A TOOLBOX FOR NONLINEAR DYNAMICS

SHANNON COFFEY* [1], ANDRÉ DEPRIT† [2], ÉTIENNE DEPRIT*,
LIAM HEALY,‡ [3] AND BRUCE R. MILLER†

Abstract. Using the main problem of artificial satellite theory as an illustration, we review several developments which have had a significant impact on research in nonlinear dynamics. On the mathematical front, we point to the theory of Lie transformations; in the area of computational software, we explain how massively data parallel machines open the way for symbolic solution of large problems. Finally, we show how color graphics assist in the qualitative analysis of dynamical systems.

1. Introduction. The rapid progress in computers, software as well as hardware, has revolutionized research about dynamical systems. Prior to the computer age, mathematicians like Ch. Delaunay, G. W. Hill and E. W. Brown envisioned no other way for predicting the motion of a celestial object than by analytical theories. They encapsulated solutions to the differential equations in the form of power series into a body of formulas which they called a 'theory.' That phase once completed, they would turn to the task of breaking the theory into a sequence of steps, each one arranged to save as much of the intermediary calculations as possible. These arrangements which, today, we call 'flow charts', they referred to as the 'Tables' of a theory because they consisted for the main part in preparing partial results in the form of reference tables. Developing a Theory required years of effort, and so did the job of restructuring it to produce Tables. Mathematicians engaged in computational astronomy could not evade meeting head on the ultimate challenge: calculations in the almanac offices had to be done in 'real time.' Of what use would a lunar theory be if an average clerk could not predict the position of the moon from one place to the next in less time than it takes the moon to move from that place to the next?

With the advent of computers, numerical integration became the favored technique in orbit prediction for several reasons — modest use of processor time and memory, greater accuracy for shorter time periods, and ability to accommodate nonconservative forces too difficult to model analytically. More recently, however, analytic theories have enjoyed a resurgence owing to the increasing sophistication of algebraic software, the appearance of novel computational methods and the availability of color graphics. Nonlinear dynamics, at last, is finding a toolbox to serve its purposes.

Our line of research started many years ago, with the study of families of periodic orbits emanating from the triangular equilibria in the restricted problem of three bodies. In the neighborhood of the equilibria, the periodic orbits are functions of a small parameter to be expanded as Fourier series; coefficients in these series are obtained

*Naval Research Laboratory, Washington, D.C. 20375.

†National Institute of Standards and Technology, Gaithersburg, MD 20899. Partial support from the Computational and Applied Mathematics Program at the Defense Advanced Research Project Agency.

‡NAS/NRC Cooperative Research Associate at the Naval Research Laboratory.

Reprinted from the book, Computer Aided Proofs in Analysis
edited by Kenneth R. Meyer and Dieter S. Schmidt

as roots of recursive sets of linear equations [1,2]. These symbolic calculations led to devise general procedures for manipulating *en bloc* symbolic expressions of the form

$$(1) \qquad \sum_{i,j} C_{i,j} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \left\{ \begin{array}{c} \cos \\ \sin \end{array} \right\} (j_1 y_1 + j_2 y_2 + \ldots j_n y_n).$$

Expressions of that kind were dubbed Poisson series, and the name has stuck ever since [3].

During the same period, Imre Iszak [4], David Barton [5] and others were looking at ways of reproducing and expanding the grand literal theories upon which astronomers of the XIX$^{\text{th}}$ century had lavished so much time. Foremost among those stands the lunar theory. Rigorous to a fault, Delaunay had taken great pains to document his hand calculations, even taking the extraordinary precaution of reporting from one series to the next how each individual term arose. No wonder then that researchers wrapped up in algebraic processors eagerly tested the power and robustness of their systems against the achievements of Delaunay.

Armed with their own symbolic algebra system MAO (short for Mechanized Algebraic Operations), A. Deprit, J. Henrard and A. Rom took the challenge in steps. They first automated Birkhoff's normalization technique in the neighborhood of an equilibrium for a Hamiltonian system with two degrees of freedom [6], then expanded the main problem of artificial satellite theory in powers of the eccentricity [7] — a task that D. Brouwer speculated would be intractable with the computers available at the time.

Much has happened since MAO succeeded in checking Delaunay's theory, and extending it from order 8 to order 13 and even higher [8]. MAO's creators wavered for a while between assembler and Fortran implementations [9], eventually settling upon Fortran with a RATFOR preprocessing stage, yet, at the same moment, branching into PL/I where packaging the code into macros yielded more flexibility. The increasing need for flexibility in coding and ease of application to problems prompted a radical change of direction in 1984, when B. Miller transported MAO to a Lisp workstation to make programming 'objects' out of Poisson series. The Lisp version of MAO readily incorporated the new Lie algebraic methods devised to normalize Hamiltonian systems by canonical transformations near the identity mapping. Implementing the method of Lie transformations [10] in MAO stimulated mutual refinement and extensions between computational algorithms and their software implementation.

Over the years, applications extended from the three-body problem to other sectors of celestial mechanics, overflowing eventually into the general areas of classical mechanics. The latter applications include the Stoermer problem, i.e. the motion of a charged particle in a magnetic dipole, the dynamics of orbiting dust [11], the Hénon-Heiles oscillator, the quadratic Zeeman effect [12,13], and the Toda 3-point lattice [14]. All these problems pertain to the class of perturbed integrable Hamiltonian systems with a principal part that is either a Keplerian problem or a harmonic oscillator.

2. Symbolic algebra and Hamiltonian systems. In the 1960's and 70's, astronomers were not talking of normalizing Hamiltonians and their equations. Instead, they spoke of averaging Hamiltonians over fast variables, which they did by using one of Poincaré's *méthodes nouvelles*. In that context, averaging transformations

$$\chi : (q, Q) \rightarrow (p, P)$$

are represented by a system of implicit equations

$$P = \partial S / \partial p, \qquad q = \partial S / \partial Q$$

derived from a function in mixed variables $S \equiv S(p, Q)$.

Ideal for eliminating fast variables from a Hamiltonian, Poincaré's method is not well suited, though, to obtaining the averaging transformation in explicit form. For it necessitates solving simultaneous implicit equations for half of the system, and substituting the solutions into the other half. Astronomers were not yet equipped computationally to perform these operations beyond the first order. They were, of course, familiar with Lagrange's *inversion formula* for solving elementary implicit equations in one independent variable; beyond that, they knew only of the method of successive approximations. The theory of Lie transformations [15,10] evolved in the late 1960's to stave off these difficulties.

2.1. Normalization as an algebraic operation. We concern ourselves exclusively with canonical transformations depending on a small parameter ϵ of a particular type, namely those which are the general solution $(p(q, Q, \epsilon), P(q, Q, \epsilon))$ of a Hamiltonian system

$$\frac{dp}{d\epsilon} = \frac{\partial W}{\partial P}, \qquad \frac{dP}{d\epsilon} = -\frac{\partial W}{\partial p} \tag{2}$$

satisfying the initial condition

$$p(q, Q, 0) = q, \qquad P(q, Q, 0) = Q.$$

In regard to these objects, our major problem is a basic one: Given the power series

$$F(p, P, \epsilon) = \sum_{n \geq 0} \frac{\epsilon^n}{n!} F_n(p, P),$$

if W itself is a power series in ϵ, how do we obtain by machine the transformed function

$$F^*(q, Q, \epsilon) = F(p(q, Q, \epsilon), P(q, Q, \epsilon))$$

as a series

$$F^* = \sum_{n \geq 0} \frac{\epsilon^n}{n!} F_n^*(q, Q)$$

in powers of the small parameter? Well, we found that automated conversion works well when we recognize that the coefficients F_n^* are the values at $\epsilon = 0$ of the iterates $\mathcal{L}_W^n$ for the Lie derivative

$$\mathcal{L}_W : F \rightarrow (F; W) \tag{3}$$

of F in the direction of the vector field W. The notation $(F; W)$ in (3) represents the Poisson brackets of the two functions F and W.

Along those lines, normalizing a Hamiltonian that is a series in the powers of a small parameter ϵ

$$\mathcal{H} = \mathcal{H}_0 + \sum_{n\geq 0} \frac{\epsilon^n}{n!}\mathcal{H}_n \tag{4}$$

means building a Lie transformation χ such that $(\mathcal{H}^*;\mathcal{H}_0^*) = 0$ for the transformed Hamiltonian $\mathcal{H}^* = \chi(\mathcal{H})$.

Actually, for most of the problems handled in the literature, normalization is merely an algebraic operation. Indeed, let $\mathcal{A}$ denote an algebra of functions containing the terms $\mathcal{H}_n$, and let $\mathcal{L}_0$ stand for the Lie derivative

$$\mathcal{L}_0 : F \to (F;\mathcal{H}_0).$$

In most cases, $\mathcal{L}_0$ acts as a semi-simple operator in $\mathcal{A}$, i.e. the vector space $\mathcal{A}$ may be decomposed into the direct sum

$$\mathcal{A} = \ker(\mathcal{L}_0|\mathcal{A}) \oplus \mathrm{im}(\mathcal{L}_0|\mathcal{A}).$$

Under that assumption, normalizing $\mathcal{H}$ means finding a Lie transformation which projects $\mathcal{H}$ into an element $\mathcal{H}^*$ in the kernel of the Lie derivative restricted to $\mathcal{A}$.

2.2. Automated normalizations. Without entering into details, recall that Lie transformations, and hence normalizations, are built by induction. Precisely this property makes normalizations most amenable to construction by computers. As the construction progresses at each order in ϵ, one encounters a partial differential identity of the form

$$\mathcal{L}_0(W_n) + \mathcal{K}_{n,0} = \tilde{\mathcal{H}}_{n,0} \tag{5}$$

which we try to satisfy by choosing the two unknowns, namely the term W_n of order n in the Hamiltonian from which are derived the transformation equations (2) on the one hand, and the term $\mathcal{K}_{n,0}$, likewise of order n, in the transformed Hamiltonian of the system on the other.

As it happens most often, while $\tilde{\mathcal{H}}_{n,0}$ belongs to the Lie algebra $\mathcal{A}$, W_n belongs to an algebra $\mathcal{B}$ of functions such that $(f;g) \in \mathcal{A}$ for any f in $\mathcal{A}$ and any g in $\mathcal{B}$. In those circumstances, we choose $\mathcal{K}_{n,0}$ to be the part of $\tilde{\mathcal{H}}_{n,0}$ in the kernel of $\mathcal{L}_0$, which makes W_n a counter-image of $\tilde{\mathcal{H}}_{n,0} - \mathcal{K}_{n,0}$ in $\mathcal{B}$.

Solving (5) for W_n rarely proves trivial. Our favorite practice has been to select a standard representation for the algebra $\mathcal{A}$ and determine the action of $\mathcal{L}_0$ upon the appropriate expressions in $\mathcal{A}$. These rules of action indicate membership in the kernel, and tell us how to determine counter-images.

As a simple example, consider the elliptic oscillator

$$\mathcal{H}_0 = \tfrac{1}{2}\left(X^2 + x^2\right) + \tfrac{1}{2}\left(Y^2 + y^2\right).$$

In complex variables the Hamiltonian and Lie derivative take the form

$$\mathcal{H}_0 = i(uU + vV) \qquad \mathcal{L}_0 = i\left(u\frac{\partial}{\partial u} - U\frac{\partial}{\partial U} + v\frac{\partial}{\partial v} - V\frac{\partial}{\partial V}\right).$$

Assume that the perturbation belongs to the complex algebra $\mathcal{A}$ of multivariate polynomials in the variables u, v, U, V. The action of the Lie derivative on a given monomial is

$$\mathcal{L}_0(u^\alpha U^\beta v^\gamma V^\delta) = i(\alpha - \beta + \gamma - \delta) u^\alpha U^\beta v^\gamma V^\delta.$$

A monomial belongs to the kernel of $\mathcal{L}_0$ if and only if $\alpha - \beta + \gamma - \delta = 0$; otherwise, $u^\alpha U^\beta v^\gamma V^\delta$ is the image of the monomial

$$\frac{i u^\alpha U^\beta v^\gamma V^\delta}{(\beta - \alpha + \delta - \gamma)}.$$

Clearly, all algebraic computations involved in the normalization of an elliptic oscillator with polynomial perturbations are closed in the algebra $\mathcal{A}$. The successive differential equations of type (5) are solved automatically by syntactic matching of patterns and mappings. A similar argument holds when it is necessary to introduce dependent variables to reduce the complexity of expressions; the canonical forms aid in performing simplifications of these expressions.

2.3. Canonical simplifications. Not many of the systems one encounters in physics or astronomy belong to the class of polynomial Hamiltonians. Furthermore, given an algebra of perturbations, there exists no general procedure for building generators for the kernel of the Lie derivative restricted to that algebra.

Out of these difficulties grew a new line of research. Rather than going to the extreme of forcing the transformed Hamiltonian into the kernel of the Lie derivative $\mathcal{L}_0$, one conceives of converting the original Hamiltonian by Lie transformation into a 'simpler' function not necessarily in the kernel. So far, the few trials performed using this compromise solution have been very promising. In the next section, we mention how 'canonical simplifications' helped in normalizing the main problem of artificial satellite theory to the fourth order without resorting to developments in the powers of the eccentricity. As a result, the idea of canonical simplifications is now explored vigorously to see how it would contribute a lunar theory without expansions in the powers of the eccentricity of the sun, and also to study the attitude of a rigid body rotating about a fixed point.

Finally, we should question our fixation with canonical transformations. It is quite conceivable that normalization could be achieved by Lie transformations that are not canonical. Consider, for instance, a perturbed circular pendulum, the principal part being

$$\mathcal{H}_0 = \tfrac{1}{2}\Theta^2 - \omega^2 \cos\theta$$

when all terms $(\mathcal{H}_n)_{n>0}$ in the perturbation belong to the algebra $\mathcal{A}$ of periodic functions of the elongation θ with coefficients in the kernel of $\mathcal{L}_0$. Already at first order, a normalization by canonical Lie transformation introduces various types of elliptic integrals, thereby making it necessary for the software to extend $\mathcal{A}$ into a broader algebra mixing ordinary trigonometric functions with elliptic functions.

Some time ago, a canonical simplification was found to remove all periodic terms at the first and second order [16]. But one can achieve far more if one is willing to operate with Lie transformations that are not canonical. It is shown in [17] how to build by induction a non-canonical transformation $(\theta, \Theta) \to (\phi, \Phi)$ to convert not the Hamiltonian itself but the canonical equations derived from the perturbed Hamiltonian $\mathcal{H}$ into differential equations of the form

$$\frac{\partial \phi}{\partial \tau} = \Phi, \qquad \frac{\partial \Phi}{\partial \tau} = -\omega^2 \sin \phi. \tag{6}$$

Someone interested in performing the reduction by machine will appreciate the fact that elliptic functions and integrals do not show up in the literal developments. The construction in [17] takes place solely in the algebra of trigonometric functions in θ. There is, of course, a price to pay for using non-canonical transformations around Hamiltonian systems. The price here is a change of independent variable defined through an implicit equation. But, as we learn from K. R. Meyer in these proceedings, the symbolic solution of implicit equations is advancing rapidly.

3. Application: satellite theory. The discussion of Section can be made more concrete by examining the role of normalizations and canonical simplifications in the theory of an artificial satellite.

We assume the satellite can be taken as a point of negligible extent and consider the attraction by the Earth as the only force. Attach to the Earth an orthonormal frame $(\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$, where $\boldsymbol{b}_3$ is the direction of the polar axis, and $\boldsymbol{b}_1$ lies on the Greenwich meridian. The position of the satellite in the Earth frame is given by the vector

$$\boldsymbol{x} = r\left[\boldsymbol{b}_3 \sin\beta + (\boldsymbol{b}_1 \cos\lambda + \boldsymbol{b}_2 \sin\lambda) cos\beta\right],$$

with $r > 0$ representing the distance from the center of the Earth to the satellite, β the geographic latitude such that $-\pi/2 \leq \beta \leq \pi/2$, and λ the geographic longitude such that $0 \leq \lambda < 2\pi$. Let μ stand for the Gaussian constant of the Earth, i.e., the product $\mu = k^2 m_\oplus$ of the Newtonian constant of universal attraction by the mass of the Earth; let also α denote the equatorial radius of the Earth. Then the gravity field of the Earth at the position occupied by the satellite is minus the gradient $\nabla_{\boldsymbol{x}} \mathcal{V}$ of the potential

$$\mathcal{V} = -\frac{\mu}{r} \sum_{n \geq 0} \left(\frac{\alpha}{r}\right)^n \left[\sum_{0 \leq m \leq n} (C_{n,m} \cos m\lambda + S_{n,m} \sin m\lambda)\, P_m^n(\sin\beta)\right].$$

The functions $P_m^n(w)$ are the associated Legendre polynomials

$$P_m^n(z) = (-1)^m \sqrt[m]{1 - z^2}\, \frac{d^m}{dz^m} P_n(z).$$

The field parameters $C_{n,m}$ and $S_{n,m}$ are dimensionless constants. By definition of the Gaussian constant, $C_{0,0} = 1$, and by setting the Earth-bound frame at the center of mass of the Earth, there follows that all three coefficients $C_{1,0}$, $C_{1,1}$ and $S_{1,1}$ are zero.

Let $(\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3)$ be an orthonormal frame fixed in space set at the center of mass of the Earth. Assume that the polar axis of the Earth $\boldsymbol{b}_3$ is fixed in space, and that the Earth rotates at a constant angular velocity $n_\oplus$ about $\boldsymbol{b}_3$. Then we choose $\boldsymbol{s}_3 = \boldsymbol{b}_3$ and consider only the *zonal* form of the problem in which the potential reduces to the function

$$\mathcal{V} = -\frac{\mu}{r}\Big(1 + \sum_{n \geq 2} \left(\frac{\alpha}{r}\right)^n C_{n,0} P_n(\sin\beta)\Big),$$

$P_n(w) = P_{n,0}(w)$ being the Legendre polynomial of degree n. By common usage, one sets $C_{n,0} = -J_n$. For the Earth, $J_2 \approx 10^{-3}$ whereas for $n \geq 3$, $|J_n| \approx 10^{-6} \approx J_2^2$. On account of these relative sizes, we retain only the first two terms. The dynamical system $\mathcal{H}_0 + J_2\mathcal{H}_1$, where

$$\begin{aligned} \mathcal{H}_0 &= \frac{1}{2}\left(R^2 + \frac{\Theta^2}{r^2}\right) - \frac{\mu}{r}, \\ \mathcal{H}_1 &= \frac{\mu\alpha^2}{r^3} P_2(\sin\beta), \end{aligned}$$

constitutes the main problem in artificial satellite theory.

The main problem being a perturbed Keplerian system, normalization amounts to eliminating the mean anomaly. After many trials, we discovered at last that the normalization arises most directly as the product of two canonical transformations [18]. The first step is only a canonical simplification whereby the Hamiltonian is stripped of all dependence on the mean anomaly except for terms in $(p/r)^2$. More precisely, $\mathcal{H}$ is converted into a series of the type

$$\mathcal{H} = \mathcal{H}_0 + \left(\frac{\alpha}{r}\right)^2 \sum_{n>0} J_2^n C_n$$

whose coefficients C_n belong to the kernel of $\mathcal{L}_0$. We dubbed this simplification the 'elimination of the parallax.' From the point of view of symbolic algebra, this simplification has the incomparable advantage of taking place entirely in the algebra of functions of the type

$$F = \left(\frac{\alpha}{r}\right)^2 \sum_{n\geq 0} C_n \left\{ \begin{array}{c} \cos \\ \sin \end{array} \right\} n\theta$$

with coefficients C_n in the kernel of $\mathcal{L}_0$. More significantly, elimination of the parallax removed most of the difficult trigonometric functions in θ which cause difficulty in an ordinary Delaunay normalization, allowing us to eliminate the mean anomaly to the fourth power of J_2 by machine without resorting to developments in the powers of the eccentricity. The latter alternative is no longer attractive with today's aerospace engineers launching satellites at eccentricities greater than 0.1. This achievement paved the way for development of a complete third order theory for the main problem of an artificial satellite.

Following the elimination of the parallax, a Delaunay normalization eliminated the remaining short period terms for the main problem. To complete the theory, a long period transformation, once again a Lie transformation, was constructed to produce the secular Hamiltonian [19].

Our first attack on the satellite problem produced the third order theory mentioned above and a second order theory which included zonals through J_7. The algebra was first performed on the early PL/I versions of MAO and later on Dasenbrock's Fortran processor. Integer overflows, however, caused by the 32 bit word length on the Texas Instruments ASC vector computer limited the number of zonals that could be included in the potential to J_7. Recently, in exercising the Lisp version of MAO, we revisited the satellite problem. Given the extended precision integer arithmetic of the Lisp machine, we were able to produce the transformations to second order for the combined potential of J_2 through J_{10}. We mention this detail to underscore the fact that integer arithmetic with indefinite length has become an indispensable instrument in the toolbox of nonlinear mechanics.

4. Symbolic algebra systems. In principle, general purpose symbolic manipulation systems, such as Macsyma, Reduce and Mathematica, provide all the functions necessary to perform the desired normalizations and simplifications. These general purpose processors, however, suffer from the "free-form" specification of expressions which ignores the underlying algebraic structure. Specifying the exact operation desired or producing results in the precise form required often proves quite difficult. Moreover, by handling mathematical expressions of arbitrary structure, these systems cannot optimize their storage schemes or their algebraic algorithms for expressions of a fixed structure. Our experience with both general and special purpose processors clearly demonstrates that our research problems must be attacked with software specifically designed for the task at hand. In this paper, therefore, we concentrate on special purpose algebraic systems.

4.1. Serial processing. The early symbolic processors were special purpose programs, most of them, if not all, developed by individuals interested in solving specific classes of problems. Although each processor grew in capabilities with the research interests of its creator, nonetheless, the processor never departed from the algebraic structure characterizing the class of problems for which it was originally designed.

The Echeloned Series Processor (ESP) was developed in assembler on the IBM 360/44 computer for Delaunay's theory [20]. The ESP turned out to be barely adequate for the lunar theory, but spurred development of symbolic codes along the same pattern. In 1973, R. Dasenbrock at the Naval Research Laboratory developed a processor in Fortran [21]. This processor finds continued use at NRL [22]; portable versions were made at the University of Zaragoza for IBM/RT's, the CNES in Toulouse for Sun's, and the University of Cincinnati for PC's.

Developing the artificial satellite theory on Dasenbrock's processor exposed several limitations of the compilers and hardware of the time. The integer length of 32 bits limited the size of the rational coefficients. Memory limitations always posed a serious problem, partially as a result of scrimped core and no virtual memory, and partially as a result of the flat data representation for the expressions. Long run times of several hours on the ASC advanced array processor were common for large problems. Indeed, the ASC proved to be of little advantage for these types of calculations. Algebraic manipulations require tremendous amounts of localized operations like sorting, combining terms, and other operations to maintain the canonical form of the expressions. Such local operations prove difficult to vectorize, and we ended up using the ASC as a fast scalar machine. In sum, while the software could exploit very efficiently the algebraic structures inherent to the canonical representations, it made it very inconvenient to fit the problem at hand.

In 1984, Miller reincarnated MAO on a Symbolics workstation using an object-oriented dialect of Lisp. This version of MAO introduced a number of new features, most notably the polymorphism of functions and advanced mathematical typography on a bitmapped screen.

In the current version of MAO, both mathematical functions (such as multiplication) and operational functions (such as printing) are polymorphic. Each class of algebraic object knows how to handle addition, multiplication, printing, etc. in the appropriate way. For example, the product of two algebraic objects may be a multiplication or a scaling depending on the types of the operands. By using these generic operations, the code which implements polynomial multiplication may be optimized quite readily, since this routine concerns itself only with manipulating the two expressions at the polynomial level. The types of the coefficients and even the number of variables prove irrelevant, as long as generic routines exist to multiply and add terms.

The polymorphism of functions also provides a user of MAO with great freedom in structuring the solutions to his problem. By combining modules defined for manipulating polynomials, series and Fourier sums, the user constructs expressions matching exactly the algebra of his computational methods. For example, one might choose to work with series in the small parameter ϵ, whose terms are polynomials in L and n, whose coefficients are Fourier sums in the angles ℓ and g, whose coefficients are polynomials in e, whose coefficients are rational numbers (see Figure 1). Having this hierarchy of algebras, MAO keeps all expressions in this form, ready for any simplifications or other operations requiring the results to be in canonical form.

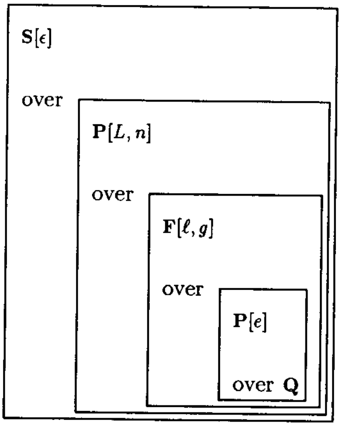

Figure 1: Hierarchy of algebras

On the whole, MAO has turned into a very handy tool for solving medium size problems. Yet, in spite of all provisions to ensure speed and efficiency, MAO appears too slow to handle very large problems. We do not invent these problems for the sake of pushing the equipment to the limit; they are out there, begging for a solution. The main problem of lunar theory was solved to an accuracy of 50m in the radius. Presently, the U.S. Naval Observatory time service requires a solution accurate to a

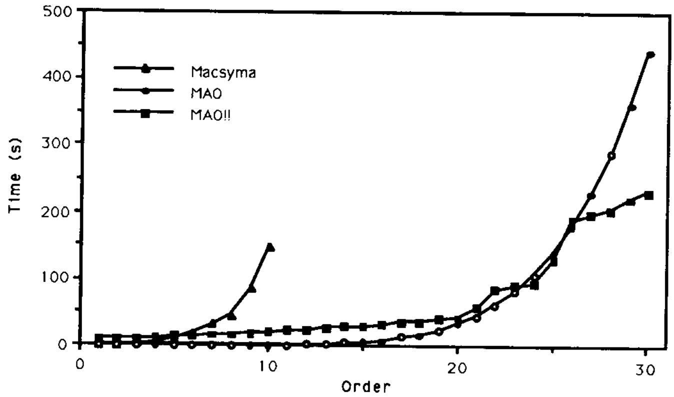

Figure 2: $(a/r)^5$ with rational coefficients

few centimeters. Most likely, so stringent a precision will tax the Symbolics environment beyond its present capabilities in both speed and memory. Thus, we turn once again to rebuilding MAO on a new type of machine, this time on a massively parallel processor — the Connection Machine (CM).

4.2. Parallel processing. As suggested in Figure 2, a massively parallel processor constitutes a powerful tool for manipulating large Poisson series. In the benchmark picture, we started with the ratio a/r as a Fourier series in the mean anomaly ℓ with polynomial coefficients in the eccentricity e over the rationals. Textbooks in celestial mechanics give only the first few terms of the series a/r:

$$\begin{aligned}\frac{a}{r} = 1 \quad & + e\cos\ell\,(1 - \tfrac{1}{8}e^2 + \tfrac{1}{192}e^4 + \ldots) + e^2\cos 2\ell\,(1 - \tfrac{1}{3}e^2 + \ldots) \\ & + e^3\cos 3\ell\,(\tfrac{9}{8} - \tfrac{81}{128}e^2 + \ldots) + e^4\cos 4\ell\,(\tfrac{4}{3} + \tfrac{2}{5}e^2 + \ldots) \\ & + e^5\cos 5\ell\,(\tfrac{625}{384} + \ldots) + \ldots.\end{aligned}$$

Acting upon a suggestion made by Prof. K. R. Meyer, we solved Kepler's equation by means of a Lie transformation which we then applied to obtain the series a/r to power 30 in e. We could have applied the same transformation to the power $(a/r)^5$, but we did not. Our purpose here is to produce an expression long enough that, by raising it to the fifth power, would exercise to the fullest the capabilities of the Connection Machine.

In one calculation of Figure 2, we use Macsyma; in another, we employ MAO on a Symbolics Lisp workstation. In the third, we compute the product on a Connection Machine, for which we have developed a package of procedures (MAO!!) written in *Lisp. In each run, we timed the operations for increasing orders in e. At order $n = 30$, each factor contains 256 terms. Macsyma is desperately slow—it took almost two and a half minutes to compute the result to order 10. MAO reached order 30 in approximately seven and a half minutes, while MAO!! did the same nearly twice as fast. The shape of the curve representing timings on the CM exhibits several interesting features. Up to order 22, the curve is practically linear; this is easily explained by the small increase in overhead due to the incremental growth in the length of the series from one order to the next. Then, the timing curve jumps to another linear slope slightly steeper than the initial one, and the phenomenon reproduces itself periodically at higher orders. At order 22, the jump results from the program having exhausted the available pool of physical processors and reconfiguring the real machine to behave as a logical machine with twice as many virtual processors. At order 26, the configuration changed again, again redoubling the number of virtual processors.

The tool developed on the Connection Machine has been dubbed MAO!! [24] — the suffix '!!' follows the CM programmer's convention for denoting parallelism. MAO!! achieves its gains over MAO by spreading Poisson series over thousands of processors. Each processor in the CM holds a single term of a Poisson series. This distribution provides a simple resource allocation scheme flexible enough to deal with the constant explosion and implosion of partial results so typical of symbolic algebra. In addition, the scheme permits many series to remain active in the CM memory.

From the standpoint of parallelism, algebraic operations fall into two classes. Multiplication by a monomial, partial differentiation and integration are local operations, requiring only isolated computation in each processor. On the other hand, multiplication and simplification are global operations in the sense that processors representing terms of the series must communicate among themselves. Global operations bring forth the real power of the CM.

Among the global operations, we concentrated on two problems: the simplification of like terms, and the multiplication of Poisson series. In both cases, we succeeded in introducing a high degree of parallelism. The secret was to take advantage of the possibility of restructuring the CM as a grid on which global patterns of communication act like translations in n-space. In this regard, combination of like terms turns into a sorting to put all like terms next to one another along intervals of a one-dimensional grid; after sorting comes a scanning to sum the like terms in the processor at the end of each segment.

MAO!! multiplies Poisson series by replicating the factors and forming all partial products at the same time. The code is best understood by looking at a simple example. To multiply a second degree polynomial in one variable $a + bx + cx^2$ by a polynomial $A + Bx$, we arrange the machine so that the first six processors on the one-dimensional grid contain the following quantities.

$$\begin{array}{cccccc} a & b & c & a & b & c \\ x^0 & x^1 & x^2 & x^0 & x^1 & x^2 \\ A & A & A & B & B & B \\ x^0 & x^0 & x^0 & x^1 & x^1 & x^1 \end{array}$$

Then, all partial products are computed in parallel so that the one-dimensional grid now holds the quantities below.

a	b	c	a	b	c
x^0	x^1	x^2	x^0	x^1	x^2
A	A	A	B	B	B
x^0	x^0	x^0	x^1	x^1	x^1
aA	bA	cA	aB	bB	cB
x^0	x^1	x^2	x^1	x^2	x^3

There remains to pass the terms to the simplification routine and store away the remaining terms. The dynamic virtualization mechanism of the CM makes the multiplication of large series possible. For instance, when multiplying two Poisson series of 256 terms each, the intermediate result will have 2^{17} terms, while our CM provides only 2^{14} physical processors. Nevertheless, the CM may be configured as if it had 2^{17} virtual processors, where each physical processor emulates eight virtual ones.

In our opinion, massive parallelism presents a viable option for processing very large Poisson series. We intend to continue the development of MAO!! by introducing hierarchical structures like those in MAO, as well as refining the parallel and numerical algorithms. In addition, the introduction of the Data Vault — a parallel mass-storage device connected directly to the CM — offers the possibility of caching large numbers of series, constituting in effect a virtual memory.

5. Graphical studies of flows. Although a normalization may vastly simplify a system, there remains much to do to understand its global behavior. By virtue of the normalization described in Section, the system has been reduced from one describing the evolution of the position of the satellite to one describing the evolution of the 'state' of the orbit, the instantaneous ellipses upon which the satellite moves. The stable equilibria of the reduced system indicate which orbits are 'safe'; a satellite would remain on or near such an orbit for a long time. The purpose of performing the normalization is to better understand the dynamics of the system. Once the algebraic manipulations are done and we have the normalized Hamiltonian, we still need to extract the qualitative features of the dynamics.

We are facing, from an algebraic point of view, a comparatively unstructured problem: whereas previously we were able to identify algebraic structures amenable to automated manipulations, we are now dealing with Hamiltonians usually so short that we can afford to process them by general purpose systems for symbolic and algebraic manipulations. Our guides at this point are insight and experience. For we are now trying to capture the major features in the global flow determined by normalized Hamiltonian equations.

As expected we start by locating the singularities in the system. In particular we are especially attentive to the creation and annihilation of equilibria as the parameters of the system are changing. One can envision several ways of learning these things, and, in fact, we use no single tool but a combination of numeric, symbolic and graphical techniques.

5.1. Topology to the rescue. Carried away by tradition, the first people to study the satellite problem assimilated the phase space (g, G) to a cylinder, in part because they felt comfortable in looking at the system as a simple pendulum under perturbations. The fact is that the model of the pendulum is totally misleading. It misses those exceptional situations at $G = L$ where, the eccentricity being zero, the argument of perigee g has no meaning. Furthermore, it excludes the whole manifold of equatorial orbits because when the inclination I is zero, i.e., at $G = H$, the longitude h of the ascending node has no meaning.

However, as Elie Cartan showed some eighty years ago, on each manifold $L =$ constant, the set of bounded orbits that one gets from making constant all Delaunay elements but the mean anomaly ℓ consists of a pair of two-dimensional spheres. One can see that easily: indeed the vector functions

$$\boldsymbol{G} = \boldsymbol{x} \times \boldsymbol{X} \quad \text{and} \quad \boldsymbol{A} = (L/\mu)(\boldsymbol{X} \times \boldsymbol{G} - \mu\boldsymbol{x}/r)$$

are independent of ℓ, and so are $\boldsymbol{\sigma} = \frac{1}{2}(\boldsymbol{G} + \boldsymbol{A})$ and $\boldsymbol{\delta} = \frac{1}{2}(\boldsymbol{G} - \boldsymbol{A})$. The identities $\|\boldsymbol{\sigma}\|^2 = \|\boldsymbol{\delta}\|^2 = L^2/4$ define the two spheres recognized by Cartan.

As we are interested mainly in perturbed Keplerian problems where the longitude of the ascending node is ignorable we found convenient to use coordinates other than the vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\delta}$. We take

$$\begin{aligned} \eta_1 &= \sqrt{L^2 - H^2}\cos h, & \xi_1 &= LG\sin I\, e\cos g, \\ \eta_2 &= \sqrt{L^2 - H^2}\sin h, & \xi_2 &= LG\sin I\, e\sin g, \\ \eta_3 &= H, & \xi_3 &= G^2 - \tfrac{1}{2}(L^2 + H^2). \end{aligned}$$

In these coordinates, the orbital space is represented by the two spheres

$$\eta_1^2 + \eta_2^2 + \eta_3^2 = L^2, \qquad \xi_1^2 + \xi_2^2 + \xi_3^2 = \tfrac{1}{4}(L^2 - H^2)^2.$$

So, for a perturbed Keplerian system in which the longitude h is ignorable, the reduction by the group SO(2) identifies the orbital space to the unique sphere in the Euclidean space based on the coordinates (ξ_1, ξ_2, ξ_3).

Once the Hamiltonian is expressed in these coordinates, the equations of motion are

$$\dot{\xi}_i = (\xi_i; \mathcal{H}) \qquad \text{for} \quad i = 1, 2, 3. \tag{7}$$

These equations cover all possible motions on the spheres, including those in the neighborhood of the north pole $(0, 0, \frac{1}{2}\sqrt{L^2 - H^2})$ and the south pole $(0, 0, -\frac{1}{2}\sqrt{L^2 - H^2})$. At the north pole, $G = L$, hence that point represents circular orbits with an inclination such that $\cos I = H/L$; at the south pole, since G is equal there to H, we find the class of equatorial orbits with eccentricity $e = \sqrt{1 - H^2/L^2}$. The coordinates (g, G) amount to a Mercator projection of the sphere, which excludes the north and south poles. Curves on a sphere are easy to draw on the screen of a terminal. In an orthographic projection, there is no need for a special algorithm to decide whether a point is visible on the screen.

5.2. Phase flows by numerical integration. In the main problem of artificial satellite theory, the global equations (7) tell that the south pole is an equilibrium, call it $E_\dagger$, and that this equilibrium is stable for any H. The north pole is also an equilibrium, call it E_0, but sometimes stable, at other times unstable. This change of stability stems from two bifurcations. These facts are established by using a Newton-Raphson iteration to solve analytically the equilibria equations

$$\partial\mathcal{H}/\partial g = 0, \qquad \partial\mathcal{H}/\partial G = 0 \tag{8}$$

in the neighborhood of $H = L/\sqrt{5}$. Details of the calculations have been published elsewhere [25]. This was accomplished with Macsyma on a Lisp workstation, and, later, Mathematica [26] on a Macintosh.

Once the equilibria were located and their stability character established, we drew a sample of averaged orbits by integrating numerically the differential equations

$$\dot{G} = \partial\mathcal{H}/\partial g, \qquad \dot{g} = -\partial\mathcal{H}/\partial G \tag{9}$$

Sampling was first accomplished at the terminal by selecting what we thought would be the appropriate initial conditions. After a long while, we came to an algorithm for selecting automatically those initial conditions that would produce the most telling image of the phase flow.

Our tool for doing the plots, which we named the "Doodler," operates somewhat like a hybrid of a plotting package and an object-oriented drawing program. Lines, curves, and figures can be drawn under program control. In addition, one can use the mouse to annotate or manipulate the images. For instance, one can superimpose the plots to make a collage of the kind that one sees in [25, Figures 2 and 5].

General purpose algebraic systems like Macsyma and Mathematica, on the one hand, and interactive graphic software like the Doodler proved effective in attacking more difficult systems. On account of the interest engineers have for "frozen orbits," we enlarged the main problem to involve a few more zonal harmonics, namely those of degree 4 and 6.

Save for a rotation by 90° the harmonic of degree 4 introduces no qualitative changes. The same, however, cannot be said for the harmonic of degree 6. Eliminating G between the equations (8) produces a resultant that is quadratic in $\sin g$, it opens the prospect of equilibria appearing at values of g away from multiples of $\pi/2$ whose appearance and annihilation are the effect of additional pitchfork bifurcations. At this stage of our analysis we began to feel very painfully how much we were straining the capabilities of our equipment. We had reached the point where a diagram of the phase flow produced by numerical integration required 60 uninterrupted hours on the Lisp workstation.

5.3. Phase flows by color painting. Investigation of the critical inclination in the main problem of satellite theory was initially limited to a neighborhood of $H/L = 1/\sqrt{5}$. As the value of H/L decreases, what becomes of the global dynamics? Answering the question analytically by iterating the Newton-Raphson procedure is out of the question. The result would be so complex as to preclude any significant interpretation. The only way then is to proceed numerically. The function `Solve` of Mathematica proves eminently suitable in that direction. To our surprise we found that, close to $H/L = 0.0306$ for $J_2\alpha^2/a^2 = 0.001$, there appear four more roots in G for equations (8) when g equals 0 or π.

This discovery made us wish to be able to see snapshots of the phase flow as the ratio H/L runs from $1/\sqrt{5}$ to 0. Numerical integration will not serve this purpose. Let us approach the problem from a different perspective. Since we are dealing with conservative Hamiltonians having only one degree of freedom, the average orbits are the level curves of the Hamiltonian over the orbital spheres. Initially, we experimented with the contouring features of Mathematica; we found them wanting. Not all interesting features could be shown simultaneously due to insensitivity to changes of scale. Furthermore, numerical roundoff gave poorly defined trajectories.

There is an alternative that solves all these problems and furthermore is ideally suited to the data-parallel architecture of the Connection Machine. At each point in the phase space we compute the value of the Hamiltonian; we convert that value into a color code; we assign that color code to the pixel representing the point. This is what we call "painting the Hamiltonian onto phase space." These computations are performed for each point independently of the computations for any other points, and thus they can proceed in parallel. On the screen, we see strips of different colors; the boundaries between adjoining strips are a substitute for contour levels of the Hamiltonian.

A straightforward painting of the Hamiltonian would suffer from the same problem as contour drawing in that it would be insensitive to changes in scale. The algorithm mapping value onto color must be selected carefully, for there is a difference between the map we seek and a cartographer's contour map. The cartographer wants to represent altitudes uniformly throughout the map. For us, however, comparison of altitudes is not the issue. The altitude of a point over the sphere is determined by the value of the Hamiltonian at that point. We are not interested in reading values of the Hamiltonian in the map; we only want to relate points if they are of the same value and to contrast them if they are not. In particular, we care about marking peaks, hollows, and passes; we are especially attentive to the contours around these singularities. These are the features that we want to emphasize even if it means losing all information about their relative heights.

The new equilibria just mentioned have a Hamiltonian value which is about 10^4 times smaller than that of the critical inclination points. In a uniform height scale, it would take an enormous number of colors (or, for a contour plot, an enormous number of contour lines) to enhance these very low features, so many in fact that the interesting areas would have only slight color gradation. To a cartographer, the problem is one of separating a huge mountain from a shallow lake at its foot, and furthermore of showing the details in the lake and around the summit. A cartographer might solve the problem by printing the heights of the mountain and the lake on the map. We do not care so much about the heights as we do about showing which points in the neighborhood of the mountain and the lake are of the same height and about contrasting points at the same elevation with nearby points at slightly different altitudes.

One way of achieving this result is by covering the range of Hamiltonian values not by a single spectrum, but by covering with several spectra. This will effectively show finer detail. The procedure, however, has its limits. Too many bands of color would make the picture confusing.

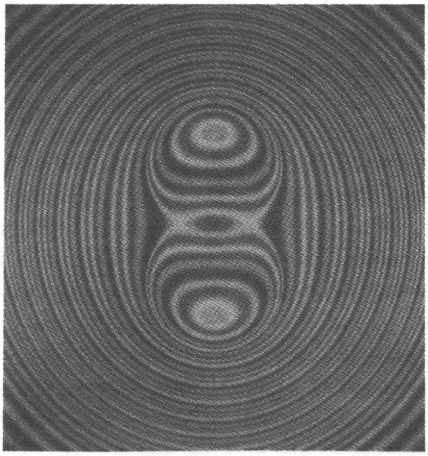

Figure 3: The phase flow slightly below $H/L = 1/\sqrt{5}$. The region depicted is an orthographic projection of the northern hemisphere upon the equatorial plane $\xi_3 = 0$. The north pole is at the center of the figure; the positive horizontal axis corresponds to $g = 0$.

Another way is to give up a uniform height scale. Whether shallow or high, the interesting points and their neighboring flows generally occupy about the same area in phase space. We draw a very fine scale around the stationary points, and take longer strides along the mountain sides and other less interesting areas. Thus we shall weigh the assignment of colors by the distribution of values, so that any strip of a given color has approximately the same number of points as any other strip. Our shallow lake, the newly discovered equilibria, occupies about the same area in phase space as the mountain of critical inclination, so it will now appear with as many color strips, even though the range of values of the Hamiltonian are dramatically different.

Calculating the Hamiltonian at each point and mapping its value to a color are operations that can be performed in parallel. On the Connection Machine (16K processors with floating point accelerator), the rendering of a 512×512 plot takes about one second; contrast this to the Symbolics where it takes about 40 minutes.

The first illustration of the painting technique, Figure 3, reproduces in black and white a color phase diagram on the display of the Connection Machine [27]. It shows the flow of the averaged equations in a neighborhood of the critical inclination after the two bifurcations which have alternatively changed the stability of the north pole from stable to unstable and back to stable again. Figure 4 shows the newly discovered equilibria. These equilibria appear by a saddle node bifurcation. The ability of the Connection Machine to display rapidly the phase flow as it evolves through the bifurcation is very effective in animating the mechanism by which these equilibria come to existence.

At the speed of the Connection Machine, one can make a movie of phase flow as the ratio H/L is varied. Pictures are produced fast enough that one's mind re-

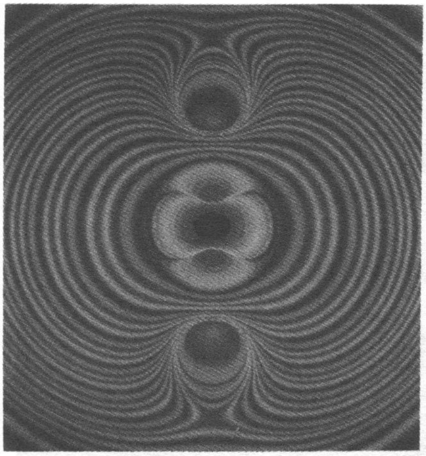

Figure 4: The eyes of the hippopotamus. A view from the south pole.

tains knowledge of the behavior, so that creative exploration with different values of parameters, or zooming in on some section of phase space yields a solid feel of the behavior, sometimes even new knowledge as well. In fact, a suspicious looking pattern at the previously unstable points of $g = \pi/2$ and $g = 3\pi/2$ at low values of H turned out to be, on closer examination, the aftermath of a pitchfork bifurcation into one stable and two unstable equilibria, the new unstable points being at slightly different values of g. This bifurcation had not been anticipated, although examination of the equilibria equations after the fact showed how they occur. Figure 5 is a zoom of the region around the equilibrium $g = \pi/2$ where the unanticipated bifurcation occurred.

6. Conclusions.

Most recently, visualization has been added to our collection of tools to provide insight into the global behavior of dynamical systems. The appeal of color graphics as an exploratory tool is not to be turned down. For it is at the start of a qualitative analysis rather than in the middle of writing the research report that global pictures should be sought.

Unlike other disciplines identified with pure geometry, nonlinear dynamics should not and never shall be one of those lofty towers where individuals feel compelled to live up to some intellectual asceticism away from machines and gadgets. The discipline instead must try with tenacity to keep pace with computational technology and make room for its innovations the same way. The challenge thus is endless, for each generation of mathematical physicists needs to keep abreast of techniques relentlessly emerging from the engineering shops. There is excusable dabbling and playfulness in shopping for a tool.

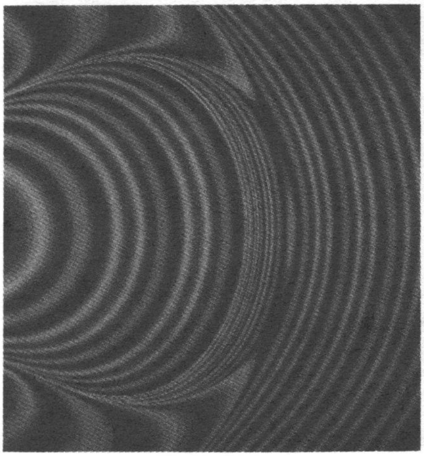

Figure 5: Zooming onto a surprise bifurcation in Figure 4.

References

[1] A. Deprit and A. Delie, *Trojan Orbits I. D'Alembert Series at L_4*, Icarus, **4** (1965), pp. 242–266.

[2] A. Deprit, *Limiting Orbits at the Equilateral Centers of Libration*, Astron. J., **71** (1966), pp. 77–87.

[3] A. Deprit, J. M. A. Danby and A. Rom, *The Symbolic Manipulation of Poisson Series*, Boeing Document D1–82–0481 (1965).

[4] J. M. Gerard, I. Iszak and M. P. Barnett, *Mechanization of Tedious Algebra: The Newcomb Operators of Planetary Theory*, Comm. ACM, **8** (1965), pp. 27–32.

[5] D. Barton, *A scheme for manipulative algebra on a computer*, Computer J., **9** (1967), pp. 340–344.

[6] A. Deprit, J. Henrard and A. Rom, *Trojan Orbits II. Birkhoff's Normalization*, Icarus, **6** (1967), pp. 381–406.

[7] A. Deprit and A. Rom, *The Main Problem of Artificial Satellite Theory for Small and Moderate Eccentricities*, Celest. Mech., **2** (1970), pp. 166–206.

[8] A. Deprit, J. Henrard and A. Rom, *Lunar Ephemeris: Delaunay's Theory revisited*, Science, **168** (1970), pp. 1569–1570.

[9] A. Rom, *Mechanized Algebraic Operations (MAO)*, Celest. Mech., **1** (1970), pp. 301–319.

[10] A. Deprit, *Canonical Transformations Depending on a Small Parameter*, Celest. Mech., **1** (1969), pp. 12–31.

[11] A. Deprit, *Dynamics of Orbiting Dust under Radiation Pressure*, The Big Bang and Georges Lemaître, ed A. Berger, The Reidel Publishing Company, Dordrecht, (1984), pp. 151–180.

[12] S. L. Coffey, A. Deprit and B. R. Miller, *The Quadratic Zeeman Effect in Moderately Strong Magnetic Fields*, Ann. New York Acad. Sc., **497** (1987), pp. 22–36.

[13] A. DEPRIT, S. FERRER, M.-J. MARCO AND J. PALACIÁN, *The Hydrogen Atom in Parallel Electric and Magnetic Fields (m=0)*, Physics B, submitted for publication.

[14] A. DEPRIT AND B. R. MILLER, *Normalization in the Face of Integrability*, Ann. New York Acad. Sc., **536** (1988), pp. 101–126.

[15] G.-I. HORI, *Theory of General Perturbations with Unspecified Canonical Variables*, Proc. Astron. Soc. Japan, **18** (1969), pp. 1287–1296.

[16] A. DEPRIT, *The Ideal Resonance Problem at First Order*, Advances in the Astronautical Sciences, **46** (1982), pp. 521–526.

[17] J. HENRARD AND P. WAUTHIER, *A Geometric Approach to the Ideal Resonance Problem*, Celest. Mech., **44** (1989), pp. 227–238..

[18] A. DEPRIT, *The Elimination of the Parallax in Satellite Theory*, Celest. Mech., **24** (1981), pp. 111–153.

[19] S. L. COFFEY AND A. DEPRIT, *A Third Order Solution to the Main Problem in Satellite Theory*, J. Guidance, Control, and Dynamics, **5** (1982), pp. 366–371.

[20] A. ROM, *Echeloned Series Processor (MAO)*, Celest. Mech., **3** (1971), pp. 331–345.

[21] R. R. DASENBROCK, *Algebraic Manipulation by Computer*, NRL Report **7564** (1973).

[22] ———————— *A FORTRAN-Based Program for Computerized Algebraic Manipulation*, NRL Report **8611** (1982).

[23] A. DEPRIT AND E. DEPRIT, *Massively Parallel Symbolic Computation*, Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation, ACM Press, New York, (1989), pp. 308–316.

[24] ———————— *Processing Poisson Series in Parallel*, J. Symbolic Computation, submitted for publication.

[25] S. L. COFFEY, A. DEPRIT AND B. R. MILLER, *The Critical Inclination in Artificial Satellite Theory*, Celest. Mech., **39** (1987), pp. 365-406.

[26] S. WOLFRAM, *Mathematica. A System for Doing Mathematics by Computer*, Addison-Wesley, Reading, MA, 1988.

[27] S. L. COFFEY, A. DEPRIT, E. DEPRIT AND L. HEALY, *Visualization of Phase Flows*, submitted for publication.