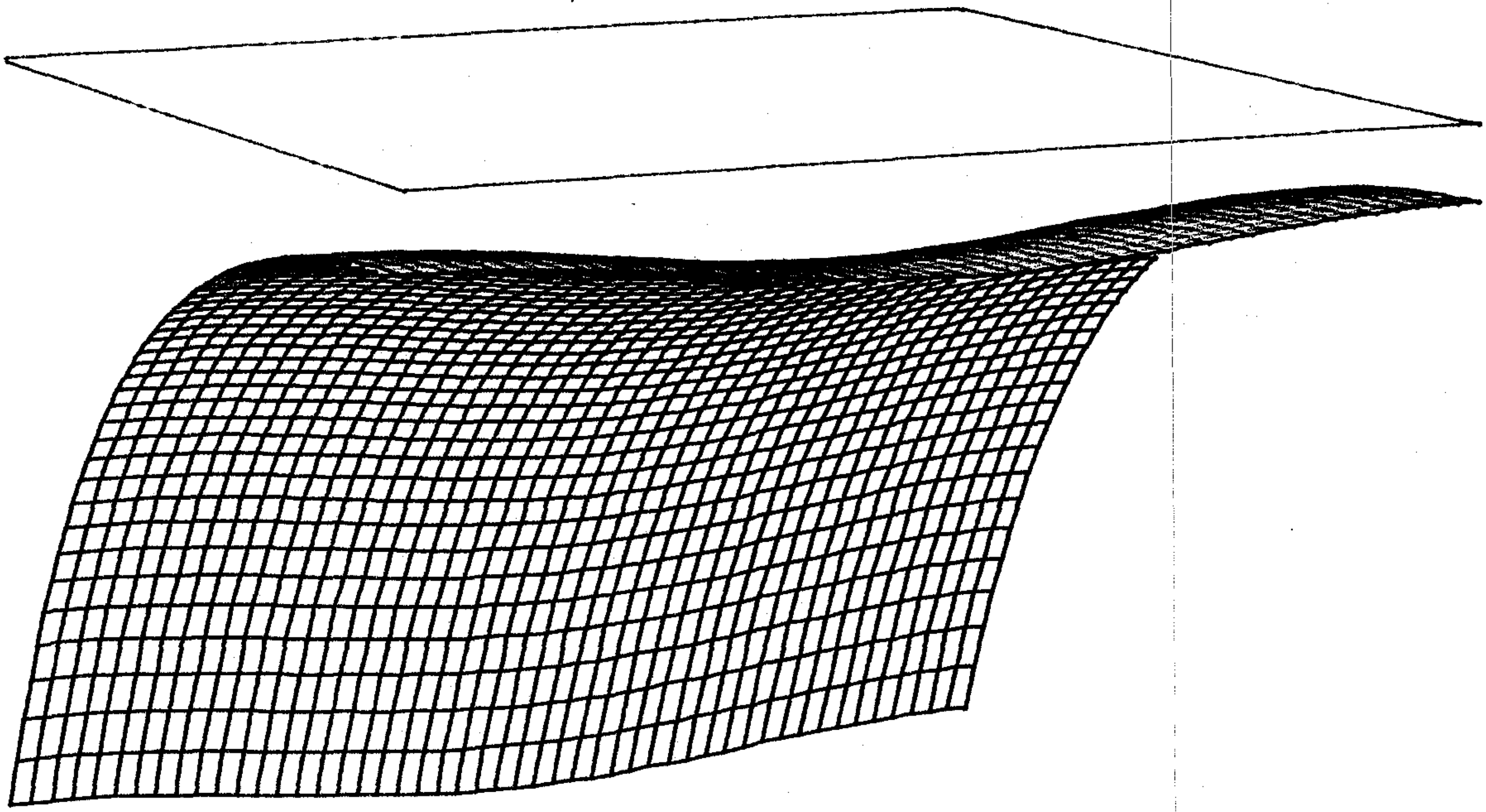


fig. 6-11a.

V^{eff} (c.f. 8a)

$0 < \mu < 2$

$0 < \theta < \pi$

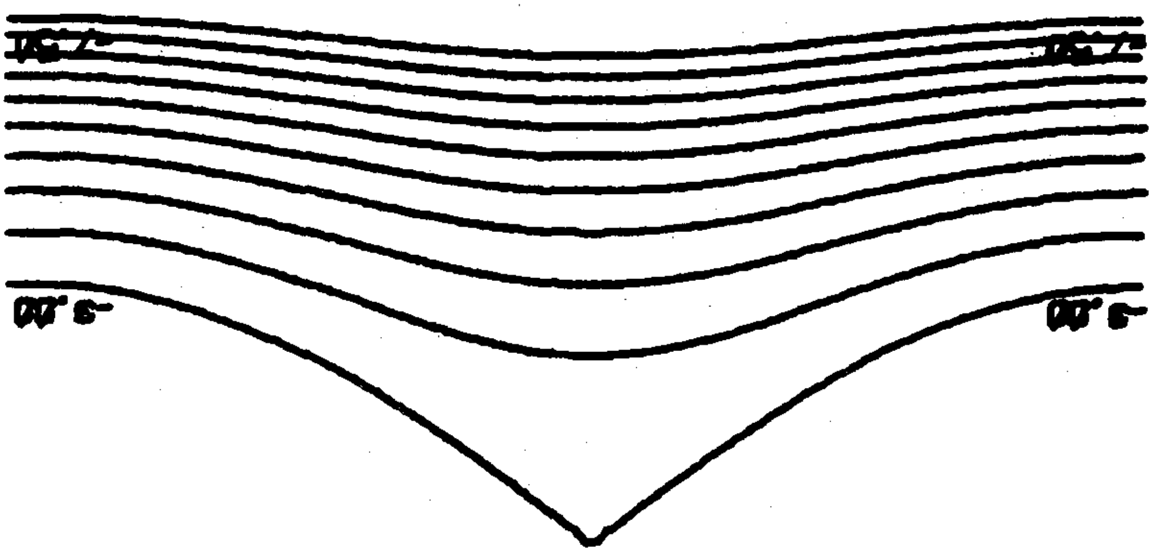


fig. 6-11b.

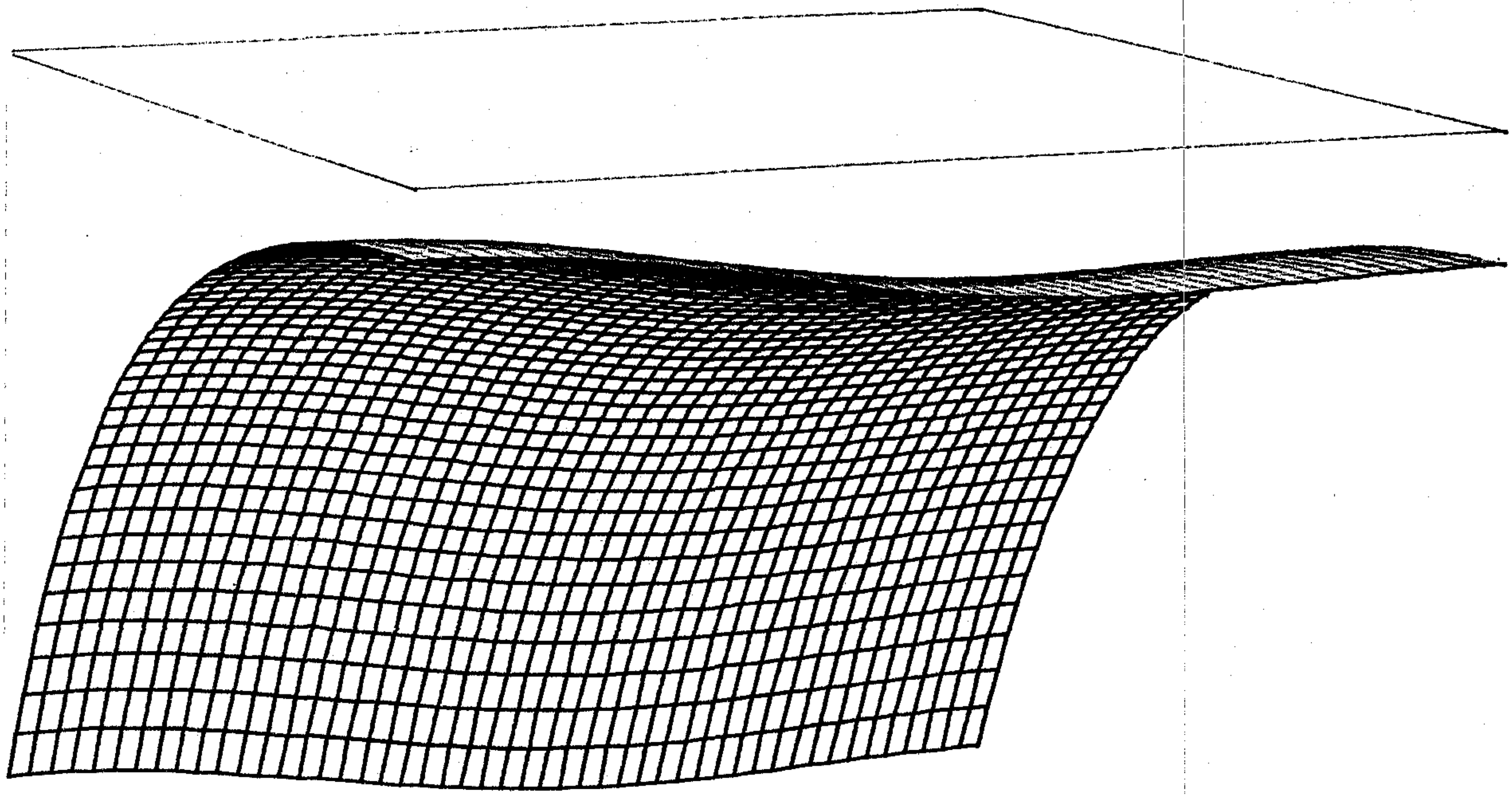


fig. 6-12a.

V^{eff} (c.f. eqn. 8b)

for $0 < \mu < 2$
$0 < \theta < \pi$

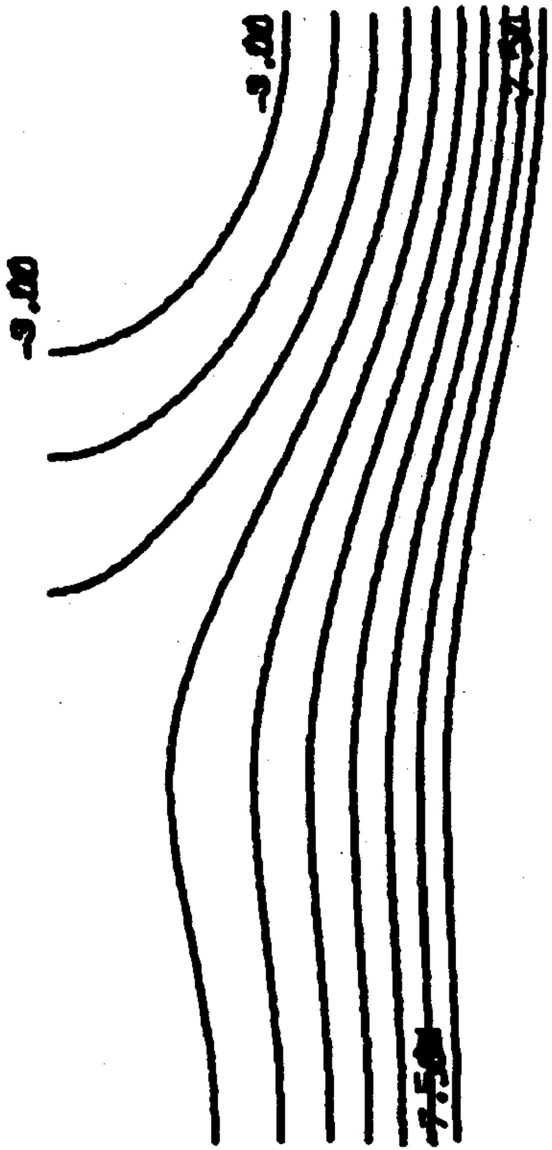


fig. 6-12b.

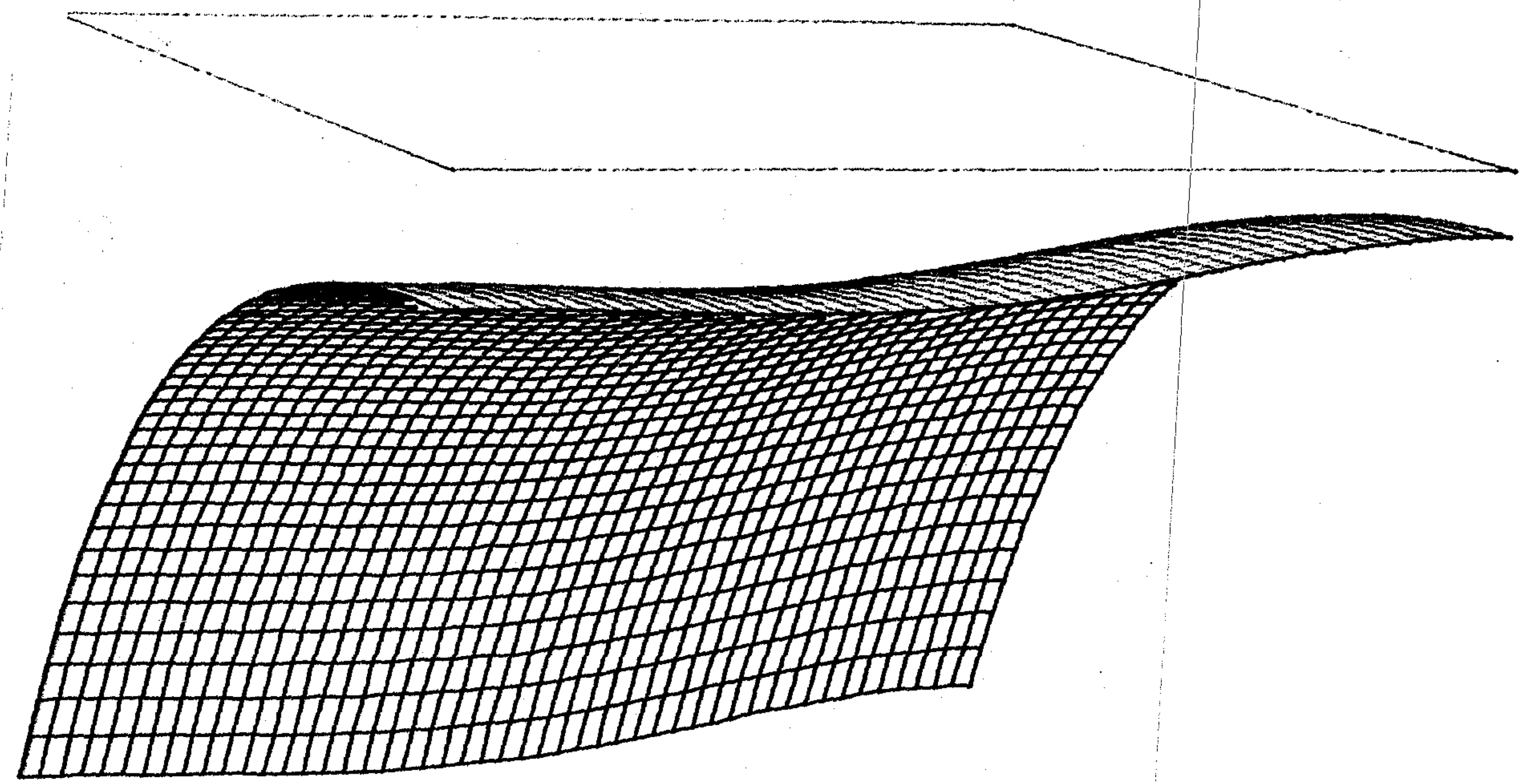


fig. 6-13a.

V^{eff} (c.f. eqn. 8c)

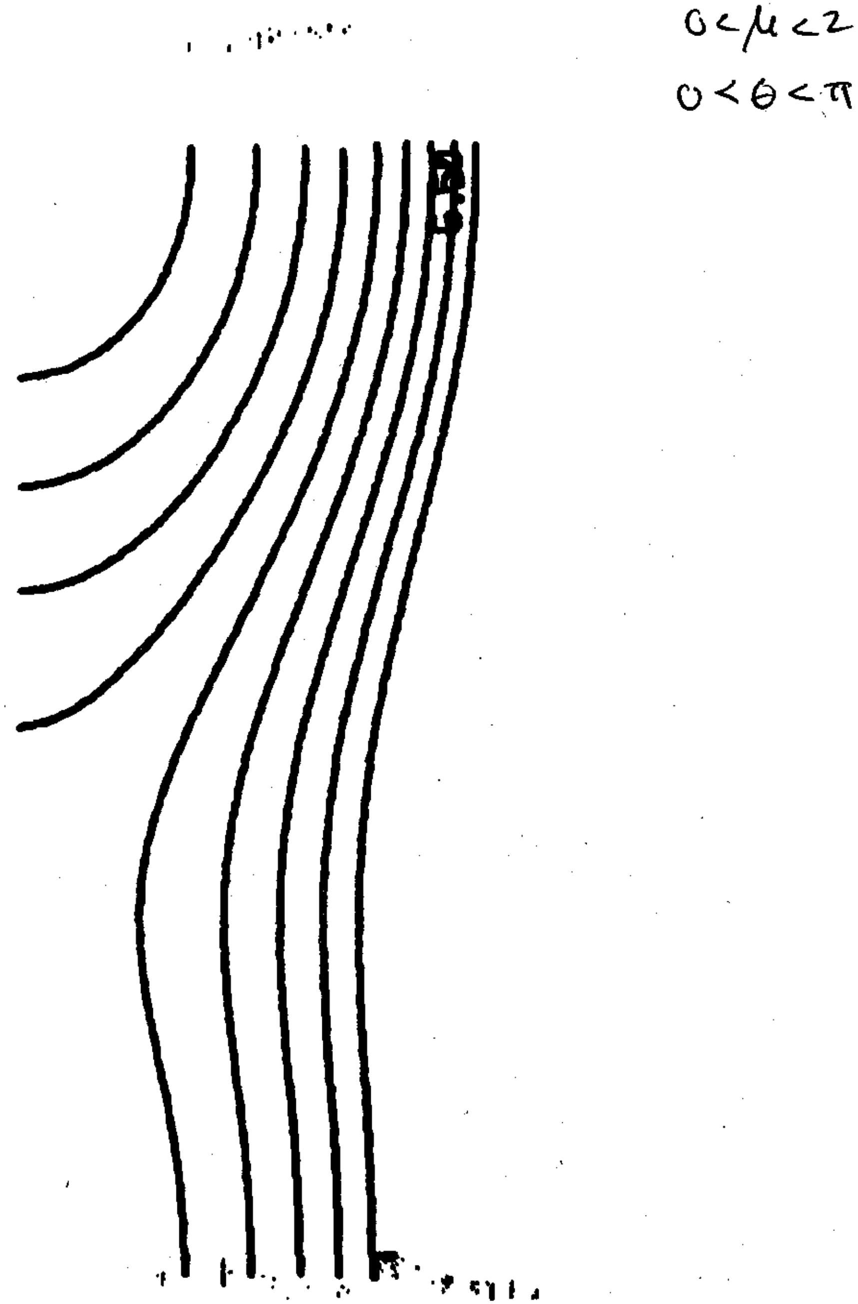


fig. 6-13b.

6-4. The Planar Two-Center Problem As a Two-Dimensional Anisotropic Oscillator

Any Hamiltonian function possessing a critical point may be expanded around that point and made to resemble a harmonic oscillator. Through a study of the topology of our Hamiltonian, we have found such a point at the origin ($\mu = \theta = 0$). We may obtain a second order approximation which resembles a harmonic oscillator through an expansion of the hyperbolic cosine and cosine functions

$$\begin{aligned} \cosh\mu &= 1 + \frac{1}{2!}\mu^2 + \frac{1}{4!}\mu^4 + \ldots + + \\ \cos\theta &= 1 - \frac{1}{2!}\theta^2 + \frac{1}{4!}\theta^4 + \ldots + + \\ \cosh^2\mu &= 1 + \mu^2 + \ldots + + \\ \cos^2\theta &= 1 - \theta^2 + \ldots + + . \end{aligned} \tag{1}$$

Then to second order, our Hamiltonian becomes

$$\begin{aligned} \bar{W}^{\text{approx}} &= \bar{W}_\mu^{\text{approx}} + \bar{W}_\theta^{\text{approx}} \\ \bar{W}_\mu^{\text{approx}} &= \frac{1}{2m} p_\mu^2 + \gamma\cosh^2\mu - c\alpha_1\cosh\mu \\ &= \frac{1}{2m} [p_\mu^2 + \kappa_\mu\mu^2] + \frac{1}{2}[\kappa_\mu - c\alpha_1] \\ \bar{W}_\theta^{\text{approx}} &= \frac{1}{2m} p_\theta^2 - \gamma\cos^2\theta + c\alpha_2\cos\theta \\ &= \frac{1}{2m} [p_\theta^2 + \kappa_\theta\theta^2] - \frac{1}{2} [\kappa_\theta - c\alpha_2] \end{aligned} \tag{2}$$

where $\kappa_\mu = 2(\gamma - \frac{c\alpha_1}{2})$ and $\kappa_\theta = 2(\gamma - \frac{c\alpha_2}{2})$ when $m = 1$. Thus $\bar{W}_\mu$ and $\bar{W}_\theta$ take the form of two independent harmonic oscillators with force constants κ_μ and κ_θ, respectively. Depending upon the relative values of the parameters (c, Z_1, Z_2, and E), these oscillators may be either repulsive or attractive. Their relations are given explicitly by

$$\kappa_\mu = c(Z_1Z_2 - 2Ec - Z_1 - Z_2) \quad (3a)$$

$$\kappa_\theta = c(Z_1Z_2 - 2Ec - Z_1 + Z_2) \quad (3b)$$

Thus, for a <u>repulsive</u> 2D oscillator, we have

$$0 > \kappa_\mu \qquad \underline{\text{and}} \qquad 0 > \kappa_\theta \;.$$

$$0 > c(Z_1Z_2 - 2Ec - Z_1 - Z_2) \qquad 0 > c(Z_1Z_2 - 2Ec - Z_1 + Z_2)$$

$$E > \frac{1}{2c}(Z_1Z_2 - Z_1 - Z_2) \qquad E > \frac{1}{2c}(Z_1Z_2 - Z_1 + Z_2) \quad (4)$$

<u>or</u>

$$E > \frac{Z_1}{2c}(Z_2 - 1)$$

and similarly, we have for an <u>attractive</u> oscillator

$$E < \frac{Z_1}{2c}(Z_2 - 1). \quad (5)$$

It is interesting to note that in the special cases $Z_2 = 1$ or $Z_1 = 0$, this approximation coincides nicely with the two-dimensional harmonic oscillator where energies less than zero give an attractive oscillator and energies greater than zero give a repulsive oscillator. Furthermore, we note the parameters c, Z_1 , and Z_2 are non-negative. This suggests that for $Z_2 \geq 1$, the oscillator will be attractive for a range of positive as well as negative energies.

The Hamiltonian can be written in action-angle variables by employing the following canonical transformation

$$p_\mu \rightarrow\rightarrow\rightarrow J_\mu = \frac{p_\mu^2}{\sqrt{\kappa_\mu}} + \sqrt{\kappa_\mu}\,\mu^2$$

$$p_\theta \rightarrow\rightarrow\rightarrow J_\theta = \frac{p_\theta^2}{\sqrt{\kappa_\theta}} + \sqrt{\kappa_\theta}\,\theta^2 \quad (6)$$

$$\mu \rightarrow\rightarrow\rightarrow \psi_\mu = \arctan \frac{p_\mu}{\mu\sqrt{\kappa_\mu}}$$

$$\theta \rightarrow\rightarrow\rightarrow \psi_\theta = \arctan \frac{p_\theta}{\theta\sqrt{\kappa_\theta}} \;.$$

Our Hamiltonian then becomes

$$\bar{W}^{approx} = \frac{1}{2}[\sqrt{\kappa_\mu}\, J_\mu + \sqrt{\kappa_\theta}\, J_\theta] + \frac{1}{2}[\kappa_\mu - c\alpha_1] - \frac{1}{2}[\kappa_\theta - c\alpha_2]$$
$$= \frac{1}{2}[\omega_\mu J_\mu + \omega_\theta J_\theta] - \varepsilon \qquad (7)$$

where $\omega_\mu = \sqrt{\kappa_\mu}$, $\omega_\theta = \sqrt{\kappa_\theta}$, $\varepsilon \equiv \varepsilon_\mu + \varepsilon_\theta = -\ [-\omega_\mu^2 + c\alpha_1 + \omega_\theta^2 - c\alpha_2]$ and ω corresponds to the frequency of the oscillator. Our approximate Hamiltonian thus admits the constants of the motions as described in chapter V

$$H = \omega_\mu J_\mu + \omega_\theta J_\theta$$
$$D = \omega_\mu J_\mu - \omega_\theta J_\theta \qquad (8)$$
$$K = \omega_\theta \psi_\mu - \omega_\mu \psi_\theta .$$

We mentioned earlier (at the end of section 6-1) that the work of Coulson and Joseph showed a relation between the Kepler two-center problem and the harmonic oscillator through the extra constant of the motion Ω. It has now been explicitly shown how such a connection can be made through this approximation.

6-5. A Second Order Approximation

Expanding the cosine and hyperbolic cosine functions in equations (1b) of section 6-3 to fourth order in μ and θ, we get for the next higher approximation to $\bar{W}$

$$\bar{W}^{approx} = \bar{W}_\mu^{approx} + \bar{W}_\theta^{approx}$$

$$\bar{W}_\mu^{approx} = \frac{1}{2}p_\mu^2 + \gamma(1 + \mu^2 + \frac{\mu^4}{3}) - c\alpha_1(1 + \frac{\mu^2}{2} + \frac{\mu^4}{24}) \qquad (10)$$

$$= \frac{1}{2}(p_\mu^2 + \omega_\mu^2\mu^2) + \frac{1}{2}(\omega_\mu^2 - c\alpha_1) + \frac{1}{6}\mu^4(\omega_\mu^2 + \frac{3}{4}c\alpha_1)$$

$$\bar{W}_\theta^{approx} = \frac{1}{2}p_\theta^2 - \gamma(1 - \theta^2 + \frac{\theta^4}{3}) + c\alpha_2(1 - \frac{\theta^2}{2} + \frac{\theta^4}{24})$$

$$= \frac{1}{2}(p_\theta^2 + \omega_\theta^2\theta^2) - \frac{1}{2}(\omega_\theta^2 - c\alpha_2) - \frac{1}{6}\theta^4(\omega_\theta^2 + \frac{3}{4}c\alpha_2).$$

Employing the canonical transformations of equation (6), where

$$\mu = J_\mu^{1/2}\cos\psi_\mu \qquad p_\mu = J_\mu^{1/2}\sin\psi_\mu$$

$$\theta = J_\theta^{1/2}\cos\psi_\theta \qquad p_\theta = J_\theta^{1/2}\sin\psi_\theta$$

we have to third order in the action-angle variables

$$\bar{W}_\mu^{approx} = \frac{1}{2}[J_\mu\sin^2\psi_\mu + \omega_\mu J_\mu\cos^2\psi_\mu] + \frac{1}{6}J_\mu^2\cos^4\psi_\mu[\omega_\mu^2 + \frac{3}{4}c\alpha_1] + \frac{1}{2}[\omega_\mu^2 - c\alpha_1]$$

$$= \frac{1}{2}[J_\mu(\psi_\mu - \frac{\psi_\mu^2}{2} + \ldots +)^2 + \omega_\mu J_\mu(1 - \frac{\psi_\mu^2}{2} + \ldots +)^2] + \frac{1}{6}J_\mu^2(1 - \frac{\psi_\mu^2}{2} + \ldots +)^4(\omega_\mu^2 + \frac{3}{4}c\alpha_1) - \varepsilon_\mu \qquad (12a)$$

$$= \frac{1}{2}\omega_\mu J_\mu + \frac{1}{6}(\omega_\mu^2 + \frac{3}{4}c\alpha_1)J_\mu^2 + \frac{1}{2}(1 - \omega_\mu)J_\mu\psi_\mu^2 - \varepsilon_\mu$$

$$\bar{W}_\theta^{approx} = \frac{1}{2}\omega_\theta J_\theta - \frac{1}{6}(\omega_\theta^2 + \frac{3}{4}c\alpha_2)J_\theta^2 + \frac{1}{2}(1 - \omega_\theta)J_\theta\psi_\theta^2 - \varepsilon_\theta \qquad (12b)$$

where $\varepsilon_\mu = -\frac{1}{2}[\omega_\mu^2 - c\alpha_1]$ and $\varepsilon_\theta = \frac{1}{2}[\omega_\theta^2 - c\alpha_2]$.

Let us consider for now, only those terms of up to second order in these variables. We then have

$$\bar{W}_\mu^{approx} = \frac{1}{2}\,\omega_\mu J_\mu + \frac{1}{6}(\omega_\mu + \frac{3}{4}\,c\alpha_1)J_\mu - \varepsilon_\mu$$
$$\bar{W}_\theta^{approx} = \frac{1}{2}\,\omega_\theta J_\theta - \frac{1}{6}(\omega_\theta + \frac{3}{4}\,c\alpha_2)J_\theta - \varepsilon_\theta \qquad (13)$$

or, reexpressing the α's in terms of the ω's and ε's, this becomes

$$\bar{W}_\mu^{approx} = \frac{1}{2}\,\omega_\mu J_\mu + \frac{1}{6}(\omega_\mu + \frac{3}{4}\,\omega_\mu^2 + \frac{3}{2}\,\varepsilon_\mu)J_\mu^2 - \varepsilon_\mu$$
$$\bar{W}_\theta^{approx} = \frac{1}{2}\,\omega_\theta J_\theta - \frac{1}{6}(\omega_\theta + \frac{3}{4}\,\omega_\theta^2 - \frac{3}{2}\,\varepsilon_\theta)J_\theta - \varepsilon_\theta \,. \qquad (14)$$

Defining

$$k_\mu = \frac{1}{6}(\omega_\mu + \frac{3}{4}\,\omega_\mu^2 + \frac{3}{2}\,\varepsilon_\mu)$$
$$k_\theta = -\frac{1}{6}(\omega_\theta + \frac{3}{4}\,\omega_\theta^2 - \frac{3}{2}\,\varepsilon_\theta) \qquad (15)$$

(14) becomes

$$\bar{W}_\mu^{approx} = \frac{1}{2}\,\omega_\mu J_\mu + k_\mu J_\mu^2 - [4k_\mu - \frac{2}{3}\,\omega_\mu(1 + \frac{3}{4}\,\omega_\mu)]$$
$$\bar{W}_\theta^{approx} = \frac{1}{2}\,\omega_\theta J_\theta + k_\theta J_\theta^2 - [4k_\theta + \frac{2}{3}\,\omega_\theta(1 + \frac{3}{4}\,\omega_\theta)] \,. \qquad (16)$$

Completing the squares on the J's and employing a canonical transformation, we obtain

$$\bar{W}_\mu^{approx} = k_\mu[J_\mu + \frac{1}{4}\,\frac{\omega_\mu}{k_\mu}]^2 - k_\mu(\frac{1}{4}\,\frac{\omega_\mu}{k_\mu})^2 - [4k_\mu - \frac{2}{3}\,\omega_\mu(1 + \frac{3}{4}\,\omega_\mu)]$$
$$= \nu_\mu \bar{J}_\mu - \bar{\varepsilon}_\mu$$
$$\bar{W}_\theta^{approx} = k_\theta[J_\theta + \frac{1}{4}\,\frac{\omega_\theta}{k_\theta}]^2 - k_\theta(\frac{1}{4}\,\frac{\omega_\theta}{k_\theta})^2 - [4k_\theta + \frac{2}{3}\,\omega_\theta(1 + \frac{3}{4}\,\omega_\theta)] \qquad (17)$$
$$= \nu_\theta \bar{J}_\theta - \bar{\varepsilon}_\theta$$

where

$$\bar{J}_\mu^{1/2} = J_\mu + \frac{1}{4}\frac{\omega_\mu}{k_\mu} \; ; \qquad \bar{J}_\mu^{1/2} = J_\mu + \frac{1}{4}\frac{\omega_\mu}{k_\mu}$$

$$\bar{\psi}_\mu = \frac{1}{2}\left(J_\mu + \frac{1}{4}\frac{\omega_\mu}{k_\mu}\right)^{-1}\psi_\mu \; ; \qquad \bar{\psi}_\theta = \frac{1}{2}\left(J_\theta + \frac{1}{4}\frac{\omega_\theta}{k_\theta}\right)^{-1}\psi_\theta$$

$$\nu_\mu = \sqrt{k_\mu} \; ; \qquad \nu_\theta = \sqrt{k_\theta} \tag{18}$$

$$\bar{\varepsilon}_\mu = k_\mu\left(\frac{1}{4}\frac{\omega_\mu}{k_\mu}\right)^2 + 4k_\mu - \frac{2}{3}\,\omega_\mu\left(1 + \frac{3}{4}\,\omega_\mu\right)$$

$$\bar{\varepsilon}_\theta = k_\theta\left(\frac{1}{4}\frac{\omega_\theta}{k_\theta}\right)^2 + 4k_\theta + \frac{2}{3}\,\omega_\theta\left(1 + \frac{3}{4}\,\omega_\theta\right)$$

and

$$\begin{aligned} \bar{W}^{\text{approx}} &= \bar{W}_\mu^{\text{approx}} + \bar{W}_\theta^{\text{approx}} \\ &= \nu_\mu \bar{J}_\mu + \nu_\theta \bar{J}_\theta - \bar{\varepsilon} \; ; \qquad \bar{\varepsilon} = \bar{\varepsilon}_\mu + \bar{\varepsilon}_\theta \end{aligned} \tag{19}$$

Thus, we again have an anisotropic harmonic oscillator where our constants of the motion are now

$$\begin{aligned} H' &= \nu_\mu \bar{J}_\mu + \nu_\theta \bar{J}_\theta \\ D' &= \nu_\mu \bar{J}_\mu - \nu_\theta \bar{J}_\theta \\ K' &= \nu_\theta \bar{\psi}_\mu - \nu_\mu \bar{\psi}_\theta . \end{aligned} \tag{20}$$

We see that this approximation sophisticates the generators, but does not alter the Lie algebra admitted by the system. As we shall see in the following section, there is a theorem by Birkoff which generalizes this.

6-6. Stability of the Algebras

In his studies of dynamical systems, George Birkoff [6] established the following theorem:

Let there be given a Hamiltonian system of equations

$$dq_i/dt = \partial H/\partial p_i \ , \qquad dp_i/dt = -\partial H/\partial q_i \ ; \quad i = 1,\ldots,n$$

i.e.,

$$dz_i/dt = \{H, z_i\} \tag{1}$$

which possesses a critical point z_0 :

$$\partial H/\partial p_i\big|_{z_0} = 0 = \partial H/\partial q_i\big|_{z_0} \ . \tag{2}$$

Then there exists a canonical transformation $q \to \bar{q}$, $p \to \bar{p}$ defineable by a formal power series

$$z_i = \Sigma\ \ell_{ij}\bar{z}_j + \tfrac{1}{2}\Sigma\ \ell_{ijk}z_j z_k + \ldots + + \tag{3}$$

in which $\det|\ell_{ij}| \neq 0$, and such that, in terms of the new coordinates, the equations (1) become

$$d\bar{q}_i/dt = c_i\bar{q}_i \ , \qquad d\bar{p}_i/dt = -c_i\bar{p}_i \ . \tag{4}$$

In terms of the new coordinates the Hamiltonian is

$$H = \Sigma\ c_i p_i q_i \ . \tag{5}$$

It follows immediately that

I. Every Hamiltonian system of n-degrees of freedom with a critical point admits the same abstract Lie algebras with generators $U = Q_i\partial/\partial q_i + P_i\partial/\partial p_i + C_i\partial/\partial c_i$.

2. Every such Hamiltonian system admits the same abstract Lie subalgebras in which the generators U leave the c_i invariant, that is the same algebras with generators in which the $C_i = 0$.

In our analysis, we expand H or W in power series about a critical point and write

$$\bar{W} \equiv W - W_0 = W_2 + W_3 + W_4 + \ldots + + = 0. \tag{6}$$

Here W_m is a polynomial of degree m. From the previous discussion, it is apparent that the Lie generators and commutation relations obtained when $\bar{W} = W_2$ can be expected to persist as successive terms W_m are allowed in the approximation to W. As the approximation to W becomes more sophisticated. the functional dependence of the coefficients C_i, Q_i, P_i in the generators U upon the original dynamical variables c_i, p_i, q_i will, in general, become more sophisticated.

This stability of the Lie algebraic properties of our system is of course extremely convenient from a computational standpoint. However, it must be recognized that a great deal of the physics of the system depends upon the particular functional form taken on by the C's, P's, and Q's.

Appendix I

Programs
DETERMININGEQNS, MFGR, COMMUTATOR, MFGRB, and DETERMININGEQNSB

Use of DETERMININGEQNS For Large Problems

We consider problems of matrix dimensions larger than 100x100 as large problems. It is approximately at this point that the program MFGRDETEQNS7 begins to take up approximately 30,000 words or more of core memory. As the regular user on the B6700 system here is allowed to run a top maximum job of 30,000 words, it is wise to usually run at 20,000 or less words for any problem taking more than 5 minutes of processing time (≈ 15 minutes of execution time).

In such cases that the user must run large problems, the user should use program DETEQNS7B and MFGRDETEQNS7B. This alternate version of DETERMININGEQNS uses packed arrays; thus the data structure used is somewhat different than in the original version. For example, on page 2 of section 4-3 we note that the first field of a "term" (the coefficient field) is a real number. In order to pack the matrix of coefficients (MAT) which we must eventually diagonalize, these coefficients must all be of integer data types. This presents the first problem then, as the program must be able to somehow determine what part of the integral number represents the decimal portion. We circumvent this problem by introducing the parameter SIGDIG (line #10350 of DETEQNS7B and line #10500 of MFGRDETEQNS7B) which will set the number of significant digits to the right of the decimal. Thus, this parameter must be adjusted to the user's specification for various degrees of accuracy before he runs the program. Furthermore, in order to make the program more efficient, another parameter (NODIGITS) was introduced which specifies the largest integer value that the coefficients will have.

Note: The integral number represents both the decimal and whole fraction of the coefficient. For example, let us assume that the largest coefficient to be set in the matrix is 100.5; then the value to be set for NODIGITS would be 1005, and SIGDIG must be set at 1 (i.e. NODIGITS = 1005; SIGDIG = 1).

However, we must now consider the sign of the coefficient (±). Since the data type we must use does not include the sign, we must reserve another digit of the integral number to represent the sign. If the coefficient is positive, we set the last of the integer to 1; if it is negative, we set it to 0. Thus, we must now allow for an extra digit in NODIGITS.

e.g. Suppose the largest magnitude of the coefficients in the matrix is ±100.5. Then set NODIGITS = 10051 and SIGDIG = 1.

One other set of parameters may be adjusted to save core space. These parameters are MROWS and MCOLS (line 10300 in MFGRDETEQNS7 and MFGRDETEQNS7B). These parameters simply reserve the neccessary space for the coefficient matrix. They should not be set any higher than neccessary as specified by the output of DETEQNS7 and DETEQNS7B.

Program DETERMININGEQNS

```
#FILE (A015006)DETEQNS7 ON SWAT
      (*$U-*)
       (*$ N 064 *)
       (*$ SPACE 30000*)
PROGRAM DETERMININGEQNS (INPUT,OUTPUT);
    CONST   MAXDIM   =   8;
            NOTERMS  =1023;   (*  NOTERMS = (MAXPWR+1)**MAXDIM  *)
            EPS = 1.0E-7;
  TYPE      POINTER  =  @TERM;
            TERM  =  RECORD
                       COEFF    : REAL;
                       A        : REAL;
                       EXP      : INTEGER;
                       PTR      : POINTER;
                     END;
            DATA1FILE = FILE OF INTEGER;
            DATA2FILE = FILE OF REAL;
            DATA3FILE = FILE OF INTEGER;
            MATRIXFILE = FILE OF REAL;
 VAR        C         : REAL;
            D1        : DATA1FILE;
            D2        : DATA2FILE;
            D3        : DATA3FILE;
            MAT       : MATRIXFILE;
            REPLY    : CHAR;
            CH       : CHAR;
            EQN: ARRAY [1..NOTERMS] OF POINTER;
            THEAD : ARRAY [0..MAXDIM] OF POINTER;
            REQ       : ARRAY [1..100] OF POINTER;
            FHEAD : ARRAY [1..MAXDIM] OF POINTER;
            TDHEAD,FDHEAD: ARRAY [1..MAXDIM,1..MAXDIM] OF POINTER;
            MATRIX,COL,ROW :  ARRAY[1..NOTERMS] OF REAL;
            T,T1,I,K,COUNT,E,L,MAXPWR,TTLMAX,TENDIM,NOEQNS,
            DIM,NW,J,NOCOL,IRANK,TTERMS:  INTEGER;
            MAXUW:  ARRAY[1..10] OF INTEGER;
            IROW,ICOL:  ARRAY[1..NOTERMS] OF INTEGER;
            Z1,Z2,Z,PHEAD :  POINTER;
            PEQNS,MANIFOLD,TOTALMXPWR   :  BOOLEAN;
            COND:  ARRAY [1..10] OF BOOLEAN;
            UW,W,W1 : ARRAY[1..10] OF POINTER;
            DW : ARRAY [1..10,1..MAXDIM] OF POINTER;
            WTERMS : ARRAY [1..10] OF INTEGER;
PROCEDURE PRINT1 (HEAD:POINTER;J:INTEGER);
  VAR P       : POINTER;
      I,K     : INTEGER;
BEGIN
P := HEAD;
WRITELN('EQN[',J:DIM,'] IS:');
WRITELN('     COEFF          A        EXP');
```

```
WHILE P<> NIL DO
  BEGIN
  I := TRUNC (P@.A);
  K := TRUNC ((P@.A - I) * TENDIM);
  WRITELN (P@.COEFF,P@.A,P@.EXP);
  P := P@.PTR;
  END;
END;  (*  OF PRINT1  *)
PROCEDURE REMOVE (VAR HEAD : POINTER);
  VAR       P,P1  : POINTER;
BEGIN
P1 := HEAD;
P := HEAD;
WHILE P <> NIL DO
  BEGIN
  P := P@.PTR;
  DISPOSE(P1);
  P1 := P;
  END;
HEAD := NIL;
END;    (*  OF REMOVE  *)
PROCEDURE NEGATE (VAR  HEAD : POINTER);
  VAR   P : POINTER;
BEGIN
P := HEAD;
WHILE P<>NIL DO
  BEGIN
  P@.COEFF := -P@.COEFF;
  P := P@.PTR;
  END;
END;   (*  OF NEGATE  *)
PROCEDURE ADD(HEAD1,HEAD2:POINTER);
  VAR    P:  POINTER;
BEGIN
P := HEAD1;
WHILE P@.PTR <> NIL DO P := P@.PTR;
P@.PTR := HEAD2;
END;  (*  OF ADD  *)
PROCEDURE CREATETHETA (VAR  THEAD : POINTER;  I : INTEGER);
   VAR  A     : INTEGER;
        P,P1  : POINTER;
BEGIN
 P1 := NIL; P:= NIL;
RESET (D1);
WHILE NOT EOF(D1) DO
  BEGIN
  A := D1@; GET (D1);
  NEW(P);
  P@.EXP := A;
  P@.A := A/TENDIM + I;
```

```
  P@.PTR := NIL;
  IF P1<>NIL THEN P1@.PTR := P
             ELSE THEAD := P;
  P1 := P;
  END;
END;     (*  OF CREATETHETA  *)
PROCEDURE CREATEF (VAR  FHEAD : POINTER  ;  I : INTEGER);
  VAR      C        : REAL;
           P,P1     :  POINTER;
           COUNT,E : INTEGER;
BEGIN
P := NIL; P1 := NIL;
RESET (D2); RESET (D3);
COUNT := 1; C := 0.0; E := 0;
WHILE COUNT < I DO
  BEGIN
  WHILE (D2@<5E10) AND (D3@>=0) DO
    BEGIN
    GET (D2);
    GET (D3);
    END;
  GET (D2); GET (D3);
  COUNT := COUNT + 1;
  END;
WHILE (D2@<5E10) AND (D3@>=0) DO
  BEGIN
  NEW(P);
  C := D2@; GET (D2);
  P@.COEFF := C;
  E := D3@; GET (D3);
  P@.EXP := E;
  P@.PTR := NIL;
  IF FHEAD = NIL THEN FHEAD := P
                 ELSE P1@.PTR := P;
  P1 := P;
  END;
END;  (*   OF CREATEF   *)
PROCEDURE DERIV (HEAD :POINTER; VAR  DHEAD : POINTER;  K :\
  \ INTEGER);
  LABEL  100;
  VAR    P,Q,Q1           : POINTER;
         H,I                 : INTEGER;
BEGIN
P := HEAD; Q := NIL; Q1 := NIL;
WHILE P<>NIL DO
  BEGIN
  NEW(Q);
  WITH Q@ DO
    BEGIN
```

```
      H := H MOD 10;
      COEFF := P@.COEFF*H;
      IF COEFF = 0.0 THEN
        BEGIN
         DISPOSE (Q) ;
        GOTO 100;
        END;
      H := 1;
      FOR I:=1 TO (DIM-K) DO H := H*10;
      EXP := P@.EXP - H;
      A := P@.A;
      PTR := NIL;
      END;
    IF DHEAD = NIL THEN DHEAD := Q
                   ELSE Q1@.PTR := Q;
    Q1 :=Q;
100:  P:=P@.PTR;
    END;
END;   (*  OF DERIV  *)
PROCEDURE MULTPOLYS(VAR HEADF,HEADT:POINTER;VAR PHEAD:POINTER;MAX
                     :INTEGER);
  LABEL  110;
  VAR    P,Q,R,R1,P1      : POINTER;
         H,H1,H2,SUM,G1,G2,I,J,K  : INTEGER;
BEGIN
P := HEADF;P1:=HEADF;
R := PHEAD;
Q := HEADT;
IF R<>NIL THEN WHILE R@.PTR <>NIL DO R:= R@.PTR;
R1 := R;
WHILE (P<>NIL) DO
  BEGIN
  Q := HEADT;
  WHILE Q<>NIL DO
    BEGIN
    H := 0; SUM := 0; G1 := P@.EXP; G2 := Q@.EXP;
    FOR I := 1 TO DIM DO
      BEGIN
      H1 := G1 MOD 10; G1 := G1 DIV 10;
      H2 := G2 MOD 10; G2 := G2 DIV 10;
      H1 := H1 + H2;
      SUM := SUM + H1;
      IF SUM > MAX THEN GOTO 110;
      J := 1;
      FOR K := 1 TO (I-1) DO J := J*10;
      H := H + (H1 * J);
      END;
    NEW(R);
    WITH R@ DO
      BEGIN
      A := Q@.A;
```

```
        COEFF := P@.COEFF * Q@.COEFF;
        EXP := H;
        PTR := NIL;
        END;
      IF R@.COEFF = 0 THEN
        BEGIN
        DISPOSE (R);
        GOTO 110;
        END;
      IF PHEAD = NIL THEN PHEAD := R
                     ELSE R1@.PTR := R;
      R1 := R;
      110:  Q := Q@.PTR;
      END;
    P1:=P;
    P := P@.PTR;
    END;
END;     (*  OF MULTPOLYS  *)
PROCEDURE SORT (VAR HEAD:POINTER;KIND:BOOLEAN);
  VAR   R,F           : ARRAY [0..9] OF POINTER;
        P,P1,PREV     : POINTER;
        I,J,K,L,L1 : INTEGER; (*  KIND=TRUE IF SORTING ON X'S  *)
                              (*  KIND=FALSE IF SORTING ON A'S  \
   \*)
BEGIN
FOR J := 1 TO DIM DO
  BEGIN
  FOR I := 0 TO 9 DO   (*  INITIALIZE POCKETS  *)
    BEGIN
    F[I] := NIL; R[I] := NIL;
    END;
  P := HEAD;
  WHILE P<>NIL DO
    BEGIN
    L := 1;
    FOR L1 := 1 TO J-1 DO L := L * 10;
    IF KIND THEN K := (P@.EXP DIV (L)) MOD 10
            ELSE
              BEGIN
              K := (TRUNC (P@.A * TENDIM*10) DIV (L)) MOD 10;
              END;
    P1 := P@.PTR;
    IF R[K] = NIL
      THEN
        BEGIN
        R[K] := P;
        F[K] := P;
        END
      ELSE
        BEGIN
        R[K]@.PTR := P;
```

```
    R[K] := P;
    END;
  P@.PTR := NIL;
  P := P1;
  END;
 K := 0;  (*  CONCATENATE THE POCKETS  *)
        (*  FIND FIRST NON-EMPTY POCKET  *)
 WHILE( F[K] = NIL) AND (K<9) DO K:=K+1;
 HEAD := F[K];
 IF K<9
  THEN
   FOR I := (K+1) TO 9 DO
    BEGIN
    PREV := R[I-1];
    IF R[I] <> NIL
     THEN PREV@.PTR := F[I]
     ELSE R[I] := PREV;
    END;
 END;  (*  OF FOR J  *)
END;         (*    OF    SORT     *)
PROCEDURE PRINT (HEAD : POINTER);
 LABEL   5;
 VAR    P          : POINTER;
        I,J        : INTEGER;
BEGIN
P := HEAD;
5:  WHILE P<>NIL DO
 BEGIN
 I := TRUNC(P@.A);
 J := TRUNC ((P@.A-I) * TENDIM);
 WRITE ('+(', P@.COEFF:6:2,'A[',I:2,',',J:DIM,'])');
 P := P@.PTR;
 END;
WRITELN (' * (',HEAD@.EXP:2, ')    = 0');
END;        (*  OF PRINT  *)
PROCEDURE STORETHETA (DIM,MAXPWR: INTEGER; VAR D1:DATA1FILE);
 LABEL  10;
 VAR    COUNT,EXPONENT,DIGIT,POWER,SUM,I,N,J,H,E: INTEGER;
BEGIN
REWRITE(D1);
COUNT := 0; EXPONENT := 0;
D1@ := EXPONENT; PUT (D1);
10: WHILE COUNT<DIM DO
 BEGIN
 EXPONENT := EXPONENT + 1;
 N := 1;
 FOR I := 1 TO COUNT DO N := N*10;
 DIGIT := (EXPONENT DIV N) MOD 10;
 IF DIGIT <= MAXPWR
  THEN
   BEGIN
   IF NOT TOTALMXPWR
    THEN
     BEGIN
```

```
        IF NOT TOTALMXPWR
          THEN
            BEGIN
            POWER := EXPONENT;
            FOR J := 1 TO DIM DO
              BEGIN
              H := POWER MOD 10;
              POWER := POWER DIV 10;
              IF H > MAXPWR THEN GOTO 10;(*EXIT*)
              END;
            D1@ := EXPONENT;
            PUT(D1);
            END
          ELSE
            BEGIN
            POWER := EXPONENT;
            SUM := 0;
            FOR J:=1 TO DIM DO
              BEGIN
              H := POWER MOD 10;  POWER := POWER DIV 10;
              SUM := SUM  + H;
              END;
            IF SUM <= MAXPWR
              THEN
                BEGIN
                D1@ := EXPONENT;
                PUT (D1);
                END;
            END;
        END
      ELSE
        BEGIN
        COUNT := COUNT+1;
        EXPONENT := (10*N) - 1;
        END;
    END;   (*  OF WHILE  *)
  END;  (*  OF STORETHETA  *)
  PROCEDURE SEPERATETERMS(POLYNOMIAL: POINTER);
     LABEL    25;
     VAR      Z,Z1: POINTER;
    BEGIN
    NOEQNS := 0;
    WHILE POLYNOMIAL <> NIL DO
      BEGIN
      NOEQNS := NOEQNS+1; T := T+1;
      EQN[T] := POLYNOMIAL;
      ROW[T] := EQN[T]@.EXP;
      IF EQN[T]@.PTR <> NIL
        THEN
          BEGIN
          Z := EQN[T]@.PTR; Z1 := EQN[T];
```

```
  REMOVE(EQN[I]);
  END;
END;(* OF CREATEMATRIX *)

PROCEDURE SUBSTITUTE (VAR HEAD1:  POINTER; HEAD2:  POINTER);
  LABEL     70,100;
  VAR       P,P1,Z,HEAD3,Q,Q1,P2,P3:  POINTER;
BEGIN
IF HEAD2 = NIL THEN
  BEGIN
  WRITELN('MANIFOLD INVALID');
  GOTO 100;  (*EXIT*);
  END;
IF HEAD1 = NIL THEN
  BEGIN
  WRITELN('EMPTY POLYNOMIAL');
  GOTO 100; (*EXIT*)
  END;
P1 := HEAD1;
WHILE P1@.EXP <> HEAD2@.EXP DO
  BEGIN
  P2 := P1;
  P1 := P1@.PTR;
  IF P1 = NIL THEN GOTO 100;  (*EXIT*)
  END;
Z := HEAD2@.PTR;
HEAD3 := NIL;
WHILE Z <> NIL DO
  BEGIN
  P := P1;
  WHILE P@.EXP = HEAD2@.EXP DO
    BEGIN
    NEW (Q);
    WITH Q@ DO
      BEGIN
      COEFF := -P@.COEFF * Z@.COEFF/HEAD2@.COEFF;
      A := P@.A;
      EXP := Z@.EXP;
      PTR := NIL;
      END;
    IF HEAD3 = NIL THEN HEAD3 := Q
                   ELSE Q1@.PTR := Q;
    Q1 := Q;
    P3 := P;
    P := P@.PTR;
    IF P = NIL THEN GOTO 70;
    END;
  70:  Z := Z@.PTR;
  END;
P2@.PTR := HEAD3;
Q@.PTR := P3@.PTR;
```

```
          WHILE EQN[T]@.EXP = Z@.EXP DO
            IF Z@.PTR <> NIL
              THEN BEGIN
                   Z1 := Z;
                   Z := Z@.PTR;
                   END
              ELSE BEGIN
                   POLYNOMIAL := NIL;
                   GOTO 25;
                   END;
          Z1@.PTR := NIL;
          POLYNOMIAL := Z;
          25:  END
        ELSE POLYNOMIAL := NIL;
        IF PEQNS THEN PRINT1(EQN[T],T);
      END;
    END; (* OF SEPERATETERMS *)
PROCEDURE CREATEMATRIX (NOROWS:INTEGER);
    VAR   I,L:  INTEGER;
          Z,Z1,POLYNOMIAL:  POINTER;
BEGIN
FOR I := (T-NOROWS+1) TO T DO   (*  SORT AND PRINT EQN[I]  *)
    BEGIN
    POLYNOMIAL := EQN[I];  (*  NOTE: CHANGE MEANING OF POLYNOMIAL  \
     \*)
    FOR L := 1 TO NOCOL DO MATRIX[L] := 0.0;
    WHILE POLYNOMIAL <> NIL DO
      BEGIN
      Z := POLYNOMIAL@.PTR; Z1 := POLYNOMIAL;
      WHILE Z <> NIL DO
        BEGIN
        IF Z@.A = POLYNOMIAL@.A
          THEN
            BEGIN
            POLYNOMIAL@.COEFF := POLYNOMIAL@.COEFF + Z@.COEFF;
            Z1@.PTR := Z@.PTR;
            DISPOSE (Z) ;
            END
          ELSE Z1 := Z;
        Z := Z1@.PTR;
        END;
      L := 1;
      WHILE (POLYNOMIAL@.A <> COL[L]) DO L:=L+1;
      MATRIX [L] := POLYNOMIAL@.COEFF;
      POLYNOMIAL := POLYNOMIAL@.PTR;
      END;
    FOR L := 1 TO NOCOL DO
      BEGIN
      MAT@ := MATRIX[L];
      PUT(MAT);
      END;
```

```
P3@.PTR := NIL;
IF HEAD1 = P1 THEN HEAD1 := HEAD3;
REMOVE (P1);
100:  END;  (*           OF SUBSTITUTE        *)
PROCEDURE TRUNKMARK(HEAD:  POINTER; I,M: INTEGER;
          VAR Z: ARRAY[1..100] OF POINTER; VAR N: INTEGER);
  LABEL 100;
  VAR   SUM,G,H,K  : INTEGER;
        P,P1          : POINTER;
BEGIN
P := HEAD; P1 := HEAD; N := 0;
WHILE P <> NIL DO
  BEGIN
  G := P@.EXP;
  SUM := 0;
  FOR K := 1 TO DIM DO
    BEGIN
    H := G MOD 10;
    G := G DIV 10;
    SUM := SUM + H;
    END;
  IF SUM > M THEN
    BEGIN
    IF HEAD = P THEN REMOVE (HEAD)
                ELSE
                  BEGIN
                  REMOVE (P);
                  P1@.PTR := NIL;
                  END;
    GOTO 100;
    END;
  N := N + 1;
  Z[N] := P;
  FOR K := 1 TO WTERMS[I] DO
    BEGIN
    P1 := P;
    P := P@.PTR;
    END;
  P1@.PTR := NIL;
  END;
100:  END;   (*              OF TRUNKMARK            *)
 (*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*)
 BEGIN     (*!!!!!!!!!!MAINLINE!!!!!!!!!!!!!!!!*)
TENDIM := 1;
WRITELN('WHAT IS THE DIMENSION OF THE PROBLEM?');
READ(DIM);
FOR I := 1 TO DIM DO TENDIM := TENDIM * 10;
WRITELN ('DO YOU WISH TO CHANGE YOUR THETA EXPANSION?');
WRITELN ('ANSWER Y OR N FOR YES OR NO RESPECTIVELY.');
REPLY := ' ';
WHILE REPLY=' ' DO READ(REPLY);
```

```
IF (REPLY='Y') THEN
  BEGIN
  WRITELN('DO YOU WANT TO ENTER YOUR OWN THETA EXPANSION?');
  REPLY := ' '; WHILE REPLY = ' ' DO READ(REPLY);
  IF REPLY = 'Y'
    THEN
      BEGIN
      WRITELN('ENTER THE THETA EXPANSION.  TERMINATE WITH');
      WRITELN('A NEGATIVE VALUE.');
      REWRITE(D1);READ(D1@);
      WHILE  D1@ >= 0 DO
        BEGIN
        PUT(D1);
        READ(D1@);
        END;
      END
    ELSE
      BEGIN
      TOTALMXPWR := FALSE;
      WRITELN('DO YOU WANT A TOTAL MAXPWR TRUNCATION?  IF NOT,');
      WRITELN('WE WILL TRUNCATE AT A MAXIMUM POWER OF EACH');
      WRITELN('VARIABLE.');
      REPLY := ' ';WHILE REPLY = ' ' DO READ (REPLY);
      IF REPLY = 'Y' THEN TOTALMXPWR := TRUE;
      WRITELN ('WHAT IS THE TOTAL MAXIMUM POWER OF YOUR EXPANSION\
    \?');
      READ (TTLMAX);
      STORETHETA(DIM,TTLMAX,D1);
      END;
  END;
RESET (D1);
TTERMS := 0;
WHILE NOT EOF (D1) DO
  BEGIN
  TTERMS := TTERMS + 1;
  E := D1@; GET (D1);
  WRITE (E);
  END;
WRITELN;
WRITELN('DO YOU WISH TO CHANGE YOUR FUNCTIONS F[I]?');
REPLY := ' '; WHILE REPLY = ' ' DO READ(REPLY);
IF REPLY = 'Y' THEN
  BEGIN
  WRITELN   ('ENTER COEFFICIENT AND EXPONENTS OF EACH TERM OF F[I\
  \] ');
  WRITELN   ('AS FOLLOWS:');
  WRITELN   ('          COEFF                EXPONENT');
  WRITELN;
  WRITELN   ('ENTER THE EXPONENT AS AN INTEGER, EACH DIGIT\
  \ REPRESENT-');
  WRITELN   ('ING THE POWER OF AN X-VARIABLE');
```

```
  WRITELN   ('TERMINATE WITH A COEFF-FIELD OF 5E10 AND NEGATIVE\
   \ EXP');
  REWRITE   (D2); REWRITE (D3);
  FOR   I:= 1 TO DIM DO
    BEGIN
    WRITELN ('ENTER TERMS OF FUNCTION F[',I:2,']:');
    C := 0.0; E := 0;
    REPEAT
      READ (C); READ (E);
      D2@ := C; PUT (D2);
      D3@ := E; PUT (D3);
    UNTIL (E<0);
    END;
  END;
RESET (D2); RESET (D3);
WRITELN ('ECHO DATA:');
FOR I := 1 TO DIM DO
  BEGIN
  WRITELN ('THE FUNCTION F [',I:2,'] IS:');
  WHILE (D2@<5E10) AND (D3@>=0) DO
    BEGIN
    C := D2@; GET (D2);
    E := D3@; GET (D3);
    WRITELN (C,E);
    END;
  GET (D2); GET (D3);
  END;
MANIFOLD := FALSE;
WRITELN('DO YOU WANT TO IMPOSE A MANIFOLD RESTRICTION?');
REPLY := ' ';
WHILE REPLY=' ' DO READ(REPLY);
WRITELN('REPLY IS:',REPLY);
IF (REPLY = 'Y') THEN MANIFOLD := TRUE;
IF MANIFOLD = TRUE THEN
  BEGIN
  WRITELN('HOW MANY MANIFOLD RESTRICTIONS DO YOU WANT?');
  READ(NW);
  FOR I := 1 TO NW DO
    BEGIN
    WRITELN('WHAT IS THE MAXPWR OF W[',I:2,']?');
    READ(MAXUW[I]);
     WRITELN('MAXUW[',I:2,'] =',MAXUW[I]);
    COND[I]   := FALSE;
    WRITELN;WRITELN('DO YOU WANT UW=0 WITH CONDITION W[',I:2,']=0\
   \?');
    REPLY   := ' ';WHILE REPLY=' ' DO READ(REPLY);
    WRITELN('REPLY   IS:',REPLY);
    IF   REPLY = 'Y' THEN COND[I] := TRUE;
    WRITELN ('ENTER THE EXPRESSION FOR THE MANIFOLD W[',I:2,']');
    WRITELN('WITH THE HIGHEST MAXPWR TERM FIRST.');
    W[I]:=NIL;Z1:=NIL;Z:=NIL;WTERMS[I]:=0;
```

```
    READ (C,E);
    WHILE (C < 5E10) AND (E >= 0) DO
      BEGIN
      WTERMS[I] := WTERMS[I] + 1;
      NEW(Z);
      Z@.COEFF := C; Z@.EXP := E; Z@.PTR := NIL;
      IF W[I]=NIL THEN W[I] := Z
               ELSE Z1@.PTR := Z;
      Z1 := Z; READ(C,E);
      END;
    WRITELN ('ECHO DATA');
    WRITELN;WRITELN('THE ENTERED W[',I:2,'] IS:');
    PRINT1(W[I],-1);
    END;
  END;
(*!!!!!!!!!!!!!!!!!!SOLVE FOR DETERMINING EQUATIONS!!!!!!!!!!!!!!\
    \!!!!*);
WRITELN ('WHAT TOTAL MAXIMUM POWER DO YOU WISH TO TRUNCATE ON?');
READ (MAXPWR);
FOR I := 1 TO NW DO
  BEGIN
    MAXUW[I] := MAXUW[I] + MAXPWR - 1;
     WRITELN('MAXPWR =',MAXPWR);
      WRITELN('MAXUW[',I:2,'] =',MAXUW[I]);
   END;
WRITELN ('DO YOU WANT THE EQUATIONS PRINTED OUT?');
REPLY := ' ';
WHILE REPLY=' ' DO READ (REPLY);
WRITELN('REPLY IS:',REPLY);
PEQNS := FALSE;
IF REPLY = 'Y' THEN PEQNS := TRUE;
NOCOL:=0;              (*      INITIALIZE COLUMN HEADS        *)
FOR I:=1 TO DIM DO
  BEGIN
  RESET(D1);
  WHILE NOT EOF(D1) DO
    BEGIN
    NOCOL:=NOCOL+1;
    COL[NOCOL] := I+(D1@/TENDIM);
    GET (D1);
    END;
  END;
T := 0;
FOR I := 1 TO NW DO
  BEGIN
  UW[I] := NIL; W1[I] := NIL;
  END;
REWRITE(MAT);
FOR I := 1 TO DIM DO
  BEGIN
  WRITELN;WRITELN;WRITELN;
```

```
WRITELN ('DET. EQNS FOR I =',I:2,' ARE :');
RESET (D2);RESET(D3);
THEAD[I] := NIL;
CREATETHETA (THEAD[I],I);
PHEAD := NIL;
FOR K:= 1 TO DIM DO
  BEGIN
  TDHEAD[I,K] := NIL;
  DERIV (THEAD[I],TDHEAD[I,K],K);
  FHEAD[K] := NIL;
  CREATEF (FHEAD[K],K );
  MULTPOLYS (FHEAD[K],TDHEAD[I,K],PHEAD,MAXPWR);
  REMOVE(FHEAD[K]);
  REMOVE (TDHEAD[I,K]);
  END;
IF MANIFOLD THEN
  FOR J := 1 TO NW DO
    BEGIN
    DERIV(W[J],DW[J,I],I);
    MULTPOLYS(DW[J,I],THEAD[I],UW[J],MAXUW[J]);
    REMOVE(DW[J,I]);
    END;
REMOVE (THEAD[I]);
FHEAD[I] := NIL;
CREATEF (FHEAD[I],I);
NEGATE (FHEAD[I]);
FOR K := 1 TO DIM DO
  BEGIN
  DERIV (FHEAD[I],FDHEAD[I,K],K);
  THEAD[K] := NIL;
  CREATETHETA (THEAD[K],K);
  MULTPOLYS (FDHEAD[I,K],THEAD[K],PHEAD,MAXPWR);
  REMOVE(FDHEAD[I,K]);
  REMOVE (THEAD[K]);
  END;
REMOVE (FHEAD[I]);
SORT (PHEAD,TRUE);      (*  SORT ON X'S  *)
CREATETHETA(THEAD[0],0);
FOR J := 1 TO NW DO
  IF COND[J] THEN
    BEGIN
    MULTPOLYS(THEAD[0],W[J],W1[J],100);
    TRUNKMARK(W1[J],J,MAXPWR,REQ,L);
    FOR K := 1 TO L DO
      BEGIN
      SUBSTITUTE(PHEAD,REQ[K]);
      REMOVE(REQ[K]);
      END;
    W1[J] := NIL;
    SORT(PHEAD,TRUE);
    END;
```

```
  SEPERATETERMS (PHEAD);
  CREATEMATRIX (NOEQNS);
  END;
IF MANIFOLD = TRUE THEN
  FOR J := 1 TO NW DO
    BEGIN
    SORT (UW[J],TRUE);
    WRITELN;WRITELN;WRITELN;
    WRITELN('THE CONDITION FOR W[',J:2,']=0 IS:');
    IF COND[J] THEN
      BEGIN
      MULTPOLYS(THEAD[0],W[J],W1[J],100);
      REMOVE(W[J]);
      TRUNKMARK(W1[J],J,MAXUW[J],REQ,L);
      FOR K := 1 TO L DO
        BEGIN
        SUBSTITUTE(UW[J],REQ[K]);
        REMOVE(REQ[K]);
        END;
      W1[J] := NIL;
      SORT(UW[J],TRUE);
      END;
    SEPERATETERMS(UW[J]);
    CREATEMATRIX(NOEQNS);
    END;
REMOVE(THEAD[0]);
RESET(MAT);
K := T* NOCOL;
WRITELN('THE SIZE OF THE MATRIX IS:',T:4,' X',NOCOL:4,'=',K);
END.
#
```

Program MFGR

```
#FILE (A015006)MFGRDETEQNS7 ON SWAT
(*$U-*)
PROGRAM      MFGR(INPUT,OUTPUT);
   LABEL   500;
   CONST   MROWS  =  89; MCOLS = 140;
           EPS = 1.0E-7;
   TYPE    REALFILE = FILE OF REAL;
           INTEGERFILE = FILE OF INTEGER;
   VAR     MAT : REALFILE;
           GEN:  REALFILE;
           D1:  INTEGERFILE;
           RVAR:  REALFILE;
           IVAR:  INTEGERFILE;
           ROWVECTOR:  INTEGERFILE;
           COLVECTOR:  INTEGERFILE;
           COL:  ARRAY[1..MCOLS] OF REAL;
           REPLY   : CHAR;
           DONE  : BOOLEAN;
           T,T1,N,Q,COUNT,E,MAXPWR,TTLMAX,TENDIM,NOEQNS:  INTEGER;
           H,ROUNDS,NOCOL,IRANK,DIM  :  INTEGER;
           MATRIX:  ARRAY[1..MROWS,1..MCOLS] OF REAL;
           IROW:  ARRAY[1..MROWS] OF INTEGER;
           ICOL:  ARRAY[1..MCOLS] OF INTEGER;
           PIV,HOLD,TOL,SAVE:  REAL;
           I,II,J,JJ,K,KK,L,LL,IR,IC,NM,MM,NCOL,P:  INTEGER;
PROCEDURE MFGR(VAR A: ARRAY[1..MROWS,1..MCOLS] OF REAL;
                M:  INTEGER);
  LABEL  5,10,25,100;
BEGIN
IF (H >= 1) AND (H <= NOCOL) THEN GOTO 5;
H := 1;
(*          TEST OF SPECIFIED DIMENSIONS          *)
IF (M<=0) OR (NOCOL<=0) THEN
  BEGIN    (*    RETURN IN CASE OF FORMAL ERRORS    *)
  IRANK := -1;
  WRITELN('ERROR IN MATRIX DIMENSIONS!!!');
  GOTO 100;
  END;


(*           INITIALIZE COLUMN INDEX VECTOR          *)
 (*           SEARCH FIRST PIVOT ELEMENT          *)
IRANK := 0;
PIV := 0.0;
JJ := 0;
FOR J := 1 TO NOCOL DO
  BEGIN
  ICOL[J] := J;
  FOR I := 1 TO M DO
```

```
    BEGIN
    HOLD := A[I,J];
    IF (ABS(PIV)-ABS(HOLD)) < 0 THEN
      BEGIN
      PIV := HOLD;
      IR := I;
      IC := J;
      END;
    END;
  END;
(*            INITIALIZE ROW INDEX VECTOR*)
FOR I := 1 TO M DO IROW[I] := I;

(*            SET UP INTERNAL TOLERANCE*)
TOL := ABS(EPS*PIV);

(*            INITIALIZE ELIMINATION LOOP*)
 5: FOR NCOL := M TO NOCOL DO
  BEGIN(*       TEST FOR FEASIBILITY OF PIVOT ELEMENT        *)
  IF ROUNDS <= (NCOL-M) THEN GOTO 100;
  IF (ABS(PIV)-TOL) <= 0
    THEN GOTO 10
    ELSE
      BEGIN
      IRANK := IRANK +1;  (*      UPDATE RANK*)

      IF (IR>IRANK) THEN         (*   INTERCHANGE ROWS IF\
   \ NECESSARY  *)
        BEGIN
        FOR J := 1 TO NOCOL DO
          BEGIN
          SAVE := A[IRANK,J];
          A[IRANK,J] := A[IR,J];
          A[IR,J] := SAVE;
          END;
(*                 UPDATE ROW INDEX VECTOR              *)
        JJ := IROW[IR];
        IROW[IR] := IROW[IRANK];
        IROW[IRANK] := JJ;
        END;
(*                INTERCHANGE COLUMNS IF NECESSARY*)
      IF (IC>IRANK) THEN
        BEGIN
        FOR J := 1 TO M DO
          BEGIN
          SAVE := A[J,IRANK];
          A[J,IRANK] := A[J,IC];
          A[J,IC] := SAVE;
          END;
(*                  UPDATE COLUMN INDEX VECTOR                 *)
        JJ := ICOL[IC];
```

```
          ICOL[IC] := ICOL[IRANK];
          ICOL[IRANK] := JJ;
          END;
        KK := IRANK + 1;


        IF (IRANK >= M) THEN GOTO 25; (*  TEST FOR LAST ROW  *)


         (* TRANSFORM CURRENT SUBMATRIX AND SEARCH NEXT PIVOT *)

        SAVE := PIV;
        PIV := 0.0;
        FOR J := KK TO M DO
          BEGIN
          HOLD := A[J,IRANK]/SAVE;
          A[J,IRANK] := HOLD;

          (* TEST FOR LAST COLUMN *)
          IF (IRANK<NOCOL) THEN
            BEGIN
            FOR I := KK TO NOCOL DO
              BEGIN
              A[J,I] := A[J,I] - HOLD*A[IRANK,I];
              IF (ABS(A[J,I])-ABS(PIV)) > 0 THEN
                BEGIN
                PIV := A[J,I];
                IR := J;
                IC := I;
                END;
              END;
            END;
          END;
        END;
    END;
(*          SET UP MATRIX EXPRESSING ROW DEPENDENCIES              \
   \*)
10:  DONE := TRUE;
IF (IRANK<1) THEN
  BEGIN
  WRITELN('FORMAL ERROR :   RANK<1');
  GOTO 100; (*  EXIT  *)
  END;
IF (IRANK = 1) THEN GOTO 25;
IR := IRANK;
FOR J := 2 TO IRANK DO
  BEGIN
  IR := IR - 1;
  JJ := IRANK;
  FOR I := KK TO M DO
    BEGIN
```

```
      HOLD := 0.0;
      JJ := JJ + 1;
      MM := IRANK;
      IC := IRANK;
      FOR L := 1 TO (J-1) DO
        BEGIN
        HOLD := HOLD + (A[JJ,MM]*A[IC,IR]);
        IC := IC - 1;
        MM := MM - 1;
        END;
      A[JJ,MM] := A[JJ,MM] - HOLD;
      END;
    END;
(*                TEST FOR COLUMN REGULARITY*)
25  :  IF (NOCOL-IRANK) <= 0 THEN
    BEGIN
    WRITELN('ERROR!! COLUMN IRREGULARITY.');
    GOTO 100;(*  EXIT  *)
    END;
DONE := TRUE;

(* SET UP MATRIX EXPRESSING BASIC VARIABLES IN TERMS OF FREE *)
 (*          PARAMETERS (HOMOGENEOUS SOLUTION).           *)
IR := IRANK;
FOR J := 1 TO IRANK DO
    BEGIN
    FOR I := (IRANK + 1) TO NOCOL DO
      BEGIN
      JJ := IRANK;
      LL := IRANK;
      HOLD := 0.0;
      FOR II := (J - 1) DOWNTO 1 DO
        BEGIN
        HOLD := HOLD-A[IR,JJ]*A[LL,I];
        JJ := JJ - 1;
        LL := LL - 1;
        END;
      A[LL,I] := (HOLD-A[LL,I])/A[IR,JJ];
      END;
    IR := IR - 1;
    END;
 100:   END;  (*        OF MFGR         *)
 (*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*)
BEGIN
TENDIM := 1;DONE := FALSE;
WRITELN('WHAT IS THE DIMENSION OF THE PROBLEM?');
READ(DIM);
FOR N := 1 TO DIM DO TENDIM := TENDIM * 10;
WRITELN('IS THIS A NEW PROBLEM?');
READ(REPLY); WHILE REPLY = ' ' DO READ(REPLY);
IF (REPLY = 'N') THEN
```

```
  BEGIN
  RESET(IVAR);
  I  := IVAR@;  GET(IVAR);          II := IVAR@;  GET(IVAR);
  J  := IVAR@;  GET(IVAR);          JJ := IVAR@;  GET(IVAR);
  K  := IVAR@;  GET(IVAR);          KK := IVAR@;  GET(IVAR);
  L  := IVAR@;  GET(IVAR);          LL := IVAR@;  GET(IVAR);
  IR := IVAR@;  GET(IVAR);          IC := IVAR@;  GET(IVAR);
  NM := IVAR@;  GET(IVAR);          MM := IVAR@;  GET(IVAR);
  H  := IVAR@;  GET(IVAR);          P  := IVAR@;  GET(IVAR);
  NOCOL := IVAR@; GET(IVAR);        IRANK := IVAR@; GET(IVAR);
  T  := IVAR@;  GET(IVAR);
  RESET(RVAR);
  PIV := RVAR@;  GET(RVAR);     HOLD := RVAR@;  GET(RVAR);
  TOL := RVAR@;  GET(RVAR);     SAVE := RVAR@;  GET(RVAR);
  RESET(ROWVECTOR);
  FOR N := 1 TO T DO
    BEGIN
    IROW[N] := ROWVECTOR@;
    GET(ROWVECTOR);
    END;
  RESET(COLVECTOR);
  FOR N := 1 TO NOCOL DO
    BEGIN
    ICOL[N] := COLVECTOR@;
    GET(COLVECTOR);
    END;
  END;
NOCOL:=0;              (*      INITIALIZE COLUMN HEADS       *)
FOR N:=1 TO DIM DO
  BEGIN
  RESET(D1);
  WHILE NOT EOF(D1) DO
    BEGIN
    NOCOL:=NOCOL+1;
    COL[NOCOL] := N+(D1@/TENDIM);
    GET (D1);
    END;
  END;
RESET(MAT);
T := 0;
WHILE NOT EOF(MAT) DO
  BEGIN
  T := T + 1;
  FOR N := 1 TO NOCOL DO
    BEGIN
    MATRIX[T,N] := MAT@;
    GET(MAT);
    END;
  END;
T1 := T * NOCOL;
```

```
WRITELN('HOW MANY ITERATIONS DO YOU WANT?');
READ(ROUNDS);
MFGR(MATRIX,T);
IF NOT DONE THEN
  BEGIN
  REWRITE(IVAR);
  IVAR@ :=  I;  PUT(IVAR);          IVAR@ := II;  PUT(IVAR);
  IVAR@ :=  J;  PUT(IVAR);          IVAR@ := JJ;  PUT(IVAR);
  IVAR@ :=  K;  PUT(IVAR);          IVAR@ := KK;  PUT(IVAR);
  IVAR@ :=  L;  PUT(IVAR);          IVAR@ := LL;  PUT(IVAR);
  IVAR@ := IR;  PUT(IVAR);          IVAR@ := IC;  PUT(IVAR);
  IVAR@ := NN;  PUT(IVAR);          IVAR@ := MM;  PUT(IVAR);
  IVAR@ := NCOL; PUT(IVAR);         IVAR@ :=  P;  PUT(IVAR);
  IVAR@ := NOCOL; PUT(IVAR);        IVAR@ := IRANK; PUT(IVAR);
  IVAR@ :=  T;  PUT(IVAR);
  RESET(IVAR);
  REWRITE(RVAR);
  RVAR@ := PIV;  PUT(RVAR);          RVAR@ := HOLD;  PUT(RVAR);
  RVAR@ := TOL;  PUT(RVAR);          RVAR@ := SAVE;  PUT(RVAR);
  RESET(RVAR);
  REWRITE(ROWVECTOR);
  FOR N := 1 TO T DO
    BEGIN
    ROWVECTOR@ := IROW[N];
    PUT(ROWVECTOR);
    END;
  RESET(ROWVECTOR);
  REWRITE(COLVECTOR);
  FOR N := 1 TO NOCOL DO
    BEGIN
    COLVECTOR@ := ICOL[N];
    PUT(COLVECTOR);
    END;
  RESET(COLVECTOR);
  REWRITE(MAT);
  FOR Q := 1 TO T DO
    FOR N := 1 TO NOCOL DO
    BEGIN
    MAT@ := MATRIX[Q,N];
    PUT(MAT);
    END;
  RESET(MAT);
  GOTO 500;   (*EXIT*)
  END;
WRITELN('THE RANK IS:',IRANK);
WRITELN('THE RELATION VECTOR IS:');
FOR N := 1 TO NOCOL DO WRITE(ICOL[N]:5);
WRITELN;
WRITELN('THE RESULTING EQUATIONS ARE:');
WRITELN; WRITELN; WRITELN; WRITELN;
COUNT := 0;
```

```
REWRITE(GEN);
FOR N := (IRANK + 1) TO NOCOL DO
  BEGIN
  IF COL[ICOL[N]] >= 1 THEN
    BEGIN
    COUNT := COUNT + 1;
    WRITE('U[',COUNT:4,'] = (A',COL[ICOL[N]]:5,') + ');
    GEN@:= 1.0; PUT(GEN);
    GEN@:= COL[ICOL[N]]; PUT(GEN);
    FOR Q := 1 TO IRANK DO
      BEGIN
      IF (ABS(MATRIX[Q,N]) > 1E-5) THEN
        BEGIN
        WRITE(MATRIX[Q,N]:5,'(A',COL[ICOL[Q]]:5,') + ');
        GEN@ := MATRIX[Q,N]; PUT(GEN);
        GEN@ := COL[ICOL[Q]]; PUT(GEN);
        END;
      END;
    WRITELN;WRITELN;
    GEN@ := 5E10 ; PUT(GEN);
    END;
  END;
RESET(GEN);
WRITELN('DO YOU WANT TO SEE THE MATRIX?');
REPLY := ' ';
WHILE REPLY = ' ' DO READ (REPLY);
IF (REPLY='Y') THEN
  BEGIN
  WRITELN('THE INPUT MATRIX WAS:');
  RESET(MAT);
  FOR N := 1 TO NOCOL DO WRITE (COL[N]:5);
  WRITELN;
  FOR N := 1 TO NOCOL DO WRITE('*****');WRITELN;
  FOR Q := 1 TO T DO
    BEGIN
    FOR N := 1 TO NOCOL DO
      BEGIN
      WRITE(MAT@:5);
      GET (MAT);
      END;
    WRITELN;
    END;
  WRITELN;
  WRITELN('THE PROCESSED MATRIX IS:');
  WRITELN;WRITELN('L*U IS:');
  FOR Q := 1 TO IRANK DO
    BEGIN
    FOR N := 1 TO IRANK DO
      BEGIN
      WRITE(MATRIX[Q,N]:5);
      END;
```

```
    WRITELN;
    END;
  WRITELN;WRITELN('C MATRIX IS:');
  FOR N := 1 TO IRANK DO WRITE (IROW[N]:5);
  WRITELN;
  FOR N := 1 TO IRANK DO WRITE('*****');WRITELN;
  FOR Q := (IRANK + 1) TO T DO
    BEGIN
    FOR N := 1 TO IRANK DO
      BEGIN
      WRITE(MATRIX[Q,N]:5);
      END;
    WRITELN;
    END;
  WRITELN('THE IROW VECTOR IS:');
  FOR N := 1 TO T DO WRITE (IROW[N]:5);
  WRITELN;WRITELN;WRITELN;
  WRITELN('THE HOMOGENOUS SOLN MATRIX IS:');
  FOR N := (IRANK + 1) TO NOCOL DO WRITE(COL[ICOL[N]]:5);
  WRITELN;
  FOR N := (IRANK + 1) TO NOCOL DO WRITE('*****'); WRITELN;
  FOR Q := 1 TO IRANK DO
    BEGIN
    FOR N := (IRANK + 1) TO NOCOL DO
      BEGIN
      WRITE(MATRIX[Q,N]:5);
      END;
    WRITELN;
    END;
  END;
REWRITE(MAT);  (*  DISPOSE OF DISK SPACE  *)
REWRITE(RVAR);
REWRITE(IVAR);
REWRITE(ROWVECTOR);
REWRITE(COLVECTOR);
500: END.
#
```

Program COMMUTATOR

```
#FILE (A015006)COMMUTATOR2 ON SWAT
(*$ N 064 *)
(*$ SPACE 30000*)
(*$ U- *)
PROGRAM COMMUTATOR (INPUT,OUTPUT);
  CONST NOGEN = 500;
  TYPE  POINTER = @TERM;
        TERM    = RECORD
                    COEFF : REAL;
                    A     : REAL;
                    EXP   : INTEGER;
                    PTR   : POINTER;
                  END;
        REALFILE = FILE OF REAL;
  VAR   GEN:  REALFILE;
        U:  ARRAY[1..NOGEN] OF POINTER;
        NGEN,I,DIM,H,J,TENDIM:  INTEGER;
        P,PROD:  POINTER;
        DU:  ARRAY[1..NOGEN,1..8] OF POINTER;
        G:  REAL;
        B:  ARRAY[0..NOGEN] OF INTEGER;
        REPLY: CHAR;
  PROCEDURE PRINT(HEAD:  POINTER);
    VAR  P:  POINTER;
    BEGIN
    P := HEAD;
    WHILE P<>NIL DO
      BEGIN
  WRITE(P@.COEFF:5,'(A',P@.A:5,') + ');
      P := P@.PTR;
      END;
    WRITELN;WRITELN;WRITELN;
    END;  (*  OF PRINT  *)
  PROCEDURE CREATEU;
    VAR  P1,P2:  POINTER;
           I    :  INTEGER;
    BEGIN
    RESET(GEN);
    I := 0;
    WHILE NOT EOF(GEN) DO
      BEGIN
      I := I+1; U[I] := NIL; P1 := U[I];
      B[I] := I;
      WHILE GEN@ < 5E10 DO
        BEGIN
        NEW(P2);
        P2@.COEFF := GEN@;
        GET(GEN);
        P2@.A := GEN@;
        G := P2@.A - TRUNC(P2@.A);
```

```
        G := G*TENDIM;
        P2@.EXP := ROUND(G);
        P2@.PTR := NIL;
        IF P1 <> NIL THEN P1@.PTR := P2
                     ELSE U[I] := P2;
        P1 := P2;
        GET(GEN);
        END;
      GET(GEN);
      END;
  NGEN := I;
  END;
  PROCEDURE CREATEU2;
    VAR  P1,P2:  POINTER;
         I,J,K    :  INTEGER;
    BEGIN
    RESET(GEN);
    FOR I := 1 TO NGEN DO
      BEGIN
      K := I - 1;
      FOR J := B[K] TO (B[I] - 1) DO
        BEGIN
        WHILE GEN@ < 5E10 DO GET(GEN); GET (GEN);
        END;
      U[I] := NIL; P1 := NIL;
      WHILE GEN@ < 5E10 DO
        BEGIN
        NEW(P2);
        P2@.COEFF := GEN@;
        GET(GEN);
        P2@.A := GEN@;
        G := P2@.A - TRUNC(P2@.A);
        G := G*TENDIM;
        P2@.EXP := ROUND(G);
        P2@.PTR := NIL;
        IF P1 <> NIL THEN P1@.PTR := P2
                     ELSE U[I] := P2;
        P1 := P2;
        GET(GEN);
        END;
      END;
    END;
PROCEDURE REMOVE (VAR HEAD : POINTER);
  VAR      P  : POINTER;
BEGIN
P := HEAD;
WHILE HEAD <> NIL DO
  BEGIN
  HEAD := HEAD@.PTR;
  DISPOSE(P);
  P:=HEAD;
```

```
  END;
END;    (*  OF REMOVE  *)
PROCEDURE SORT(VAR HEAD: POINTER);
 LABEL 10;
  VAR     P,Q,R,P1: POINTER;
BEGIN
10:  P := HEAD;
WHILE P <> NIL DO
  BEGIN
  R := P;  Q := P@.PTR;
  WHILE Q <> NIL DO
    IF Q@.A = P@.A
      THEN
        BEGIN
        P@.COEFF := P@.COEFF + Q@.COEFF;
        R@.PTR := Q@.PTR;
        DISPOSE(Q);
        Q := R@.PTR;
        END
      ELSE
        BEGIN
        R := R@.PTR;
        Q := Q@.PTR;
        END;
  IF ABS(P@.COEFF) <= 1E-10 THEN
    BEGIN
    IF P = HEAD THEN
      BEGIN
      HEAD := HEAD@.PTR;
      DISPOSE(P);
      GOTO 10;
      END;
    P1@.PTR := P@.PTR;
    DISPOSE(P);
    P := P1;
    END;
  P1 := P;
  P := P@.PTR;
  END;
END;  (*    OF SORT    *)
PROCEDURE DERIV (HEAD :POINTER; VAR  DHEAD : POINTER;  K :\
  \ INTEGER);
  LABEL  100;
  VAR    P,Q,Q1            : POINTER;
         H,I                  : INTEGER;
BEGIN
P := HEAD; Q := NIL; Q1 := NIL;
WHILE P<>NIL DO
  BEGIN
  NEW(Q);
  WITH Q@ DO
```

```
    BEGIN
    H := P@.EXP;
    FOR I := 1 TO (DIM - K) DO H := H DIV 10;
    H := H MOD 10;
    COEFF := P@.COEFF*H;
    IF COEFF = 0.0 THEN
      BEGIN
       DISPOSE (Q) ;
      GOTO 100;
      END;
    H := 1;
    FOR I:=1 TO (DIM-K) DO H := H*10;
    EXP := P@.EXP - H;
    A := TRUNC(P@.A) + (EXP/TENDIM);
      PTR := NIL;
      END;
  IF DHEAD = NIL THEN DHEAD := Q
                 ELSE Q1@.PTR := Q;
  Q1 :=Q;
100:  P:=P@.PTR;
  END;
END;   (*  OF DERIV  *)
PROCEDURE MULTPOLYS(HEADF,HEADT:POINTER; VAR PHEAD:POINTER);
  LABEL  110;
  VAR    P,Q,R,R1,P1       : POINTER;
         H,H1,H2,SUM,G1,G2,I,J,K   : INTEGER;
BEGIN
P := HEADF;P1:=HEADF;
R := PHEAD;
Q := HEADT;
IF R<>NIL THEN WHILE R@.PTR <>NIL DO
  BEGIN
  R := R@.PTR;
  END;
R1 := R;
Q := HEADT;
WHILE Q<>NIL DO
  BEGIN
  H := 0; SUM := 0; G1 := P@.EXP; G2 := Q@.EXP;
  FOR I := 1 TO DIM DO
    BEGIN
    H1 := G1 MOD 10; G1 := G1 DIV 10;
    H2 := G2 MOD 10; G2 := G2 DIV 10;
    H1 := H1 + H2;
    J := 1;
    FOR K := 1 TO (I-1) DO J := J*10;
    H := H + (H1 * J);
    END;
  NEW(R);
  WITH R@ DO
    BEGIN
```

```
      A := TRUNC(Q@.A) +(H/TENDIM);
      COEFF := P@.COEFF * Q@.COEFF;
      EXP := H;
      PTR := NIL;
      END;
    IF R@.COEFF = 0 THEN
      BEGIN
      DISPOSE (R);
      GOTO 110;
      END;
    IF PHEAD = NIL THEN PHEAD := R
                   ELSE R1@.PTR := R;
    R1 := R;
    110:  Q := Q@.PTR;
    END;
END;     (*  OF MULTPOLYS  *)
(***************************MAINLINE***************************)


BEGIN
WRITELN('WHAT IS THE DIMENSION OF THE PROBLEM?');
READ(DIM);
TENDIM := 1;
FOR I := 1 TO DIM DO TENDIM := TENDIM *10;
WRITELN('DO YOU WANT THE COMMUTATORS OF ALL GENERATORS?');
REPLY := ' ';
WHILE REPLY = ' ' DO READ (REPLY);
IF REPLY = 'Y'
  THEN CREATEU
  ELSE
    BEGIN
    WRITELN('HOW MANY GENERATORS DO YOU WANT?');
    READ(NGEN);
    WRITELN('ENTER THE #S OF THE GENERATORS');
    B[0] := 1;
    FOR I := 1 TO NGEN DO READ(B[I]);
    CREATEU2;
    END;
FOR I := 1 TO NGEN DO (*  WRITE GENERATORS  *)
  BEGIN
  WRITE('U[',B[I]:4,'] =');
  PRINT(U[I]);
  END;

WRITELN('*********************************************************');
  (*  CALCULATE DERIVATIVES OF ALL GENERATORS  *)
FOR I := 1 TO NGEN DO
  BEGIN
  FOR J := 1 TO DIM DO
    BEGIN
    DERIV(U[I],DU[I,J],J);
```

```
      END;
    END;

    (*  CALCULATE COMMUTATORS  *)
FOR I := 1 TO NGEN DO
    BEGIN
    FOR J := I+1 TO NGEN DO
      BEGIN
      WRITE('[U(',B[I]:2,'),U(',B[J]:2,')] = ');
      P := U[I];
      WHILE P <> NIL DO
        BEGIN
        H := TRUNC(P@.A);
        MULTPOLYS (P,DU[J,H],PROD);
        P := P@.PTR;
        END;
      P := U[J];
      WHILE P <> NIL DO
        BEGIN
        H := TRUNC(P@.A);
        P@.COEFF := -P@.COEFF;
        MULTPOLYS(P,DU[I,H],PROD);
        P := P@.PTR;
        END;
      SORT(PROD);
      PRINT(PROD);
      REMOVE(PROD);
      END;
    END;
END.  (*  OF PROGRAM  *)
#
```

Program DETERMININGEQNSB

```
#FILE (A015006)DETEQNS7B ON SWAT
   (*$U-*)
   (*$ N 064 *)
   (*$ SPACE 30000*)
PROGRAM DETERMININGEQNSB(INPUT,OUTPUT);
    CONST   MAXDIM   =   8;
            NOTERMS  =1023;   (*  NOTERMS = (MAXPWR+1)**MAXDIM  *)
            SIGDIG = 2;   (*  # OF SIGNIFICANT DIGITS   *)
            EPS = 1.0E-7;
            NODIGITS = 10000000;
   TYPE     POINTER  =  @TERM;
            COEFFICIENTS = 0..NODIGITS;
            TERM  =  RECORD
                       COEFF   : COEFFICIENTS;
                       A       : REAL;
                       EXP     : COEFFICIENTS;
                       PTR     : POINTER;
                     END;
            DATA1FILE = FILE OF INTEGER;
            DATA2FILE = FILE OF REAL;
            DATA3FILE = FILE OF INTEGER;
            MATRIXFILE = FILE OF COEFFICIENTS;
  VAR       C,B,B1,B2,B3         : REAL;
            D1        : DATA1FILE;
            D2        : DATA2FILE;
            D3        : DATA3FILE;
            MATB       : MATRIXFILE;
            REPLY    : CHAR;
            CH         : CHAR;
            MATRIX:PACKED ARRAY[1..NOTERMS] OF COEFFICIENTS;
            EQN: ARRAY [1..NOTERMS] OF POINTER;
            THEAD : ARRAY [0..MAXDIM] OF POINTER;
            REQ        : ARRAY [1..100] OF POINTER;
            FHEAD : ARRAY [1..MAXDIM] OF POINTER;
            TDHEAD,FDHEAD: ARRAY [1..MAXDIM,1..MAXDIM] OF POINTER;
            COL,ROW :  ARRAY[1..NOTERMS] OF REAL;
            T,T1,I,K,COUNT,E,L,MAXPWR,TTLMAX,TENDIM,NOEQNS,
            SD,SD1,DIM,NW,J,NOCOL,IRANK,TTERMS:  INTEGER;
            MAXUW:  ARRAY[1..10] OF INTEGER;
            IROW,ICOL:  ARRAY[1..NOTERMS] OF INTEGER;
            Z1,Z2,Z,PHEAD :  POINTER;
            PEQNS,MANIFOLD,TOTALMXPWR    :  BOOLEAN;
            COND:  ARRAY [1..10] OF BOOLEAN;
            UW,W,W1 : ARRAY[1..10] OF POINTER;
            DW : ARRAY [1..10,1..MAXDIM] OF POINTER;
            WTERMS : ARRAY [1..10] OF INTEGER;
PROCEDURE PRINT1 (HEAD:POINTER;J:INTEGER);
  VAR P       : POINTER;
      I,K     : INTEGER;
BEGIN
P := HEAD;
```

```
WRITELN('EQN[',J:DIM,'] IS:');
WRITELN('      COEFF          A        EXP');
WHILE P<> NIL DO
  BEGIN
  CASE(P@.COEFF MOD 10) OF
    0:  B1 := (P@.COEFF DIV 10)/SD;
    1:  B1 := -(P@.COEFF DIV 10)/SD;
  END;
  WRITELN(B1,P@.A,P@.EXP);
  P := P@.PTR;
  END;
END;  (*  OF PRINT1  *)
PROCEDURE REMOVE (VAR HEAD : POINTER);
  VAR      P,P1  : POINTER;
BEGIN
P1 := HEAD;
P := HEAD;
WHILE P <> NIL DO
  BEGIN
  P := P@.PTR;
  DISPOSE(P1);
  P1 := P;
  END;
HEAD := NIL;
END;    (*  OF REMOVE  *)
PROCEDURE NEGATE (VAR  HEAD : POINTER);
  VAR    P : POINTER;
BEGIN
P := HEAD;
WHILE P<>NIL DO
  BEGIN
  CASE(P@.COEFF MOD 10) OF
    0:  P@.COEFF := P@.COEFF + 1;
      1:  P@.COEFF := P@.COEFF - 1;
  END;
  P := P@.PTR;
  END;
END;    (*  OF NEGATE  *)
PROCEDURE ADD(HEAD1,HEAD2:POINTER);
  VAR    P:  POINTER;
BEGIN
P := HEAD1;
WHILE P@.PTR <> NIL DO P := P@.PTR;
P@.PTR := HEAD2;
END;  (*  OF ADD  *)
PROCEDURE CREATETHETA (VAR  THEAD : POINTER;  I : INTEGER);
   VAR  A     : INTEGER;
          P,P1 : POINTER;
BEGIN
 P1 := NIL; P:= NIL;
RESET (D1);
WHILE NOT EOF(D1) DO
```

```
WHILE NOT EOF(D1) DO
  BEGIN
  A := D1@; GET (D1);
  NEW(P);
  P@.EXP := A;
  P@.A := A/TENDIM + I;
  P@.COEFF := SD1;
  P@.PTR := NIL;
  IF P1<>NIL THEN P1@.PTR := P
                ELSE THEAD := P;
  P1 := P;
  END;
END;     (*  OF CREATETHETA  *)
PROCEDURE CREATEF (VAR  FHEAD : POINTER  ;  I : INTEGER);
  VAR      C        : REAL;
           P,P1     :  POINTER;
           COUNT,E  : INTEGER;
BEGIN
P := NIL; P1 := NIL;
RESET (D2); RESET (D3);
COUNT := 1; C := 0.0; E := 0;
WHILE COUNT < I DO
  BEGIN
  WHILE (D2@<5E10) AND (D3@>=0) DO
    BEGIN
    GET (D2);
    GET (D3);
    END;
  GET (D2); GET (D3);
  COUNT := COUNT + 1;
  END;
WHILE (D2@<5E10) AND (D3@>=0) DO
  BEGIN
  NEW(P);
  C := D2@; GET (D2);
  IF (C < 0)
    THEN P@.COEFF := ROUND(ABS(C*SD))*10 + 1
    ELSE P@.COEFF := ROUND(C*SD)*10;
  E := D3@; GET (D3);
  P@.EXP := E;
  P@.PTR := NIL;
  IF FHEAD = NIL THEN FHEAD := P
                  ELSE P1@.PTR := P;
  P1 := P;
  END;
END;  (*   OF CREATEF  *)
PROCEDURE DERIV (HEAD :POINTER; VAR  DHEAD : POINTER;  K :\
  \ INTEGER);
  LABEL  100;
  VAR    P,Q,Q1          : POINTER;
         H,I                : INTEGER;
```

```
P := HEAD; Q := NIL; Q1 := NIL;
WHILE P<>NIL DO
  BEGIN
  NEW(Q);
  WITH Q@ DO
    BEGIN
    H := P@.EXP;
    FOR I := 1 TO (DIM - K) DO H := H DIV 10;
    H := H MOD 10;
    CASE (P@.COEFF MOD 10) OF
      0:  COEFF := P@.COEFF * H;
      1:  COEFF := ((P@.COEFF DIV 10) * H) * 10 + 1;
    END;
    IF (COEFF DIV 10 = 0) THEN
      BEGIN
      DISPOSE (Q) ;
      GOTO 100;
      END;
    H := 1;
    FOR I:=1 TO (DIM-K) DO H := H*10;
    EXP := P@.EXP - H;
    A := P@.A;
    PTR := NIL;
    END;
  IF DHEAD = NIL THEN DHEAD := Q
                 ELSE Q1@.PTR := Q;
  Q1 :=Q;
100:  P:=P@.PTR;
  END;
END;   (*  OF DERIV  *)
PROCEDURE MULTPOLYS(VAR HEADF,HEADT:POINTER;VAR PHEAD:POINTER;MAX
                        :INTEGER);
  LABEL  110;
  VAR    P,Q,R,R1,P1       : POINTER;
         H,H1,H2,SUM,G1,G2,I,J,K  : INTEGER;
BEGIN
P := HEADF;P1:=HEADF;
R := PHEAD;
Q := HEADT;
IF R<>NIL THEN WHILE R@.PTR <>NIL DO R:= R@.PTR;
R1 := R;
WHILE (P<>NIL) DO
  BEGIN
  Q := HEADT;
  WHILE Q<>NIL DO
    BEGIN
    H := 0; SUM := 0; G1 := P@.EXP; G2 := Q@.EXP;
    FOR I := 1 TO DIM DO
      BEGIN
      H1 := G1 MOD 10: G1 := G1 DIV 10:
```

```
        H2 := G2 MOD 10; G2 := G2 DIV 10;
        H1 := H1 + H2;
        SUM := SUM + H1;
        IF SUM > MAX THEN GOTO 110;
        J := 1;
        FOR K := 1 TO (I-1) DO J := J*10;
        H := H + (H1 * J);
        END;
      NEW(R);
      WITH R@ DO
        BEGIN
        A := Q@.A;
        CASE (P@.COEFF MOD 10) OF
          0:  B1 := (P@.COEFF DIV 10)/SD;
          1:  B1 := -(P@.COEFF DIV 10)/SD;
        END;
        CASE (Q@.COEFF MOD 10) OF
          0:  B2 := (Q@.COEFF DIV 10)/SD;
          1:  B2 := -(Q@.COEFF DIV 10)/SD;
        END;
        B := B1 * B2;
        IF (B < 0)
          THEN COEFF := ROUND(ABS(B*SD))*10 + 1
          ELSE COEFF := ROUND(B*SD)*10;
        EXP := H;
        PTR := NIL;
        END;
      IF (R@.COEFF DIV 10 = 0) THEN
        BEGIN
        DISPOSE (R);
        GOTO 110;
        END;
      IF PHEAD = NIL THEN PHEAD := R
                     ELSE R1@.PTR := R;
      R1 := R;
      110:  Q := Q@.PTR;
      END;
    P1:=P;
    P := P@.PTR;
    END;
  END;     (*  OF MULTPOLYS  *)
  PROCEDURE SORT (VAR HEAD:POINTER;KIND:BOOLEAN);
    VAR   R,F           : ARRAY [0..9] OF POINTER;
          P,P1,PREV     : POINTER;
          I,J,K,L,L1 : INTEGER; (*  KIND=TRUE IF SORTING ON X'S  *)
                                  (*  KIND=FALSE IF SORTING ON A'S  \
      \*)
  BEGIN
  FOR J := 1 TO DIM DO
    BEGIN
    FOR I := 0 TO 9 DO   (*  INITIALIZE POCKETS  *)
```

```
      BEGIN
      F[I] := NIL; R[I] := NIL;
      END;
    P := HEAD;
    WHILE P<>NIL DO
      BEGIN
      L := 1;
      FOR L1 := 1 TO J-1 DO L := L * 10;
      IF KIND THEN K := (P@.EXP DIV (L)) MOD 10
              ELSE
                BEGIN
                K := (TRUNC (P@.A * TENDIM*10) DIV (L)) MOD 10;
                END;
      P1 := P@.PTR;
      IF R[K] = NIL
        THEN
          BEGIN
          R[K] := P;
          F[K] := P;
          END
        ELSE
          BEGIN
          R[K]@.PTR := P;
          R[K] := P;
          END;
      P@.PTR := NIL;
      P := P1;
      END;
    K := 0;  (*  CONCATENATE THE POCKETS  *)
           (*  FIND FIRST NON-EMPTY POCKET  *)
    WHILE( F[K] = NIL) AND (K<9) DO K:=K+1;
    HEAD := F[K];
    IF K<9
      THEN
        FOR I := (K+1) TO 9 DO
          BEGIN
          PREV := R[I-1];
          IF R[I] <> NIL
            THEN PREV@.PTR := F[I]
            ELSE R[I] := PREV;
          END;
    END;  (*  OF FOR J  *)
  END;              (*    OF    SORT     *)
  PROCEDURE STORETHETA (DIM,MAXPWR: INTEGER; VAR D1:DATA1FILE);
    LABEL  10;
    VAR    COUNT,EXPONENT,DIGIT,POWER,SUM,I,N,J,H,E: INTEGER;
  BEGIN
  REWRITE(D1);
  COUNT := 0; EXPONENT := 0;
  D1@ := EXPONENT; PUT (D1);
  10: WHILE COUNT<DIM DO
```

```
EXPONENT := EXPONENT + 1;
N := 1;
FOR I := 1 TO COUNT DO N := N*10;
DIGIT := (EXPONENT DIV N) MOD 10;
IF DIGIT <= MAXPWR
  THEN
    BEGIN
    IF NOT TOTALMXPWR
      THEN
        BEGIN
        POWER := EXPONENT;
        FOR J := 1 TO DIM DO
          BEGIN
          H := POWER MOD 10;
          POWER := POWER DIV 10;
          IF H > MAXPWR THEN GOTO 10;(*EXIT*)
          END;
        D1@ := EXPONENT;
        PUT(D1);
        END
      ELSE
        BEGIN
        POWER := EXPONENT;
        SUM := 0;
        FOR J:=1 TO DIM DO
          BEGIN
          H := POWER MOD 10;  POWER := POWER DIV 10;
          SUM := SUM  + H;
          END;
        IF SUM <= MAXPWR
          THEN
            BEGIN
            D1@ := EXPONENT;
            PUT (D1);
            END;
        END;
    END
  ELSE
    BEGIN
    COUNT := COUNT+1;
    EXPONENT := EXPONENT - (EXPONENT MOD (10*N));
    EXPONENT := EXPONENT + (10*N) - 1;
    END;
END;  (*  OF WHILE  *)
END;  (*  OF STORETHETA  *)
PROCEDURE SEPERATETERMS(POLYNOMIAL: POINTER);
 LABEL    25;
 VAR      Z,Z1: POINTER;
BEGIN
NOEQNS := 0;
```

```
  WHILE POLYNOMIAL <> NIL DO
    BEGIN
    NOEQNS := NOEQNS+1; T := T+1;
    EQN[T] := POLYNOMIAL;
    ROW[T] := EQN[T]@.EXP;
    IF EQN[T]@.PTR <> NIL
      THEN
        BEGIN
        Z := EQN[T]@.PTR; Z1 := EQN[T];
        WHILE EQN[T]@.EXP = Z@.EXP DO
          IF Z@.PTR <> NIL
            THEN BEGIN
                 Z1 := Z;
                 Z := Z@.PTR;
                 END
            ELSE BEGIN
                 POLYNOMIAL := NIL;
                 GOTO 25;
                 END;
        Z1@.PTR := NIL;
        POLYNOMIAL := Z;
        25:  END
      ELSE POLYNOMIAL := NIL;
      IF PEQNS THEN PRINT1(EQN[T],T);
    END;
  END; (* OF SEPERATETERMS *)
PROCEDURE CREATEMATRIX (NOROWS:INTEGER);
  VAR   I,L:  INTEGER;
        Z,Z1,POLYNOMIAL:  POINTER;
BEGIN
FOR I := (T-NOROWS+1) TO T DO   (*  SORT AND PRINT EQN[I]  *)
  BEGIN
  POLYNOMIAL := EQN[I];  (*  NOTE: CHANGE MEANING OF POLYNOMIAL  \
  \*)
  FOR L := 1 TO NOCOL DO MATRIX[L] := 0;
  WHILE POLYNOMIAL <> NIL DO
    BEGIN
    Z := POLYNOMIAL@.PTR; Z1 := POLYNOMIAL;
    WHILE Z <> NIL DO
      BEGIN
      IF Z@.A = POLYNOMIAL@.A
        THEN
          BEGIN
          CASE (POLYNOMIAL@.COEFF MOD 10) OF
            0:  B1 := (POLYNOMIAL@.COEFF DIV 10)/SD;
            1:  B1 := -(POLYNOMIAL@.COEFF DIV 10)/SD;
          END;
          CASE (Z@.COEFF MOD 10) OF
            0:  B2 := (Z@.COEFF DIV 10)/SD;
            1:  B2 := -(Z@.COEFF DIV 10)/SD;
          END;
```

```
            B := B1 + B2;
            IF (B < 0)
              THEN POLYNOMIAL@.COEFF := ROUND (ABS(B*SD))*10 + 1
              ELSE POLYNOMIAL@.COEFF := ROUND(B*SD)*10;
            Z1@.PTR := Z@.PTR;
            DISPOSE (Z) ;
            END
          ELSE Z1 := Z;
        Z := Z1@.PTR;
        END;
      L := 1;
      WHILE (POLYNOMIAL@.A <> COL[L]) DO L:=L+1;
      MATRIX [L] := POLYNOMIAL@.COEFF;
      POLYNOMIAL := POLYNOMIAL@.PTR;
      END;
    FOR L := 1 TO NOCOL DO
      BEGIN
      MATB@ := MATRIX[L];
      PUT(MATB);
      END;
    REMOVE(EQN[I]);
    END;
END;(* OF CREATEMATRIX *)

PROCEDURE SUBSTITUTE (VAR HEAD1:  POINTER; HEAD2:  POINTER);
  LABEL     70,100;
  VAR       P,P1,Z,HEAD3,Q,Q1,P2,P3:  POINTER;
BEGIN
IF HEAD2 = NIL THEN
  BEGIN
  WRITELN('MANIFOLD INVALID');
  GOTO 100;  (*EXIT*);
  END;
IF HEAD1 = NIL THEN
  BEGIN
  WRITELN('EMPTY POLYNOMIAL');
  GOTO 100; (*EXIT*)
  END;
P1 := HEAD1;
WHILE P1@.EXP <> HEAD2@.EXP DO
  BEGIN
  P2 := P1;
  P1 := P1@.PTR;
  IF P1 = NIL THEN GOTO 100;  (*EXIT*)
  END;
CASE (HEAD2@.COEFF MOD 10) OF
  0:  B3 := (HEAD2@.COEFF DIV 10)/SD;
  1:  B3 := -(HEAD2@.COEFF DIV 10)/SD;
END;
Z := HEAD2@.PTR;
HEAD3 := NIL;
```

```
WHILE Z <> NIL DO
  BEGIN
  P := P1;
  WHILE P@.EXP = HEAD2@.EXP DO
    BEGIN
    NEW (Q);
    CASE (P@.COEFF MOD 10) OF
      0:  B1 := (P@.COEFF DIV 10)/SD;
      1:  B1 := -(P@.COEFF DIV 10)/SD;
    END;
    CASE (Z@.COEFF MOD 10) OF
      0:  B2 := (Z@.COEFF DIV 10)/SD;
      1:  B2 := -(Z@.COEFF DIV 10)/SD;
    END;
    B := -B1*B2/B3;
    IF (B < 0)
      THEN Q@.COEFF := ROUND(ABS(B*SD))*10 + 1
      ELSE Q@.COEFF := ROUND(B*SD)*10;
    WITH Q@ DO
      BEGIN
      A := P@.A;
      EXP := Z@.EXP;
      PTR := NIL;
      END;
    IF HEAD3 = NIL THEN HEAD3 := Q
                   ELSE Q1@.PTR := Q;
    Q1 := Q;
    P3 := P;
    P := P@.PTR;
    IF P = NIL THEN GOTO 70;
    END;
  70:  Z := Z@.PTR;
  END;
P2@.PTR := HEAD3;
Q@.PTR := P3@.PTR;
P3@.PTR := NIL;
IF HEAD1 = P1 THEN HEAD1 := HEAD3;
REMOVE (P1);
100:  END;  (*          OF SUBSTITUTE        *)
PROCEDURE TRUNKMARK(HEAD:  POINTER; I,M: INTEGER;
          VAR Z: ARRAY[1..100] OF POINTER; VAR N: INTEGER);
  LABEL 100;
  VAR   SUM,G,H,K  : INTEGER;
        P,P1          : POINTER;
BEGIN
P := HEAD; P1 := HEAD; N := 0;
WHILE P <> NIL DO
  BEGIN
  G := P@.EXP;
  SUM := 0;
  FOR K := 1 TO DIM DO
```

```
      BEGIN
      H := G MOD 10;
      G := G DIV 10;
      SUM := SUM + H;
      END;
    IF SUM > M THEN
      BEGIN
      IF HEAD = P THEN REMOVE (HEAD)
                   ELSE
                     BEGIN
                     REMOVE (P);
                     P1@.PTR := NIL;
                     END;
      GOTO 100;
      END;
    N := N + 1;
    Z[N] := P;
    FOR K := 1 TO WTERMS[I] DO
      BEGIN
      P1 := P;
      P := P@.PTR;
      END;
    P1@.PTR := NIL;
    END;
100:  END;    (*               OF TRUNKMARK            *)
 (*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*)
 BEGIN     (*!!!!!!!!!!MAINLINE!!!!!!!!!!!!!!!!*)
TENDIM := 1;  SD := 1;
WRITELN('WHAT IS THE DIMENSION OF THE PROBLEM?');
READ(DIM);
FOR I := 1 TO DIM DO TENDIM := TENDIM * 10;
FOR I:= 1 TO SIGDIG DO SD := SD*10;
SD1 := SD*10;
WRITELN ('DO YOU WISH TO CHANGE YOUR THETA EXPANSION?');
WRITELN ('ANSWER Y OR N FOR YES OR NO RESPECTIVELY.');
REPLY := ' ';
WHILE REPLY=' ' DO READ(REPLY);
IF (REPLY='Y') THEN
  BEGIN
  WRITELN('DO YOU WANT TO ENTER YOUR OWN THETA EXPANSION?');
  REPLY := ' '; WHILE REPLY = ' ' DO READ(REPLY);
  IF REPLY = 'Y'
    THEN
      BEGIN
      WRITELN('ENTER THE THETA EXPANSION.  TERMINATE WITH');
      WRITELN('A NEGATIVE VALUE.');
      REWRITE(D1);READ(D1@);
      WHILE  D1@ >= 0 DO
        BEGIN
        PUT(D1);
        READ(D1@);
```

```
          END;
        END
      ELSE
        BEGIN
        TOTALMXPWR := FALSE;
        WRITELN('DO YOU WANT A TOTAL MAXPWR TRUNCATION?  IF NOT,');
        WRITELN('WE WILL TRUNCATE AT A MAXIMUM POWER OF EACH');
        WRITELN('VARIABLE.');
        REPLY := ' ';WHILE REPLY = ' ' DO READ (REPLY);
        IF REPLY = 'Y' THEN TOTALMXPWR := TRUE;
        WRITELN ('WHAT IS THE TOTAL MAXIMUM POWER OF YOUR EXPANSION\
     \?');
        READ (TTLMAX);
        STORETHETA(DIM,TTLMAX,D1);
        END;
    END;
RESET (D1);
TTERMS := 0;
WHILE NOT EOF (D1) DO
  BEGIN
  TTERMS := TTERMS + 1;
  E := D1@; GET (D1);
  WRITE (E);
  END;
WRITELN;
WRITELN('DO YOU WISH TO CHANGE YOUR FUNCTIONS F[I]?');
REPLY := ' '; WHILE REPLY = ' ' DO READ(REPLY);
IF REPLY = 'Y' THEN
  BEGIN
  WRITELN   ('ENTER COEFFICIENT AND EXPONENTS OF EACH TERM OF F[I\
   \] ');
  WRITELN   ('AS FOLLOWS:');
  WRITELN   ('          COEFF              EXPONENT');
  WRITELN;
  WRITELN   ('ENTER THE EXPONENT AS AN INTEGER, EACH DIGIT\
   \ REPRESENT-');
  WRITELN   ('ING THE POWER OF AN X-VARIABLE');
  WRITELN   ('TERMINATE WITH A COEFF-FIELD OF 5E10 AND NEGATIVE\
   \ EXP');
  REWRITE   (D2); REWRITE (D3);
  FOR   I:= 1 TO DIM DO
    BEGIN
    WRITELN ('ENTER TERMS OF FUNCTION F[',I:2,']:');
    C := 0.0; E := 0;
    REPEAT
      READ (C); READ (E);
      D2@ := C; PUT (D2);
      D3@ := E; PUT (D3);
    UNTIL (E<0);
    END;
  END;
```

```
RESET (D2); RESET (D3);
WRITELN ('ECHO DATA:');
FOR I := 1 TO DIM DO
  BEGIN
  WRITELN ('THE FUNCTION F [',I:2,'] IS:');
  WHILE (D2@<5E10) AND (D3@>=0) DO
    BEGIN
    C := D2@; GET (D2);
    E := D3@; GET (D3);
    WRITELN (C,E);
    END;
  GET (D2); GET (D3);
  END;
MANIFOLD := FALSE;
WRITELN('DO YOU WANT TO IMPOSE A MANIFOLD RESTRICTION?');
REPLY := ' ';
WHILE REPLY=' ' DO READ(REPLY);
WRITELN('REPLY IS:',REPLY);
IF (REPLY = 'Y') THEN MANIFOLD := TRUE;
IF MANIFOLD = TRUE THEN
  BEGIN
  WRITELN('HOW MANY MANIFOLD RESTRICTIONS DO YOU WANT?');
  READ(NW);
  FOR I := 1 TO NW DO
    BEGIN
    WRITELN('WHAT IS THE MAXPWR OF W[',I:2,']?');
    READ(MAXUW[I]);
    COND[I]   := FALSE;
    WRITELN;WRITELN('DO YOU WANT UW=0 WITH CONDITION W[',I:2,']=0\
   \?');
    REPLY   := ' ';WHILE REPLY=' ' DO READ(REPLY);
    WRITELN('REPLY   IS:',REPLY);
    IF   REPLY = 'Y' THEN COND[I] := TRUE;
    WRITELN ('ENTER THE EXPRESSION FOR THE MANIFOLD W[',I:2,']');
    WRITELN('WITH THE HIGHEST MAXPWR TERM FIRST.');
    W[I]:=NIL;Z1:=NIL;Z:=NIL;WTERMS[I]:=0;
    READ (C,E);
    WHILE (C < 5E10) AND (E >= 0) DO
      BEGIN
      WTERMS[I] := WTERMS[I] + 1;
      NEW(Z);
      IF (C < 0)
        THEN Z@.COEFF := ROUND(ABS(C*SD))*10 + 1
        ELSE Z@.COEFF := ROUND(C*SD)*10;
      Z@.EXP := E; Z@.PTR := NIL;
      IF W[I]=NIL THEN W[I] := Z
                 ELSE Z1@.PTR := Z;
      Z1 := Z; READ(C,E);
      END;
    WRITELN ('ECHO DATA');
    WRITELN;WRITELN('THE ENTERED W[',I:2,'] IS:');
```

```
    PRINT1(W[I],-1);
    END;
  END;
(*!!!!!!!!!!!!!!!!!!!SOLVE FOR DETERMINING EQUATIONS!!!!!!!!!!!!!!!\
    \!!!!*);
WRITELN ('WHAT TOTAL MAXIMUM POWER DO YOU WISH TO TRUNCATE ON?');
READ (MAXPWR);
FOR I := 1 TO NW DO
  MAXUW[I] := MAXUW[I] + MAXPWR - 1;
WRITELN ('DO YOU WANT THE EQUATIONS PRINTED OUT?');
REPLY := ' ';
WHILE REPLY=' ' DO READ (REPLY);
WRITELN('REPLY IS:',REPLY);
PEQNS := FALSE;
IF REPLY = 'Y' THEN PEQNS := TRUE;
NOCOL:=0;              (*        INITIALIZE COLUMN HEADS          *)
FOR I:=1 TO DIM DO
  BEGIN
  RESET(D1);
  WHILE NOT EOF(D1) DO
    BEGIN
    NOCOL:=NOCOL+1;
    COL[NOCOL] := I+(D1@/TENDIM);
    GET (D1);
    END;
  END;
T := 0;
FOR I := 1 TO NW DO
  BEGIN
  UW[I] := NIL; W1[I] := NIL;
  END;
REWRITE(MATB);
FOR I := 1 TO DIM DO
  BEGIN
  WRITELN;WRITELN;WRITELN;
  WRITELN ('DET. EQNS FOR I =',I:2,' ARE :');
  RESET (D2);RESET(D3);
  THEAD[I] := NIL;
  CREATETHETA (THEAD[I],I);
  PHEAD := NIL;
  FOR K:= 1 TO DIM DO
    BEGIN
    TDHEAD[I,K] := NIL;
    DERIV (THEAD[I],TDHEAD[I,K],K);
    FHEAD[K] := NIL;
    CREATEF (FHEAD[K],K );
    MULTPOLYS (FHEAD[K],TDHEAD[I,K],PHEAD,MAXPWR);
    REMOVE(FHEAD[K]);
    REMOVE (TDHEAD[I,K]);
    END;
  IF MANIFOLD THEN
```

```
      FOR J := 1 TO NW DO
        BEGIN
        DERIV(W[J],DW[J,I],I);
        MULTPOLYS(DW[J,I],THEAD[I],UW[J],MAXUW[J]);
        REMOVE(DW[J,I]);
        END;
    REMOVE (THEAD[I]);
    FHEAD[I] := NIL;
    CREATEF (FHEAD[I],I);
    NEGATE (FHEAD[I]);
    FOR K := 1 TO DIM DO
      BEGIN
      DERIV (FHEAD[I],FDHEAD[I,K],K);
      THEAD[K] := NIL;
      CREATETHETA (THEAD[K],K);
      MULTPOLYS (FDHEAD[I,K],THEAD[K],PHEAD,MAXPWR);
      REMOVE(FDHEAD[I,K]);
      REMOVE (THEAD[K]);
      END;
    REMOVE (FHEAD[I]);
    SORT (PHEAD,TRUE);      (*  SORT ON X'S  *)
    CREATETHETA(THEAD[0],0);
    FOR J := 1 TO NW DO
      IF COND[J] THEN
        BEGIN
        MULTPOLYS(THEAD[0],W[J],W1[J],100);
        TRUNKMARK(W1[J],J,MAXPWR,REQ,L);
        FOR K := 1 TO L DO
          BEGIN
          SUBSTITUTE(PHEAD,REQ[K]);
          REMOVE(REQ[K]);
          END;
        W1[J] := NIL;
        SORT(PHEAD,TRUE);
        END;
    SEPERATETERMS (PHEAD);
    CREATEMATRIX (NOEQNS);
    END;
  IF MANIFOLD = TRUE THEN
    FOR J := 1 TO NW DO
      BEGIN
      SORT (UW[J],TRUE);
      WRITELN;WRITELN;WRITELN;
      WRITELN('THE CONDITION FOR W[',J:2,']=0 IS:');
      IF COND[J] THEN
        BEGIN
        MULTPOLYS(THEAD[0],W[J],W1[J],100);
        REMOVE(W[J]);
        TRUNKMARK(W1[J],J,MAXUW[J],REQ,L);
        FOR K := 1 TO L DO
          BEGIN
```

```
          SUBSTITUTE(UW[J],REQ[K]);
          REMOVE(REQ[K]);
          END;
        W1[J] := NIL;
        SORT(UW[J],TRUE);
        END;
      SEPERATETERMS(UW[J]);
      CREATEMATRIX(NOEQNS);
      END;
REMOVE(THEAD[0]);
RESET(MATB);
WRITELN('DO YOU WANT TO SEE THE MATRIX?');
REPLY := ' ';
WHILE REPLY = ' ' DO READ (REPLY);
IF (REPLY='Y') THEN
  BEGIN
  WRITELN('THE INPUT MATRIX WAS:');
  RESET(MATB);
  FOR I := 1 TO NOCOL DO WRITE (COL[I]:5);
  WRITELN;
  FOR I := 1 TO NOCOL DO WRITE('*****');WRITELN;
  FOR I := 1 TO T DO
    BEGIN
    FOR J := 1 TO NOCOL DO
      BEGIN
      CASE (MATB@ MOD 10) OF
        0:  B := MATB@ /SD1;
        1:  B := (-MATB@ DIV 10) /SD;
      END;
      WRITE(B:5);
      GET (MATB);
      END;
    WRITELN;
    END;
  WRITELN;
  END;
K := T* NOCOL;
WRITELN('THE SIZE OF THE MATRIX IS:',T:4,' X',NOCOL:4,'=',K);
END.
#
```

Program MFGRB

```
#FILE (A015006)MFGRDETEQNS7B ON SWAT
(*$U-*)
PROGRAM CHECKMFGR(INPUT,OUTPUT);
   LABEL   500;
   CONST   MROWS  =  89; MCOLS = 140;
           SIGDIG = 2;  (*  # OF SIGNIFICANT DIGITS  *)
           NODIGITS = 5000000;
   TYPE    REALFILE = FILE OF REAL;
           INTEGERFILE = FILE OF INTEGER;
           COEFFICIENTS = 0..NODIGITS;
   VAR     MATB : INTEGERFILE;
           GENB2:  REALFILE;
           D1:  INTEGERFILE;
           RVAR:  REALFILE;
           IVAR:  INTEGERFILE;
           ROWVECTOR:  INTEGERFILE;
           COLVECTOR:  INTEGERFILE;
           COL:  ARRAY[1..MCOLS] OF REAL;
           REPLY   : CHAR;
           DONE  : BOOLEAN;
           T,T1,N,Q,COUNT,E,MAXPWR,TTLMAX,TENDIM,NOEQNS:  INTEGER;
           H,ROUNDS,NOCOL,IRANK,DIM  :  INTEGER;
           MATRIX:  PACKED ARRAY[1..MROWS,1..MCOLS] OF\
   \ COEFFICIENTS;
           IROW:  ARRAY[1..MROWS] OF INTEGER;
           ICOL:  ARRAY[1..MCOLS] OF INTEGER;
           SD,SD1,EPS,B,B1,B2,HOLD,TOL:  REAL;
           I,II,J,JJ,K,KK,L,LL,IR,IC,NM,MM,NCOL,P:  INTEGER;
           PIV,SAVE:  INTEGER;
PROCEDURE MFGR(VAR A:PACKED ARRAY[1..MROWS,1..MCOLS] OF
   COEFFICIENTS;
                 M:  INTEGER);
  LABEL  5,10,25,100;
BEGIN
IF (H >= 1) AND (H <= NOCOL) THEN GOTO 5;
H := 1;
(*         TEST OF SPECIFIED DIMENSIONS          *)
IF (M<=0) OR (NOCOL<=0) THEN
  BEGIN    (*    RETURN IN CASE OF FORMAL ERRORS    *)
  IRANK := -1;
  WRITELN('ERROR IN MATRIX DIMENSIONS!!!');
  GOTO 100;
  END;


(*          INITIALIZE COLUMN INDEX VECTOR           *)
 (*          SEARCH FIRST PIVOT ELEMENT           *)
IRANK := 0;
PIV := 0;
```

```
JJ := 0;
FOR J := 1 TO NOCOL DO
  BEGIN
  ICOL[J] := J;
  FOR I := 1 TO M DO
    BEGIN
    HOLD := A[I,J];
    IF (PIV DIV 10) < (TRUNC(HOLD/10)) THEN
      BEGIN
      PIV := TRUNC(HOLD);
      IR := I;
      IC := J;
      END;
    END;
  END;
(*            INITIALIZE ROW INDEX VECTOR*)
FOR I := 1 TO M DO IROW[I] := I;
(*            SET UP INTERNAL TOLERANCE*)
TOL := EPS*(PIV DIV 10);
(*            INITIALIZE ELIMINATION LOOP*)
 5: FOR NCOL := M TO NOCOL DO
  BEGIN(*      TEST FOR FEASIBILITY OF PIVOT ELEMENT        *)
  IF ROUNDS <= (NCOL-M) THEN GOTO 100;
  IF(PIV DIV 10) <= TOL
    THEN GOTO 10
    ELSE
      BEGIN
      IRANK := IRANK +1;  (*      UPDATE RANK*)
      IF (IR>IRANK) THEN         (*   INTERCHANGE ROWS IF\
   \ NECESSARY  *)
        BEGIN
        FOR J := 1 TO NOCOL DO
          BEGIN
          SAVE := A[IRANK,J];
          A[IRANK,J] := A[IR,J];
          A[IR,J] := SAVE;
          END;
(*                 UPDATE ROW INDEX VECTOR              *)
        JJ := IROW[IR];
        IROW[IR] := IROW[IRANK];
        IROW[IRANK] := JJ;
        END;
(*                INTERCHANGE COLUMNS IF NECESSARY*)
      IF (IC>IRANK) THEN
        BEGIN
        FOR J := 1 TO M DO
          BEGIN
          SAVE := A[J,IRANK];
```

```
            A[J,IRANK] := A[J,IC];
            A[J,IC] := SAVE;
            END;
(*                      UPDATE COLUMN INDEX VECTOR                  *)
          JJ := ICOL[IC];
          ICOL[IC] := ICOL[IRANK];
          ICOL[IRANK] := JJ;
          END;
        KK := IRANK + 1;


        IF (IRANK >= M) THEN GOTO 25; (*  TEST FOR LAST ROW  *)


         (* TRANSFORM CURRENT SUBMATRIX AND SEARCH NEXT PIVOT *)

        SAVE := PIV;
        PIV := 0;
        FOR J := KK TO M DO
          BEGIN
          CASE (A[J,IRANK] MOD 10) OF
            0:  B1 := A[J,IRANK]/SD1;
            1:  B1 := (-A[J,IRANK] DIV 10)/SD;
          END;
          CASE (SAVE MOD 10) OF
            0:  B2 := SAVE /SD1;
            1:  B2 := (-SAVE DIV 10) /SD;
          END;
          HOLD := B1/B2;
          IF (HOLD < 0)
            THEN A[J,IRANK] := ROUND(ABS(HOLD*SD))*10 + 1
            ELSE A[J,IRANK] := ROUND(HOLD*SD)*10;

          (* TEST FOR LAST COLUMN *)
          IF (IRANK<NOCOL) THEN
            BEGIN
            FOR I := KK TO NOCOL DO
              BEGIN
              CASE (A[IRANK,I] MOD 10) OF
                0:  B1 := A[IRANK,I]/SD1;
                1:  B1 := (-A[IRANK,I] DIV 10) /SD;
              END;
              IF (A[J,I] MOD 10) > 1 THEN WRITELN(A[J,I]);
              CASE (A[J,I] MOD 10) OF
                0:  B2 := A[J,I] /SD1;
                1:  B2 := (-A[J,I] DIV 10) /SD;
              END;
              B := B2 - (HOLD*B1);
              IF (B < 0)
                THEN A[J,I] := ROUND(ABS(B*SD))*10 + 1
                ELSE A[J,I] := ROUND(B*SD)*10;
```

```
                IF (A[J,I] DIV 10) > (PIV DIV 10) THEN
                  BEGIN
                  PIV := A[J,I];
                  IR := J;
                  IC := I;
                  END;
                END;
              END;
            END;
          END;
    END;
(*          SET UP MATRIX EXPRESSING ROW DEPENDENCIES               \
      \*)
10:  DONE := TRUE;
IF (IRANK<1) THEN
  BEGIN
  WRITELN('FORMAL ERROR :   RANK<1');
  GOTO 100; (*  EXIT  *)
  END;
IF (IRANK = 1) THEN GOTO 25;
IR := IRANK;
FOR J := 2 TO IRANK DO
  BEGIN
  IR := IR - 1;
  JJ := IRANK;
  FOR I := KK TO M DO
    BEGIN
    HOLD := 0.0;
    JJ := JJ + 1;
    MM := IRANK;
    IC := IRANK;
    FOR L := 1 TO (J-1) DO
      BEGIN
      CASE(A[JJ,MM] MOD 10) OF
        0:  B1 := A[JJ,MM] /SD1;
        1:  B1 := (-A[JJ,MM] DIV 10) /SD;
      END;
      CASE (A[IC,IR] MOD 10) OF
        0:  B2 := A[IC,IR] /SD1;
        1:  B2 := (-A[IC,IR] DIV 10) /SD;
      END;
      HOLD := HOLD + (B1*B2);
      IC := IC - 1;
      MM := MM - 1;
      END;
    CASE (A[JJ,MM] MOD 10) OF
      0:  B1 := A[JJ,MM] /SD1;
      1:  B1 := (-A[JJ,MM] DIV 10) /SD;
    END;
    B := B1 - HOLD;
    IF (B<0)
```

```
        THEN A[JJ,MM] := ROUND(ABS(B*SD))*10 + 1
        ELSE A[JJ,MM] := ROUND(B*SD)*10;
      END;
    END;
(*                 TEST FOR COLUMN REGULARITY*)
25  :  IF (NOCOL-IRANK) <= 0 THEN
    BEGIN
    WRITELN('ERROR!! COLUMN IRREGULARITY.');
    GOTO 100;(*  EXIT  *)
    END;
DONE := TRUE;

(* SET UP MATRIX EXPRESSING BASIC VARIABLES IN TERMS OF FREE *)
 (*           PARAMETERS (HOMOGENEOUS SOLUTION).           *)
IR := IRANK;
FOR J := 1 TO IRANK DO
    BEGIN
    FOR I := (IRANK + 1) TO NOCOL DO
      BEGIN
      JJ := IRANK;
      LL := IRANK;
      HOLD := 0.0;
      FOR II := (J - 1) DOWNTO 1 DO
        BEGIN
        CASE (A[IR,JJ] MOD 10) OF
          0:  B1 := A[IR,JJ] /SD1;
          1:  B1 := (-A[IR,JJ] DIV 10) /SD;
        END;
        CASE (A[LL,I] MOD 10) OF
          0:  B2 := A[LL,I] /SD1;
          1:  B2 := (-A[LL,I] DIV 10) /SD;
        END;
        HOLD := HOLD - (B1*B2);
        JJ := JJ - 1;
        LL := LL - 1;
        END;
      CASE (A[LL,I] MOD 10) OF
        0:  B1 := A[LL,I] /SD1;
        1:  B1 := (-A[LL,I] DIV 10) /SD;
      END;
      CASE (A[IR,JJ] MOD 10) OF
        0:  B2 := A[IR,JJ] /SD1;
        1:  B2 := (-A[IR,JJ] DIV 10) /SD;
      END;
      B := (HOLD - B1)/B2;
      IF (B<0)
        THEN A[LL,I] := ROUND(ABS(B*SD))*10 + 1
        ELSE A[LL,I] := ROUND(B*SD)*10;
      END;
    IR := IR - 1;
```

```
 100:   END;  (*       OF MFGR        *)
(*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!*)
BEGIN
SD := 1; TENDIM := 1;DONE := FALSE;
WRITELN('WHAT IS THE DIMENSION OF THE PROBLEM?');
READ(DIM);
FOR N := 1 TO DIM DO TENDIM := TENDIM * 10;
FOR N := 1 TO SIGDIG DO SD := SD*10;
SD1 := SD*10; EPS := 1/SD;
WRITELN('IS THIS A NEW PROBLEM?');
READ(REPLY); WHILE REPLY = ' ' DO READ(REPLY);
IF (REPLY = 'N') THEN
  BEGIN
  RESET(IVAR);
  I  := IVAR@;  GET(IVAR);         II := IVAR@;  GET(IVAR);
  J  := IVAR@;  GET(IVAR);         JJ := IVAR@;  GET(IVAR);
  K  := IVAR@;  GET(IVAR);         KK := IVAR@;  GET(IVAR);
  L  := IVAR@;  GET(IVAR);         LL := IVAR@;  GET(IVAR);
  IR := IVAR@;  GET(IVAR);         IC := IVAR@;  GET(IVAR);
  NM := IVAR@;  GET(IVAR);         MN := IVAR@;  GET(IVAR);
  H  := IVAR@;  GET(IVAR);         P  := IVAR@;  GET(IVAR);
  NOCOL := IVAR@; GET(IVAR);       IRANK := IVAR@; GET(IVAR);
  PIV := IVAR@;  GET(IVAR);
  T := IVAR@;  GET(IVAR);          SAVE := IVAR@;  GET(IVAR);
  RESET(RVAR);
  TOL := RVAR@;  GET(RVAR);        HOLD := RVAR@;  GET(RVAR);
  RESET(ROWVECTOR);
  FOR N := 1 TO T DO
    BEGIN
    IROW[N] := ROWVECTOR@;
    GET(ROWVECTOR);
    END;
  RESET(COLVECTOR);
  FOR N := 1 TO NOCOL DO
    BEGIN
    ICOL[N] := COLVECTOR@;
    GET(COLVECTOR);
    END;
  END;
NOCOL:=0;             (*      INITIALIZE COLUMN HEADS      *)
FOR N:=1 TO DIM DO
  BEGIN
  RESET(D1);
  WHILE NOT EOF(D1) DO
    BEGIN
    NOCOL:=NOCOL+1;
    COL[NOCOL] := N+(D1@/TENDIM);
    GET (D1);
    END;
  END;
```

```
T := 0;
WHILE NOT EOF(MATB) DO
  BEGIN
  T := T + 1;
  FOR N := 1 TO NOCOL DO
    BEGIN
    MATRIX[T,N] := MATB@;
    GET(MATB);
    END;
  END;
T1 := T * NOCOL;
WRITELN('THE SIZE OF THE MATRIX IS:',T:4,'X',NOCOL:4,'=',T1);
WRITELN('HOW MANY ITERATIONS DO YOU WANT?');
READ(ROUNDS);
MFGR(MATRIX,T);
IF NOT DONE THEN
  BEGIN
  REWRITE(IVAR);
  IVAR@ :=  I;  PUT(IVAR);              IVAR@ := II;  PUT(IVAR);
  IVAR@ :=  J;  PUT(IVAR);              IVAR@ := JJ;  PUT(IVAR);
  IVAR@ :=  K;  PUT(IVAR);              IVAR@ := KK;  PUT(IVAR);
  IVAR@ :=  L;  PUT(IVAR);              IVAR@ := LL;  PUT(IVAR);
  IVAR@ := IR;  PUT(IVAR);              IVAR@ := IC;  PUT(IVAR);
  IVAR@ := NN;  PUT(IVAR);              IVAR@ := MM;  PUT(IVAR);
  IVAR@ := NCOL; PUT(IVAR);             IVAR@ :=  P;  PUT(IVAR);
  IVAR@ := NOCOL; PUT(IVAR);            IVAR@ := IRANK; PUT(IVAR);
  IVAR@ := PIV;  PUT(IVAR);
  IVAR@ := T;  PUT(IVAR);               IVAR@ := SAVE;  PUT(IVAR);
  RESET(IVAR);
  REWRITE(RVAR);
  RVAR@ := TOL;  PUT(RVAR);             RVAR@ := HOLD;  PUT(RVAR);
  RESET(RVAR);
  REWRITE(ROWVECTOR);
  FOR N := 1 TO T DO
    BEGIN
    ROWVECTOR@ := IROW[N];
    PUT(ROWVECTOR);
    END;
  RESET(ROWVECTOR);
  REWRITE(COLVECTOR);
  FOR N := 1 TO NOCOL DO
    BEGIN
    COLVECTOR@ := ICOL[N];
    PUT(COLVECTOR);
    END;
  RESET(COLVECTOR);
  REWRITE(MATB);
  FOR Q := 1 TO T DO
    FOR N := 1 TO NOCOL DO
    BEGIN
    MATB@ := MATRIX[Q,N];
```

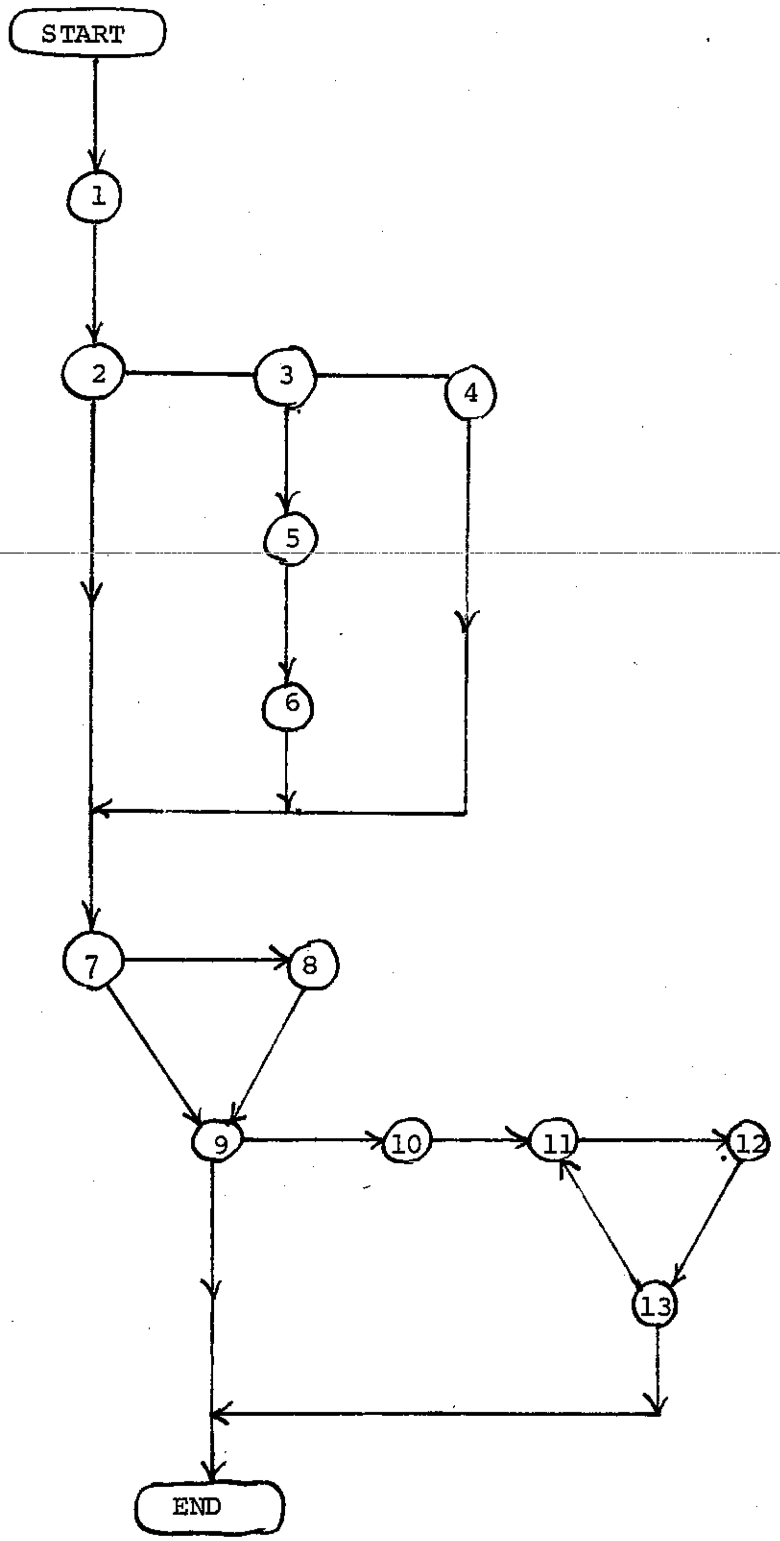
```
      PUT(MATB);
      END;
    RESET(MATB);
    GOTO 500;   (*EXIT*)
    END;
WRITELN('THE RANK IS:',IRANK);
WRITELN('THE RELATION VECTOR IS:');
FOR N := 1 TO NOCOL DO WRITE(ICOL[N]:5);
WRITELN;
WRITELN('THE RESULTING EQUATIONS ARE:');
WRITELN; WRITELN; WRITELN; WRITELN;
COUNT := 0;
REWRITE(GENB2);
FOR N := (IRANK + 1) TO NOCOL DO
    BEGIN
    IF COL[ICOL[N]] >= 1 THEN
      BEGIN
      COUNT := COUNT + 1;
      WRITE('U[',COUNT:4,'] = (A',COL[ICOL[N]]:5,') + ');
      GENB2@:= 1.0; PUT(GENB2);
      GENB2@:= COL[ICOL[N]]; PUT(GENB2);
      FOR Q := 1 TO IRANK DO
        BEGIN
        CASE (MATRIX[Q,N] MOD 10) OF
          0:  B := MATRIX[Q,N] /SD1;
          1:  B := (-MATRIX[Q,N] DIV 10) /SD;
        END;
        IF (ABS(B) > 1/SD) THEN
          BEGIN
          WRITE(B:5,'(A',COL[ICOL[Q]]:5,') + ');
          GENB2@ := B; PUT(GENB2);
          GENB2@ := COL[ICOL[Q]]; PUT(GENB2);
          END;
        END;
      WRITELN;WRITELN;
      GENB2@ := 5E10 ; PUT(GENB2);
      END;
    END;
RESET(GENB2);
WRITELN('DO YOU WANT TO SEE THE MATRIX?');
REPLY := ' ';
WHILE REPLY = ' ' DO READ (REPLY);
IF (REPLY='Y') THEN
    BEGIN
    WRITELN('THE INPUT MATRIX WAS:');
    RESET(MATB);
    FOR N := 1 TO NOCOL DO WRITE (COL[N]:5);
    WRITELN;
    FOR N := 1 TO NOCOL DO WRITE('*****');WRITELN;
    FOR Q := 1 TO T DO
      BEGIN
```

```
      FOR N := 1 TO NOCOL DO
        BEGIN
        CASE (MATB@ MOD 10) OF
          0:  B := MATB@ /SD1;
          1:  B := (-MATB@ DIV 10) /SD;
        END;
        WRITE(B:5);
        GET (MATB);
        END;
      WRITELN;
      END;
    WRITELN;
    WRITELN('THE PROCESSED MATRIX IS:');
    WRITELN;WRITELN('L*U IS:');
    FOR Q := 1 TO IRANK DO
      BEGIN
      FOR N := 1 TO IRANK DO
        BEGIN
        CASE (MATRIX[Q,N] MOD 10) OF
          0:  B := MATRIX[Q,N] /SD1;
          1:  B := (-MATRIX[Q,N] DIV 10) /SD;
        END;
        WRITE(B:5);
        END;
      WRITELN;
      END;
    WRITELN;WRITELN('C MATRIX IS:');
    FOR N := 1 TO IRANK DO WRITE (IROW[N]:5);
    WRITELN;
    FOR N := 1 TO IRANK DO WRITE('*****');WRITELN;
    FOR Q := (IRANK + 1) TO T DO
      BEGIN
      FOR N := 1 TO IRANK DO
        BEGIN
        CASE (MATRIX[Q,N] MOD 10) OF
          0:  B := MATRIX[Q,N] /SD1;
          1:  B := (-MATRIX[Q,N] DIV 10) /SD;
        END;
        WRITE(B:5);
        END;
      WRITELN;
      END;
    WRITELN('THE IROW VECTOR IS:');
    FOR N := 1 TO T DO WRITE (IROW[N]:5);
    WRITELN;WRITELN;WRITELN;
    WRITELN('THE HOMOGENOUS SOLN MATRIX IS:');
    FOR N := (IRANK + 1) TO NOCOL DO WRITE(COL[ICOL[N]]:5);
    WRITELN;
    FOR N := (IRANK + 1) TO NOCOL DO WRITE('*****'); WRITELN;
    FOR Q := 1 TO IRANK DO
      BEGIN
```

```
    BEGIN
    FOR N := (IRANK + 1) TO NOCOL DO
      BEGIN
      CASE (MATRIX[Q,N] MOD 10) OF
        0:  B := MATRIX[Q,N] /SD1;
        1:  B := (-MATRIX[Q,N] DIV 10) /SD;
      END;
      WRITE(B:5);
      END;
    WRITELN;
    END;
  END;
REWRITE(MATB);  (*  DISPOSE OF DISK SPACE  *)
REWRITE(RVAR);
REWRITE(IVAR);
REWRITE(ROWVECTOR);
REWRITE(COLVECTOR);
500: END.
#
```

Appendix II

1. What is the dimension of the problem?
2. Do you want to change your theta expansion?
3. Do you want to enter your own theta expansion?
4. Enter the theta expansion.
5. Do you want a "total maximum power" truncation? If not we will truncate on a maximum power of each variable.
6. Enter the total maximum power of the truncation.
7. Do you want to change your functions f^i?
8. Enter the functions f^i.
9. Do you want to impose a manifold restriction?
10. How many manifold restrictions are there?
11. What is the maximum power of W^i?
12. Do you want UW = 0 with the condition W = 0?
13. Enter the manifold restriction W^i with the highest power first.
14. What maximum power do you want to truncate the determining equations on?
15. Do you want the equations printed out?



Flow Chart for DETERMININGEQNS

References

1. Abraham, R., and Marsden, J.E., Foundations of Mechanics, Benjamin/Cummings Publishing Co., 1978.

2. Arnold, V.I., Ordinary Differential Equations, translated by R.A. Silverman, M.I.T. Press, Massachusetts, 1973.

3. Arnold, V.I., Mathematical Methods of Classical Mechanics, translated by K. Vogtmann and A. Weinstein, Springer-Verlag, N.Y., 1978.

4. Baber and Hasse, "The Two-Center Problem in Wave Mechanics", Cambridge Philosophical Society Proceedings 31, (1935).

5. Bates, Ledsham, and Stewart, "Wave Functions of the Hydrogen Molecular Ion", Trans. Roy. Soc., London, 246, 215-240 (1953).

6. Birkoff, G., Dynamical Systems, American Mathematical Society, Providence, R.I., 1927.

7. Bluman, G., and Cole, J.D., Similarity Methods for Differential Equations, Springer-Verlag, N.Y., 1974.

8. Buckingham, R.A., "Exactly Soluble Bound State Problems" in Quantum Theory Vol. 10-I, edited by D.R. Bates, Pure and Applied Physics, Academic Press, 1961.

9. Campbell, J.E., Introductory Treatise on Lie's Theory of Finite Continuous Transformation Groups, Chelsea Pub. Co., N.Y., 1966.

10. Cisneros and McIntosh, "Search for a Universal Symmetry Group in Two Dimensions", J. Math. Phys. 11 (3), 1970.

11. Coulson, C.A., and Joseph, A., "A Constant of the Motion for the Two-Center Kepler Problem", Int. J. of Quantum Chem. I, 337 (1967).

12. Coulson and Joseph, Proc. Phys. Soc. 90, 887 (1967).

13. Dulock and McIntosh, "On the Degeneracy of the Two-Dimensional Harmonic Oscillator", Amer. J. Phys. 33, 109-118 (1965).

14. Fock, V., Z. Physik 98, 145-154 (1935).

15. Erickson, H., and Hill, E., "A Note on the One-Electron States of Diatomic Molecules", Phys. Rev. 75, 29 (1949).

16. Gershtein, Ponomarev, and Puzynina, "A Quasiclassical Approximation in the Two-Center Problem", JETP 20, 418 (1965).

17. Gilmore, R., Lie Groups, Lie Algebras, and Some of Their Applications, John Wiley and Sons, New York, 1974.

18. Goldstein, H., Classical Mechanics, Addison-Wesley Publishing Co., Massachusetts, 1950.

19. Horowitz, E., and Sahni, S., Fundamentals of Data Structures, Computer Science Press, Potomac, Maryland, 1976.

20. Jaffe, G., Z. Physik 87, 535 (1934).

21. Jauch and Hill, "On the Problem of Degeneracy in Quantum Mechanics", Phys. Rev. 57, (1940).

22. Judd, B.R., Angular Momentum Theory of Diatomic Molecules, Academic Press, N.Y., 1975.

23. McIntosh, H.V., "Symmetry and Degeneracy" in Group Theory and Its Applications Vol. II, Academic Press, 1971.

24. Pauli, Ann. Phys. (Leipzig) 68, 177 (1922).

25. Schneider, Weingart, and Perlman, An Introduction to Programming and Problem Solving with Pascal, John Wiley and Sons, 1978.

26. Smale, S., and Hirsh, W., Differential Equations, Dynamical Systems, and Linear Algebra, Academic Press, N.Y., 1974.

27. Strand, M., and Reinhardt, W., "Semiclassical Quantization of the Low Lying Electronic States of H_2^+", J. Chem. Phys. 70, 8 (1979).

28. Sumi, T., On Canonical Transformations Admitted by Some Kepler Problems, M.S. Thesis, University of the Pacific, Ca., 1977.

29. Thirring, W., A Course In Mathematical Physics Vol. I, translated by Evans M. Harrell, Springer-Verlag, N.Y., 1978.

30. Wilson, A.H., Proc. Roy. Soc., A 118, 617 (1928).

31. Wulfman, C.E., J. Phys. A : Math, Gen., Vol. 12, No. 4, 1979.

32. Wulfman, C.E., Local Symmetry - An Introduction to the Study of Invariance Properties of Physical Systems Governed by Differential Equations (not published).

33. Wulfman, C. and Sumi, T., "New, Locally Stable, Symmetries of Keplerian Systems", Proc. of a conference at University of Western Ontario, June 7-9, 1978.

34. Wybourne, B.G., Classical Groups for Physicists, John Wiley, N.Y., 1974.