[12]Chow, S.N., Mallet-Paret, J., and Yorke, J.A., "Finding Zeros of Maps: Homotopy Methods that are Constructive with Probability One," *Math. Com.*, Vol. 32, 1978, pp. 887-889.

[13]Watson, L.T., and Fenner, D., "Chow-Yorke Algorithm for Fixed Points or Zeros of C^2 Maps," *ACM Trans. Math. Software*, Vol. 6, 1980, pp. 252-260.

[14]Businger, P. and Golub, G.H., "Linear Least Squares Solutions by Householder Transformations" *Numer. Math.*, Vol. 7, 1965, pp. 269-276.

[15]Shampine, L.F., and Gordon, M.K., *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W.H. Freeman, San Francisco, 1975.

[16]Watson, L.T., "Engineering Applications of the Chow-Yorke Algorithm," *Applied Mathematical Computations*, Vol. 9, 1981, pp. 111-133.

Rapid Poisson Series Evaluation

William H. Harr* and Bernard Kaufman†
U.S. Naval Research Laboratory, Washington, D.C.

Introduction

WITH the advent of machine automated algebraic manipulation programs, many problems which were formerly intractable because of the extreme amount of algebra required have now generated new interest. With many of these problems, a new complication has arisen in that it often occurs that series development in analytic or semianalytic methods becomes very large in the total number of terms required, which limits the practical application of these theories in a useful computer program. This paper outlines a method of optimizing the evaluation of large Poisson series in terms of speed, number of unique factors, and program storage.

Poisson Series Compression

A Poisson series can be represented symbolically in the form:

$$\sum_{i=1}^{n} a_i b_i c_i \tag{1}$$

where a_i are the numeric coefficients (M/N), b_i are the algebraic coefficients $(a^j b^k..)$, and c_i are the trigonometric terms $(\sin/\cos(A\alpha + B\beta..))$. If there are x unique numeric coefficients, y unique algebraic coefficients, and z unique trigonometric terms, then the maximum series length possible is $y \times z$. Two terms of the series with the same algebraic and trigonometric factors, but different numeric coefficients, are assumed to be combined by the distributive principle to form a single term. The maximum number of unique numeric coefficients (x) is then also $y \times z$.

Few series are composed of terms in which all three factors are unique to each term. Therefore, representing the series in Poisson form is duplicating information. A Poisson series of the form of Eq. (1) can be rewritten as

$$\sum_{i=1}^{n} d_i c_i, \quad \text{where} \quad d_i = a_i b_i \tag{2}$$

or as

$$\sum_{i=1}^{n} d_i b_i, \quad \text{where} \quad d_i = a_i c_i \tag{3}$$

If m is the total number of unique d_i, then this step will save $(n-m)$ multiplications. For the case $d_i = b_i c_i$, m will always equal n; that is, no work will be saved. The choice of which manner in which to combine factors will affect the final degree of optimization, but is dependent upon the particular series being optimized.

For series where each factor of each term is not unique in the entire series, the distributive principle can be used to sum all of the separate multipliers of the factor. For a series in the form of Eq. 2, in most cases each d_i is not unique to the ith term, i.e., $d_i = d_j$, for some $i \neq j$. In addition, the same is possibly true for the c_i. We can then rewrite the series in either of two forms:

$$\sum_{i=1}^{n'} e_i c_i, \quad \text{where} \quad e_i = \sum_{j=1}^{k} d_j \tag{4}$$

or as

$$\sum_{i=1}^{n'} d_i e_i, \quad \text{where} \quad e_i = \sum_{j=1}^{k} c_j \tag{5}$$

where n' is the new series length and is less than n. Assuming that the form $e_i c_i$ is used, the c_i factors are now unique to the series. This is not necessarily true for the e_i, however, so the distributive principle can be applied again. The series now have the form:

$$\sum_{i=1}^{n''} e_i f_i, \quad \text{where} \quad f_i = \sum_{j=1}^{\ell} c_j \tag{6}$$

and n'' is the series length and is less than n'.

The series are now composed of terms consisting of two factors, each factor being unique. The order in which the factors are summed (in this example, e_i c_i, and then e_i f_i, as opposed to d_i e_i and then f_i e_i) will affect the degree of series compression. Which order to use will depend upon the particular series being optimized. Normally the order of compression would be chosen to minimize the number of operations (not necessarily minimizing the series length). Figure 1 shows the possible orders of series compression. Note that the n's and n''s are not necessarily equal.

The sequence of evaluation of the series at this point is as follows: the numeric coefficients (a_i) are stored. At each point of evaluation, the algebraic and trigonometric terms (b_i and c_i) are calculated. The array of multiplicand pairs (d_i) are then calculated. The two series of sums (e_i and f_i) are

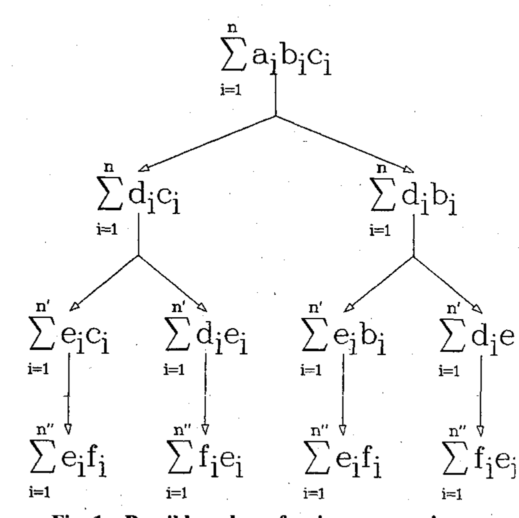

Fig. 1 Possible orders of series compression.

Presented as Paper 82-1432 at the AIAA/AAS Astrodynamics Conference, San Diego, Calif., Aug. 9-11, 1982; received Aug. 25, 1982; revision received Nov. 4, 1983. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Research Physicist, Aerospace Systems Division. Member AIAA.
†Supervisory Aerospace Engineer, Aerospace Systems Division. Associate Fellow AIAA.

then calculated, and, finally, the series themselves are calculated by the products of e_i*f_i. If several series are being calculated, the a_i, b_i, c_i, d_i, e_i, and f_i terms can be grouped and calculated for all of them. Only the final summation of the e_i*f_i need be separate for each series.

Trigonometric Term Evaluation

The computer evaluation of trigonometric functions is much slower than that for the simple arithmetic functions. For Poisson series, this problem can be alleviated by substituting trigonometric identities for system calls to sine and cosine so that only trigonometric functions of the individual variables need be made. At best, every sine or cosine can be replaced by two multiplications and one addition. In most cases, intermediate terms will need to be evaluated. This method of evaluating poisson series trigonometric terms by computer was first discussed by Shelus and Jefferys.[1] A near-optimal method has been developed by Coffey and Deprit.[2] Either of these methods has can be employed with the technique described above.

Example

To indicate the power of this technique, the authors will apply it to the problem of accurate prediction of satellite motion. Semianalytic solutions to this problem have been used by several investigators. References 3, 4, and 5 are typical of a few. Reference 3 explains in detail a semianalytic theory which uses the method of averaging to obtain very large time step integration of the averaged equations of motion for the satellite. This yields a "mean" ephemeris including perturbations due to the third body, zonals, and tesserals (if desired). Reference 6 then details the recovery of the selected short-period terms to obtain more accurate position and velocity.

As obtained with this method, the short period terms in the form of Poisson brackets for the third-body (sun and moon) terms were six series totalling 24,481 terms, including 48,962 multiplications, 24,481 additions, and 6050 sine/cosine calls. The series were generated using Dasenbrock's algebraic manipulator.[7] Each series was output in FORTRAN compilable form to a text file as a separate subroutine ordered by the trigonometric arguments, allowing the compiler to optimize the evaluation of the arguments. The chance for programming error is thus minimized and rapid proof of the theory and of the accuracy of the expansion is allowed. Steps were then taken to optimize the execution. Table 1 is a summary of the steps and results of the optimization.

Each series was first rewritten in the form of Eq. (1). For these series, there were 1335 unique numeric coefficients, 35 unique algebraic coefficients, and 4520 unique trigonometric terms, instead of the original 6050 trigonometric terms. These unique terms were stored in arrays, and pointer arrays were used to index the term arrays. This step resulted in a 31% increase in speed, due largely to the reduced number of trigonometric calls.

For the next step, analysis determined that of the 46,725 possible combinations of numeric and algebraic coefficients, only 2622 unique pairs occurred in the series, as opposed to 21,341 numeric-trigonometric pairs. The series were now in the form of Eq. (2), where the d_i's are the 2622 numeric-algebraic pairs. This step saved 21,859 multiplications for a further 17.8% speed increase.

The next step was the application of the distributive principle. Common multipliers of the same trigonometric factor were first grouped together. Of the 6050 unique sums possible, there were only 1438 in the series. The series were now in the form of Eq. (4). The distributive principle was used again to collect common trigonometric multipliers of the numeric-algebraic sums. The series are now in the form of Eq. (6). An additional 13,535 additions and 22,873 multiplications have been saved, for a further 68% speed increase.

Because the trigonometric arguments for these series were in a relatively simple form, the straightforward method of Shelus and Jefferys[1] was used to replace the 4520 calls to sine and cosine with identities. An additional 446 intermediate terms were calculated. This step resulted in a further 63% speed increase, for a total speed increase of 820%.

Table 1 Optimization summary

Step/process	Number of terms	Number of *	Number of +/−	Number of sin/cos calls	CPU time, s
Initial version	24,481	48,962	24,481	6050	1.6
Remove duplicate sin/cos calls	24,481	48,962	24,481	4520	1.095
Remove ALG-NUM pairs	24,481	27,103	24,481	4520	0.9
Distributive principle	1742	4364	10.946	4520	0.535
Trig identities	1742	14,272	15,900	12	0.195

Table 2 Number of resulting terms for each order of compression

Method	N	A	B	C	D	E	F	*	+	N[11]
Third body										
DC:EC:EF	24481	1335	35	4520	2622	1784	1561	4675	15426	2053
DC:DE:FE	24481	1335	34	4520	2622	1438	1588	4364	10946	1742[a]
DB:EB:EF	24481	1335	35	4520	21341	?	?	?	?	?
DB:DE:FE	24481	1335	35	4520	21341	?	?	?	?	?
J_2										
DC:EC:EF	844	438	80	88	717	110	88	835	765	118
DC:DE:FE	844	438	80	88	717	105	130	852	822	135
DB:EB:EF	844	438	80	88	745	124	80	869	835	124
DB:DE:FE	844	438	80	88	745	107	175	920	787	175

[a]Method used in example

Table 2 shows the results of the different possible methods of compression for the third-body series and for the J_2 series. Compression was not continued for the third body methods that generated 21,341 unique d's, as this automatically requires 21,341 multiplications, a far larger number than the other two methods. Note that for the J_2 series, a case occurs with a shorter final series length but a larger number of multiplications and additions. Note also that the smallest number of multiplications and additions requires the route DC:DF:FE for the third-body series, but requires the route DC:EC:EF for the J_2 series.

Conclusion

Which route of compression to take depends upon the particular series being compressed and upon the relative speeds of execution of multiplication and addition of the target computer. Also, there is normally no a priori method for knowing which of the four routes is optimal for a particular series, so all four should be tried. For these reasons, this technique is not intended to be used as normal output from an algebraic manipulator package, but rather as a step taken when implementing a theory in an application where run time is critical. Not every Poisson series will display the degree of compression found in this example. It is probable that every series will benefit to some extent. In particular, series generated using expansions are highly likely to compress by a significant amount. In addition, any series would benefit by the use of trigonometric identities, rather than system calls.

References

[1] Shelus, P.J. and Jefferys, W.H., "A Note on an Attempt at More Efficient Poisson Series Evaluation," *Celestial Mechanics,* Vol. 11, 1975, pp. 75-78.

[2] Coffey, S. and Deprit, A., "Fast Evaluation of Fourier Series," *Journal of Astronomics and Astrophysics,* Vol. 81, 1980, pp. 310-315.

[3] Kaufman, B. and Dasenbrock, R.R., "Semianalytic Satellite Theory for Long-Term Behavior of Earth and Lunar Orbiters," *Journal of Spacecraft and Rockets,* Vol. 10, June 1973, pp. 377-383.

[4] Cefola, P., Green, A., McCalin, W., Early, L., Proulx, R., and Taylor, S., "Semianalytic Satellite Theory – Application to Orbit Determination," AIAA Paper 80-1677, Aug. 1980.

[5] Gooding, R.H., "On the Generation of Satellite Position (and Velocity) by a Mixed Analytical-Numerical Procedure," *Advanced Space Research,* Vol. 1, 1981, pp. 83-93.

[6] Kaufman, B., "First Order Semianalytic Satellite Theory with Recovery of the Short Period Terms Due to Third Body and Zonal Perturbations," *Acta Astronautica,* Vol. 8, 1981, pp. 611-623.

[7] Dasenbrock, R.R., "A FORTRAN-Based Program for Computerized Algebraic Manipulation," Naval Research Laboratory, Washington, D.C., Rept. 8611, Sept. 1982.

Vertical Ascent to Geosynchronous Orbit

James A. Martin*
NASA Langley Research Center, Hampton, Virginia

Introduction

THE Space Shuttle was developed to provide transportation from Earth to low-Earth orbit (LEO), with transportation to geosynchronous Earth orbit (GEO) by orbit transfer vehicles. Since the final destination of many payloads is GEO, there is some incentive to consider vehicles that are not constrained to go to LEO on the way. Because a destination in GEO is always directly above a point on the equator, it is possible to ascend by going straight up. This approach is called "vertical ascent."

The purpose of this study was to examine vertical ascent to GEO and compare it to the more conventional approach. Trajectories were calculated using the Program to Optimize Simulated Trajectories (POST).[1] The chemical propulsion used represented a Space Shuttle Main Engine (SSME) with a two-position nozzle. At sea level, with an expansion ratio of 40, this engine provides 2.23 MN thrust with an exit area of 2.15 m^2 and a vacuum specific impulse of 446 s. Above 10 km, the nozzle was extended to an expansion ratio of 150, and the engine provided 2.32 MN thrust with an exit area of 8.06 m^2 and a vacuum specific impulse of 463 s.

Received Feb. 9, 1983; revision received Sept. 8, 1983. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Aerospace Engineer, Space Systems Division. Senior Member AIAA.

Vertical Ascent Trajectory

A vertical ascent trajectory is illustrated in Fig. 1 for a case with all chemical rocket propulsion. The initial acceleration in multiples of Earth's gravity a_0 is 1.3, and the maximum a_m is 3.0. In the calculations the radial velocity was approximately 20 m/s at GEO.

The thrusting strategy is illustrated in Fig. 2 for a gross mass of 1.5 Gg. As the vehicle rises, the thrust increases as the atmospheric pressure falls. At an altitude of 10 km, the nozzle exension was deployed. At about 70 km, the thrust has reached the vacuum thrust level and is nearly constant. As the vehicle mass is reduced by propellant consumption, the acceleration reaches the selected maximum. After this point, the thrust is reduced continuously such that the acceleration remains at a_m.

The thrust angle θ is vertical initially and decreases slightly as the vehicle rises. The eastward thrust component accelerates the vehicle just enough to keep the vehicle above the launch site. At some point in the ascent, the vertical thrust component is no longer needed. For the case shown, this point is at an altitude of 1662 km. Above this altitude, θ is zero. Since only a small eastward thrust is required, the thrust level drops at this point.

For the vertical ascent with $a_0 = 1.3$ and $a_m = 3.0$, the total ideal velocity increment (ΔV) is 17.7 km/s. Of the total, only about 3.5 km/s are required in the dense atmosphere, which ends at about 100 km.

Comparison with Hohmann Ascent

To provide a basis for comparison for the vertical ascent, Hohmann ascent trajectories were calculated with the same propulsion characteristics. The insertion perigee was 92.6 km. Several values of a_0 and a_m were considered for both Hohmann and vertical ascent. The results (see Fig. 3) indicate that the Hohmann ascent requires a significantly lower ΔV, about 12.8 km/s rather than 17.7 km/s for vertical ascent.

Advanced Propulsion

If chemical rockets are used for the initial part of the vertical ascent and an advanced propulsion system, such as a laser system, is used for the remainder, the performance may be improved because of the higher specific impulse of the advanced system. Laser propulsion may be achieved more easily with a vertical ascent because the laser beam could always be pointed along the path from the launch site to the destination. In Fig. 4, some estimates for the ratio of the initial mass to the final mass R are shown to provide some insight into the potential of vertical ascent with advanced propulsion. The results are based on ideal rocket calculations.

The lowest curve in Fig. 4 shows how R increases with ΔV for a chemical rocket. A mass ratio of 17 corresponds to the ΔV for a Hohmann ascent of 12.8 km/s. The vertical ascent mass ratio is off the scale at 50 for a ΔV of 17.7 km/s with