THE THIRD FACE OF COMPUTER

"COMPUTER SOLUTION OF SYMBOLIC PROBLEMS"

M.I.WANAS*

ABSTRACT

The purpose of the present paper is to bring to the attention of scientists and engineers the existence of systems capable of performing algebraic(non-numerical) and symbolic computations. These systems have been in use successfully in a number of fields such as: Relativity, Celestial Mechanics, Aerodynamics, Fluid Mechanics, Quantum Electrodynamics High Energy Physics, Astrophysics, and Cosmology.
A brief review and comparison of several systems is given. The use of algebraic computing to solve diverse problems in scientific research is shown. To illustrate the use of such systems, an example of general interest to potential users is being discussed. That is the use of the algebraic manipulation language REDUCE 2 to write a program for coordinate transformations. Input, output, and the structure of the program are being discussed. A copy of the program together with a sample of its results are presented.

* Associate Professor,Dpt.of Astronomy and Meteorology, Faculty of Science, Cairo University, Guiza, Egypt.

INTRODUCTION

In the last two decades a large number of computing systems has been constructed to carry out algebraic and symbolic manipulation. The number of these systems is about to be of the same order as the number of users. This may be due to the common belief that the computer has only two faces: one for numerical computations, and the other for data processing. Most of the users do not know that there are many programs and languages ready to manipulate non-numerical data. This represents the third face of computer.

Languages which one can program in are being classified into the following 4-categories [1]. The first is the Machine Code which varies from one computer to another. It consists of instructions to the machine in a number-coded form. It is very difficult to program using this type of languages. The second is the Assembly Language (e.g.FAB,MAP,COMPASS,...)which is similar to the machine code but less difficult to program in. This type of languages is also machine dependent. The third is the High-level Languages (e.g.FORTRAN and ALGOL for numerical calculations; LISP, REDUCE, SHEEP, and MACSYMA for algebraic manipulation) which enable the programmer to write in an english-like or mathematical-like notation. These languages possess compilers to translate into the machine code. Many of these languages are, in principle, machine independent. But not all high-level languages are available on all computers. The fourth is the Low-level Languages (e.g.CAMAL) which form an intermediate level between high-level and assembly languages.

For the algebraic manipulation, the computer will be of great help in the cases of calculations which are tedious and some times impossible to be carried out by hand. Computers are superior to hand calculations in these areas because they are faster, error free, and the machine will not get tired. Moreover, round off is not a problem since the results may be obtained in an exact form. Also the stability and convergence problems of numerical computations are not relevant [2].

In the following section a brief comparison of some of the algebraic manipulation systems is given. Fields in which algebraic manipulation are being used successfully are reviewed in the next section. An example to illustrate the use of one of these systems-REDUCE- is discussed in a separate section. A final conclusion is given in the last section.

COMPARISON BETWEEN SOME ALGEBRAIC SYSTEMS

Among systems used for algebraic manipulation, there are few which are applicable for general purposes (e.g.REDUCE, and MACSYMA). The majority are merely programs written to solve certain particular problems (e.g.LAM, SHEEP, ORTOCARTAN). The more general the capability of the system the less efficient (the slower) the system is. For example [1] the CPU-time

Table 1 Comparison Between Algebraic Systems

Language	Computer	Implimentation Lang.	Syntax	Know. of Elem. Func.	Memory in K.W.	Notes	Ref.
ALTRAN	Many	FORTRAN	FORTRAN PL/I like	Non	65 on IBM 360	Matrix operations are built in.	[3]
CAMAL	IBM 360			Most	18 on TITAN		[1]
FORMAC	IBM 360/370	IBM Assembler	FORTRAN or PL/I	Most	40 on IBM 360	Not a complete programming lang.	[3]
LAM	Many	LISP	LISP	Most	38 on IBM 360		[3]
MACSYMA	DEC KL 10	LISP	ALGOL like	Many	123	Not generally available. Tensor operations could be carried out easily.	[5]
ORTOCARTAN	DEC Cyber 73	LISP 4.1	Mixed FORTRAN &LISP	Most	29.69	Some tensor operations are built in.	[4]
REDUCE	Many	LISP 1.5	ALGOL like	Many	70 on DEC KA 10	Matrix operation and γ-matrix algebra are built in.Tensor operations could be carried out easily.	[6]
SAC 1	Many	FORTRAN	FORTRAN ALGOL like	Non	33 on IBM 360	It is a collection of some subroutines.	[3]
SHEEP	DEC PDP 10	LISP 1.6	ALGOL like	Most	25	Some tensor operations are built in.	[7]
SYMBAL	CDC 6000	CDC Assembler	Improved ALGOL	Non	25		[3]

for the standard Bondi-metric (in a certain problem in general relativity) is 30 seconds using SHEEP on DEC KA 10. The corresponding time using REDUCE is 360 seconds on the same computer for solving the same problem. However, saving the user's time by the use of computers is much more important than reducing the computer's time by specifying the system used.

Some of the high-level languages possess two important facilities. The first is the "Garbage Collector" which is a device responsible of clearing the store occupied by unwanted expressions. The absence of this device in the language causes a real problem of store and will affect the efficiency of the language. The second facility is the "Interactive Mode". It is an additional convesational component which helps as a medium for a dialogue between the user and the machine. This device enables the user to interact with the machine freely when computations are being carried out. Systems possessing such facility has the advantage of mixing the experience of the user in his field and the (knowledge) built in the system. Interactive mode is ideal for running small jobs only since it is not practical to wait 10 minutes for each responce. So, for large jobs, it is preferable to use the "Batch Mode". Unfortunatly, interactive mode can be implimented only on large computers.

Table 1 gives a brief comparison of ten algebraic manipulation systems. Some of which are general purpose systems, and the others are programs written for specific problems. All systems are high-level languages except CAMAL which is a low-level one. For more details, the reader is referred to the references cited in the table.

APPLICATIONS IN DIVERSE SYMBOLIC PROBLEMS

Many mathematicians, astronomers, theoretical physicists,and engineers spend a large amount of their time carrying out routine algebraic calculations such as differentiation, integration, substitution in complicated formulae...etc. Systems such as those given in the previous section are being used successfully as tools for solving such problems. Moreover, they have been used in solving problems that were previously considered impossible to be solved by hand.

In the following, we are going to outline briefly how algebraic manipulation using computer has been introduced successfully in some important fields.

General Theory of Relativity (GR)

One of the fields in which algebraic manipulation is being used successfully is the field of GR,(a theory which studies the gravitational fields), and its applications to astronomy and cosmology. The theory is based on a certain geometrical structure (a 4-dimensional Riemannian space) defined in terms of a symmetric second order tensor-field (the metric tensor $g_{\mu\nu}$) giving rise to a symmetric affine-connexion (the Christoffel symbols $\left\{ {\alpha \atop \mu\nu} \right\}$). To solve any particular problem in this domain, we have to calculate Einstein's field equations (10-equations in general). Although the calculations is some what straight forward but it involves tedious processes of differentiation, summation, substitution, and matrix operations. several systems have been designed especially to solve such problems, e.g. the LAM-family (ALAM, ILAM, CLAM) and ORTOCARTAN. The most efficient system known to the author in this respect is SHEEP [7] which is the interactive adaptation for the DEC PDP-10 of the system LAM. For a full review in this area the reader is referred to d'Inverno's papers [1] , [8] .

Generalized and Unified Field Theories

Another field which is similar to the the previous one is that of generalized and unified field theories, (concerned with the unification of gravitational and electromagnetic fields). In this area the geometrical structure is more complicated than the Riemannian geometry used in GR. For example, in one of these theories [9] , the geometrical structure used is the absolute parallelism space. Instead of the ten field variables of GR we have here sixteen field variables to start with. Also, instead of the 40-components of the symmetric affine connexion used in Riemannian geometry, we have 64-components of a non-symmetric connexion. To manipulate the field equations of this theory and to get the type of the space used [10] , more than twinty geometrical elements (each of which has 30-components on the avarage) have to be manipulated. The author have been able to use REDUCE to manipulate these elements [11] , [12] .

Aerodynamics

The third field in which algebraic manipulation has been of great help is the field of aerodynamics. The typical problem in this field is to derive a simulator that will enable the engineer to know how an aerospace vehicle will behave under actual flight conditions [13] . This simulator is obtained by using the solution of the equations of a mathematical model representing the vehicle and its environment. This involves in general 12-equations: 3-for locating the body, 3-to determine the force, 3-for the moments, and 3-for Euler angles which determine the orientations of the body. In the aerostatic case, certain quantities necessary to formulate these equations (force F^i, moments P^i, stability derivatives $\frac{\partial F^i}{\partial A}$, $\frac{\partial F^i}{\partial B}$, $\frac{\partial P^i}{\partial A}$, $\frac{\partial P^i}{\partial B}$ where A , B are the angles of attack and sideslip respectively (see Fig.1)) are being given in the wind-tunnel stability coordinate system ($\bar{x}^i$), in which data is obtained. While the mathematical equations, describing the model, are formulated in the aircraft body coordinate system (x^j). Each of the previous quantities transform from ($\bar{x}^i$) to (x^j) as the components of contravariant vectors. So, for each quantity we have 3-transformation equations each of which has 3-terms. In the next section, an example of using computer for coordinate transformation of a contravariant vector is being discussed in details.

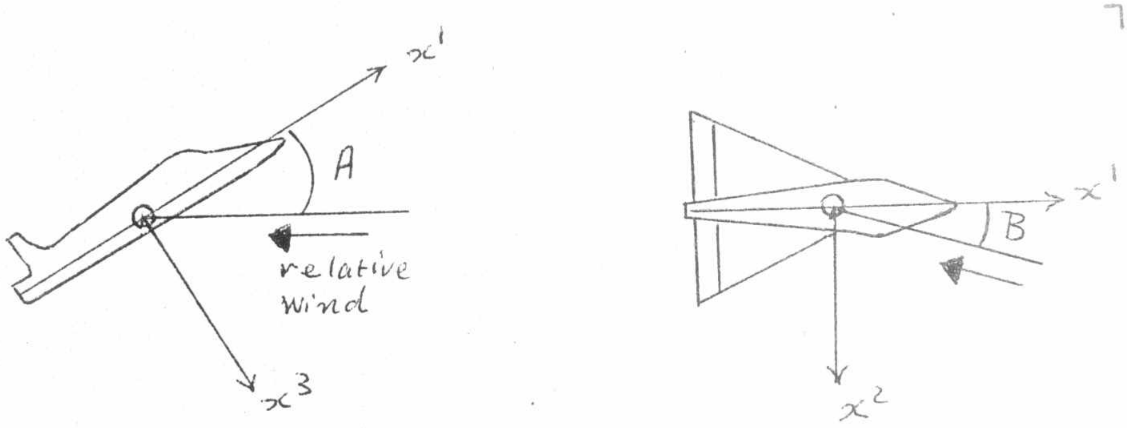

Fig.1.Angle of attack (A) and angle of sideslip (B)

In the aerodynamic case the problem will be more complicated. Each of the aerodynamic stability derivatives of the problem ($\frac{\partial F^i}{\partial V^j}$, $\frac{\partial F^i}{\partial \omega^j}$, $\frac{\partial F^i}{\partial \dot{V}^j}$, $\frac{\partial F^i}{\partial \dot{\omega}^j}$, $\frac{\partial P^i}{\partial V^j}$, $\frac{\partial P^i}{\partial \omega^j}$, $\frac{\partial P^i}{\partial \dot{V}^j}$, $\frac{\partial P^i}{\partial \dot{\omega}^j}$, where V^j, ω^j are the linear and angular velocity respectively, and $\dot{V}^j$, $\dot{\omega}^j$ are the linear and angular accelerations respectively) are being transformed as the components of a mixed tensor of the second order [14]

$$\overline{T}^{i}{}_{j} = \frac{\partial \bar{x}^{i}}{\partial x^{k}} \frac{\partial x^{\ell}}{\partial \bar{x}^{j}} T^{k}{}_{\ell} \qquad (i , j =1,2,3) \qquad (1)$$

summation convention is being carried out over repeated indices. So, for each of the previous stability derivatives there will be 9-transformation equations each of which will consist of 9-terms. So for each quantity we have to calculate 81-terms in general. The most suitable systems for carrying out such manipulation are REDUCE and MACSYMA.

Celestial Mechanics

The problem in this field is to determine the position of any celestial body (e.g.a planet, the Moon, an artificial satellite...) at any time. This needs solving the equations of motion of the body. The solution rarely exists in a closed form. One way out of this problem is to construct an analytic perturbation theory. In other words, to get the solution in the form of a series in terms of some small parameter (eccentricty or inclination of the orbit). The series appearing frequently in such calculations is the Poisson series, which is a trigonometric series with its coefficients as polynomials. Delaunay in 1860 had constructed a lunar theory up to the seventh order. His calculations were accomplished in twenty years. This work has been repeated by Deprit, Henrard and Rom [15] using algebraic computation up to the nineth order. They have found one error in the work of Delaunay. The algebraic manipulation system CAMAL was designed originally to perform such calculation. However, this type of calculations can also be done using MACSYMA or REDUCE.

Quantum Electrodynamics (QED)

This theory is concerned with the interaction of charged particles via photons in the frame of relativistic quantum mechanics. The situation here is similar to that in celestial mechanics. That is in the case of elementary particles in interaction we do not know in general the equations of motion, and where we do, we can not solve them [16] . So, we have to use some perturbation technique. The most famous technique in this field is that due to Feynman. The expansion parameter in this case is the fine structure constant (1/137). The probability of occurance of any physical event is given by

$$P = |\langle f|M|i\rangle|^2 \qquad (2)$$

where $|i\rangle$, $\langle f|$ are the initial and final states respectively, and M is the total amplitude given by

$$M = m_1^{(1)} + m_2^{(1)} + m_1^{(2)} + m_2^{(2)} + m_3^{(2)} + \ldots \qquad (3)$$

m represents the amplitude of indevidual process, $m_i^{(1)}$ represents first order diagram with two vertices, $m_i^{(2)}$ represents second order diagram with four vertices, and so on. Feynman's rules give a one-to-one mapping between diagrams and a complicated mathematical expression involving the 4x4 non-commuting γ-matrices. The calculations are straightforward but very tedious. For example to match theoretical and experimental results one has to evaluate some hundreds of Feynman diagrams which are generated within the first few orders. For a review of the use of algebraic computing in QED the reader is referred to reference [2] and references listed there in. The most suitable system for such calculations is REDUCE, since it was designed originally to solve such problems.

Fluid Mechanics

Most of the calculations in this field are somewhat similar to those of GR and cosmology. Navier-Stokes equation [17] ,which is the basis of fluid mechanics, can be written in a tensorial form. In addition to this equation, an equation of state and the equation of continuity are required to complete the system necessary to solve any isothermal problem. In order to formulate these equations, the only quantity needed is the metric tensor of the surface over which the fluid flows. From this tensor one can formulate Christoffel symbols of the second kind and hence write the previously mentioned equations. Although the equations are complicated, the only operations carried out by computer are summation and differentiation. Once a program is written for such a problem, the user has only to give the machine the metric tensor of the surface. The most suitable systems to write such programs are REDUCE and MACSYMA.

Astrophysics

The last application is in the field of astrophysics. One of the problems in this field, in which algebraic computing has been used successfully, is the problem of non-radial pulsation of stellar models. Thorne and others [18,19,20]. have used the FORMAC pakage ALBERT to solve this problem.

REDUCE AND PARTICULAR APPLICATION

The most general systems known to the author at the time of writing this paper are MACSYMA and REDUCE. Unfortunately, the MACSYMA system is available only at MIT through ARPA network. But REDUCE is available on many computers. It was developed by A.C.Hearn in 1957 to carry out calculations of interest to high energy physicists, and it has been evolved since then untill 1967 when it was announced as a system for general algebraic manipulation. This system is continuously evolving to cover the needs of different users. It appears to the author to be the most promising system.

To illustrate the use of REDUCE, we are going to discuss a sample program of general interest to many users. That is a program for coordinate transformation of contravariant vectors using a 4-dimensional space. Let λ^{μ}_{i} (x^{α}) denote a tetrad of contravariant vectors defined in the coordinate system (x^{α}) α = 0, 1, 2, 3, where μ (= 0,1,2,3) defines the coordinate components and i (= 0,1,2,3) defines the vector number. If we want to get the components of that tetrad in a new coordinate system (y^{β}) β = 0, 1, 2, 3, the relation between the new and old components is

$$\bar{\lambda}^{\mu}_{i}(y) = \frac{\partial y^{\mu}}{\partial x^{\nu}} \lambda^{\nu}_{i}(x), \qquad \mu = 0,1,2,3 \qquad (4)$$

Now $\frac{\partial y^{\mu}}{\partial x^{\nu}}$ can be written as a 4x4 matrix. If we want to transform from (y^{β}) to (x^{α}), we have

$$\lambda^{\mu}_{i}(x) = \frac{\partial x^{\mu}}{\partial y^{\nu}} \bar{\lambda}^{\nu}_{i}(y), \qquad \mu = 0,1,2,3 \qquad (5)$$

Where the matrix $\frac{\partial x^{\mu}}{\partial y^{\nu}}$ is the inverse of the matrix $\frac{\partial y^{\mu}}{\partial x^{\nu}}$. Let the old coordinate system be y^{α} (y^{0}= t,y^{1}= x,y^{2}= y,y^{3}= z), and let the new coordinate system be x^{β}(x^{0}=t,x^{1}=r,x^{2}=θ,x^{3}=∅) where we have the transformation

$$\left.\begin{aligned} y^{0} &= x^{0} \\ x &= r \sin\theta \cos\emptyset \\ y &= r \sin\theta \sin\emptyset \\ z &= r \cos\theta \end{aligned}\right\} \qquad (6)$$

Let the tetrad vectors be given in the old system of coordinates (y^{α}) in the following matrix, where A, B, D are functions of $r(=(x^{2}+y^{2}+z^{2})^{\frac{1}{2}})$.

$$\overset{\mu\ \ 0123}{\longrightarrow}$$

$$\lambda^{\mu}_{i} = \begin{vmatrix} A(r) & D(r).x & D(r).y & D(r).z \\ 0 & B(r) & 0 & 0 \\ 0 & 0 & B(r) & 0 \\ 0 & 0 & 0 & B(r) \end{vmatrix} \qquad (7)$$

The program which is written to carry out such calculations is listed below. The statements 6,38 are control statments to reduce the form of the output. The statements 7,39 are to direct the output to a file and to shut it respectively. The statements 8,9 are to declair functions and matrices used.

Extract from a REDUCE rogram

```
 6 ON NERO;
 7 OUT MENA8;
 8 OPERATOR X,Y,A,B,D;
 9 MATRIX TL(4,4),TLI(4,4),VMO(4,4),VM(4,4);
10 Y(0):=X(0);
11 Y(1):=X(1)*SIN(X(2))*COS(X(3));
12 Y(2):=X(1)*SIN(X(2))*SIN(X(3));
13 Y(3):=X(1)*COS(X(2));
14 VMO:=MAT((A(X(1)),D(X(1))*Y(1),D(X(1))*Y(2),D(X(1))*Y(3)),
15          (  0   ,  B(X(1))   ,     0      ,     0       ),
16          (  0   ,     0      ,  B(X(1))   ,     0       ),
17          (  0   ,     0      ,     0      ,  B(X(1))    ));
18 FOR I:=0:3 DO FOR J:=0:3 DO TL(I+1,J+1):=DF(Y(I),X(J))$
19 LET DF(X(1),X(1))=1,
20     DF(X(2),X(1))=0,
21     DF(X(3),X(1))=0,
22     DF(X(0),X(1))=0,
23     DF(X(1),X(2))=0,
24     DF(X(2),X(2))=1,
25     DF(X(3),X(2))=0,
26     DF(X(0),X(2))=0,
27     DF(X(1),X(3))=0,
28     DF(X(2),X(3))=0,
29     DF(X(3),X(3))=1,
30     DF(X(0),X(3))=0,
31     DF(X(1),X(0))=0,
32     DF(X(2),X(0))=0,
33     DF(X(3),X(0))=0,
34     DF(X(0),X(0))=1;
35 FOR ALL XL LET COS(XL)**2=1-SIN(XL)**2;
36 TLI:=1/TL;
37 VM:=VMO*TP(TLI);
38 OFF NERO;
39 SHUT MENA8;
40 CLEAR TL,TLI,VMO,VM;
```

Input of the program is given in the statements (10-13), and (14-17), which are just the transformation (6) and the tetrad (7) respectively. Statement 18 is to calculate the 4x4 matrix $\frac{\partial y^{\mu}}{\partial x^{\nu}}$ and to put the result in TL. The statement 19-34 represents the differentiation rules. The statement 35 is for using the identity $\sin^2 Q + \cos^2 Q = 1$, for any Q. The statement 36 is an instruction to get the inverse of TL, as we need the inverse transformation (5) not the direct (4). The transformation (5) , given by the statement 37, is written in a matrix form, where TP(TLI) is the transpose of TLI matrix. The statement 40 is to clear the store used. The following is a sample of the final results of the program.

```
70  VM(1,1):= A(X(1))
72  VM(1,2):= D(X(1))*X(1)
74  VM(2,2):= B(X(1))*COS(X(3))*SIN(X(2))
76  VM(2,3):= (B(X(1))*COS(X(2))*COS(X(3)))/X(1)
78  VM(2,4):= ( - B(X(1))*SIN(X(3))/(SIN(X(2))*X(1))
80  VM(3,2):= B(X(1)*SIN(X(3))*SIN(X(2))
82  VM(3,3):= (B(X(1))*COS(X(2))SIN(X(3)))/X(1)
84  VM(3,4):= (B(X(1))*COS(X(3)))/(SIN(X(2))*X(1))
86  VM(4,2):= B(X(1))*COS(X(2)
88  VM(4,3):= ( - B(X(1))*SIN(X(2)))/X(1)
```

The computer used for running this job is the IBM 360/168 NUMAC at Newcastle, the operating system is MTS. The top level language version : REDUCE 2(Aug-12-75(MTS July-1-77)). The CPU time used is 4.742 seconds.

CONCLUDING REMARKS

The symbolic problems explained in the third section are mere samples of the particular fields. There are more problems which can be treated successfully using algebraic computing. There are also some systems for algebraic manipulation other than those discussed in the present paper. Furthermore, there are versions for most of the systems discussed, offering more facilities to users. For example, there is a new version of REDUCE, according to which the statement (19-34),given in the program of the previous section, will be deleated.

One of the main difficulties in using algebraic manipulation is that one cannot estimate the time that a certain process will probably take, and thus the user is unable to give an upper-time limit for his job.

Although algebraic manipulation has started more than twinty years ago, the number of users is still very limited compared with those dealing with the other two faces of the computer. Many users have not yet heared about the subject. Others cannot obtain suitable systems, as most of them are not easily available. Even when a user may have access of a certain system, other difficulties may arise with regard to the core size and the operating system used. We hope that by creating general interest in algebraic computing, then potential users could find the way out of these difficulties.

ACKNOWLEDGEMENT

The author wishes to thank the authorities of King's college, London University, for providing the funds required to cover

the costs of using the computer, while the author was a visitor at the Mathematics Department there.

REFERENCES

1. d'Inverno,R.A., "Algebraic Computing in General Relativity" in "General Relativity and Gravitation" One Hundred Years After the Birth of Einstein, vol.1, 491-537, ed.A.Held(1980).
2. Hearn,A.C., "Scientific Applications of Symbolic Computations" in"Computer Science and Scientific Computations" , Academic Press Inc.,New York, 83-108 (1976).
3. Cohen,H.I., Leringe,Ö. and Sundbland,Y., "The Use of Algebraic Computing in General Relativity", Gen.Rel.Grav.,7,269-286 (1976).
4. Krasinski,A. and Perkowski,M., "ORTOCARTAN, A New Computer Program for Analytic Calculations in General Relativity", Gen.Rel.Grav.,13, 67-77 (1981).
5. Bogen,R., Golden,J., Geneseretu,M. and Doohovskoy,A., "MACSYMA Reference Manual", Laboratory for Comp.Sc.MIT(1977).
6. Hearn,A.C., "REDUCE 2 User's Manual"2nd ed.Utah Univ.(1975).
7. Frick,I., "SHEEP, A Computer Algebra System",Preprint,Institute of Theoretical Physics,Stockholm Univ.,Sweden (1977).
8. d'Inverno,R.A., "Algebraic Computing in General Relativity" Gen.Rel.Grav.,6,567-593 (1975).
9. Mikhail,F.I. and Wanas,M.I., "A Generalized Field Theory I" Proc.Roy.Soc.A,London,356, 471-481 (1977).
10. Mikhail,F.I. and Wanas,M.I.,"A Generalized Field Theory II", Int.J.Theor.Phys.,20,671-681 (1981).
11. Wanas,M.I., "Manipulation of Parameters Indicating the Physical Significance of any Absolute Parallelism Space Using REDUCE 2", To appear in the Bul.Int.Cong.Statistics, Computer Science,Social and Denographic Research (1985).
12. Wanas,M.I., "A Generalized Field Theory, Charged Spherical Symmetric Solution",to appear in Int.J.Theor.Phys. (1985).
13. McRuer,D.,Ashkenas,I. andGraham,D., "Aircraft Dynamics and Automatic Control", Princeton Univ.Press (1973).
14. Spain,B., "Tensor Calculus", 3rd ed.Oliver&Boyd (1960).
15. Diprit,A.,Henerard,J.and Rom,A., "Lunar Ephemeris: Delaunay's Theory Revisited",Science,168,1569-1570 (1970).
16. Dodd,J.E., "The Ideas of Particle Physics", Cambridge Univ. Press (1984).
17. Schlichting,H., "Boundary Layer Theory", McGrow-Hill(1955).
18. Thorne,K.S., "Non-Radial Pulsation of General Relativistic Stellar Models", Astrophys.J.,158,1-16(and 977-1020)(1969).
19. Thorne,K.S. and Campolataro,A., "Non-Radial Pulsation of Relativistic Stellar Models",Astrophys.J.,149,591-611(1967).
20. Hartle,J.B. and Thorne,K.S., "Slowly Rotating Relativistic Star,II Models for Neutron Stars and Super-Massive Stars", Astrophys.J.,153,807-834 (1968).