A. J. Surkan

Symbolic Polynomial Operations with APL

Abstract: A recursive calculation of polynomial coefficients is used to demonstrate how functions performing polynomial algebra and differentiation may be written concisely in Iverson notation and executed by an on-line, time-sharing implementation of APL. These functions will operate on symbolic polynomials with constant coefficients and display in a superscripted format results having up to 116 variables.

Introduction

This note demonstrates by example the capability and practicality of the APL[1] language for manipulating and deriving, with simultaneous on-line typewriter display, symbolic polynomial expressions in familiar mathematical form. The particular classical example of a point satellite orbiting about some center of mass is motivated by the growing requirement for computational tools yielding more precise results in celestial mechanics and by the increasing simplicity with which high-level computer languages can obtain and evaluate expressions (dating back to the time of Lagrange[2]) that are otherwise unmanageable. This example includes both differentiation and recursion; polynomial operators are employed to perform the addition, subtraction, multiplication, and differentiation required by the computation algorithm.

Example: orbit calculation

The expression for the position, **R**, of a body in slightly perturbed Keplerian motion can be expanded in a Taylor series in terms of initial position, $\mathbf{R}_0$, and velocity, $\dot{\mathbf{R}}_0$, at time t_0 in the following form[2]:

$$\mathbf{R} = F\,\mathbf{R}_0 + G\,\dot{\mathbf{R}}_0 + \cdots . \tag{1}$$

For nearly circular orbits the first two terms are often sufficient. The time-dependent functions F and G are expressible as power series in terms of the elapsed time, $t - t_0$. Each term is a member of a recursively calculated series:

$$F = \sum_{N=0}^{\infty} F_N \frac{(t-t_0)^N}{N!}, \quad G = \sum_{N=0}^{\infty} G_N \frac{(t-t_0)^N}{N!}, \tag{2}$$

where

$$F_N = \dot{F}_{N-1} - UG_{N-1}, \qquad G_N = F_{N-1} + \dot{G}_{N-1}. \tag{3}$$

In these equations X, Y, and Z are the Cartesian coordinates of the body,

$$R^2 = X^2 + Y^2 + Z^2 \quad \text{and} \quad U = 1/R^3.$$

We also define

$$S = X\dot{X} + Y\dot{Y} + Z\dot{Z}, \tag{4}$$

$$\dot{S} = \dot{X}^2 + \dot{Y}^2 + \dot{Z}^2 - (1/R),$$

$$V = S/R^2 \quad \text{and} \quad W = \dot{S}/R^2. \tag{5}$$

The F and G series can be calculated recursively by applying one level of the chain rule for differentiation to obtain the time derivatives of the parameters U, V, and W. Explicitly these are: $\dot{U} = -3UV$, $\dot{V} = 2 - 2V$, $\dot{W} = -V(U - 2W)$. This procedure yields

$$F_N = \dot{U}\frac{\partial F_{N-1}}{\partial U} + \dot{V}\frac{\partial F_{N-1}}{\partial V} + \dot{W}\frac{\partial F_{N-1}}{\partial W} - UG_{N-1}, \tag{6}$$

$$G_N = \dot{U}\frac{\partial G_{N-1}}{\partial U} + \dot{V}\frac{\partial G_{N-1}}{\partial V} + \dot{W}\frac{\partial G_{N-1}}{\partial W} + F_{N-1},$$

where $F_0 = 1$ and $G_0 = 0$.

The F and G series of coefficients are simply polynomials in U, V, and W. Thus it is sufficient to write four dyadic functions that will symbolically add, subtract, multiply, and differentiate polynomials. As an illustration, the "add" function *PLS* for the orbit calculation is shown in Fig. 1. This function adds polynomials by simple catenation and groups like terms into a single term, finding the net coefficient by summation.[3] For efficient operation the

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

```
    ∇ R←V PLS B;M
[1]   CO←(M←(ρV)ρ1,NVρ0)/V←V,B
[2]   EX←(~M)/V
[3]   EX←((NT←(ρV)÷NV+1),NV)ρEX
[4]   NM←0ρRW←K←1
[5]   CLCT1:NM←NM,(NVρ100)⊥EX[RW;]
[6]   →(NT≥RW←RW+1)/CLCT1
[7]   CLCT2:SM←+/(M←NM[K]=NM)/CO
[8]   CO[M/⍳NT]←SM,(¯1++/M)ρ0
[9]   →(NT≥K←K+1)/CLCT2
[10]  EX←(M←0≠CO)/[1]EX
[11]  CO←M/CO
[12]  EX←(0,NVρ1)\EX
[13]  EX[;1]←CO
[14]  REX←1ρρR←,EX
[15]  R←R,((NV+1)×REX=0)ρ0
    ∇
```

Figure 1 The *PLS* function which adds polynomials by catenation and groups together like terms with coefficient summation. (See Ref. 3 for an introductory description of the notation.)

```
F 11 =
                9                 7                    2 7
+ 654729075 UV  - 1240539300 UV W - 413513100 U V

                5 2               2 5                  3 5
+ 766215450 UV W  + 465404940 U V W + 54864810 U V

                3 3               2 3 2                3 3
- 170270100 UV W  - 137837700 U V W  - 27027000 U V W

              4 3               4              2   3
- 1201200 U V  + 9823275 UVW  + 9058500 U VW

              3   2              4         5
+ 2023758 U VW  + 114444 U VW + 1023 U V
G 11 =
                8                6                   2 6
- 310134825 UV  + 510810300 UV W + 132432300 U V

                4 2               2 4                3 4
- 255405150 UV W  - 113513400 U V W - 9459450 U V

               2 3              2 2 2               3 2
+ 39293100 UV W  + 21116700 U V W  + 2589840 U V W

           4 2             4            2 3           3 2
+ 63360 U V  - 893025 UW  - 457200 U W  - 50166 U W

          4        5
- 1008 U W - 1 U
```

Figure 3 Part of the displayed results obtained with symbolic polynomial operators in the orbit calculation.

Figure 2 Main program for recursive calculation of polynomial coefficients.

```
    ∇ ORBIT
[1]   E← ¯1 1 1 0 ¯2 0 1 ,1ρS← 1 0 0 1 ¯2 0 2 0 ,0ρM← ¯3 1 1 ,1ρG←2=F←(T←1),(NV←3)ρ0
[2]   SEE E←(M TMS 1 1 DRV F)PLS(S TMS 1 2 DRV F),(E TMS 1 3 DRV F), ¯1 1 0 0 TMS G,0ρ⎕←'F ',(N T),' = '
[3]   SEE G←F PLS(M TMS 1 1 DRV G),(S TMS 1 2 DRV G),E TMS 1 3 DRV G,0ρ⎕←'G ',(N T),' = '
[4]   →(30≥T←T+1)/2,0ρF←E
    ∇
```

auxiliary functions should be designed to meet the needs of the particular calculation. A computer program, using four such functions in APL for the orbit calculation, has been written in four lines to derive symbolic expressions for subsequent terms of the *F* and *G* series. Previously this type of symbolic calculation was performed with the more difficult-to-use batch-processing languages such as FORMAC,[4] ALPAK,[5] and PM.[6]

The APL algorithm is shown in Fig. 2. Note that the parameter polynomials are entered in the first line. Each term of a polynomial is represented by a numeric coefficient and an exponent vector. For example, $CU^{N_1}V^{N_2}W^{N_3}$ is represented by C, N_1, N_2, N_3. The subsequent terms of the *F* and *G* series are derived and displayed with conventional mathematical superscript notation using the functions *PLS*, *MNS*, *TMS*, *DRV*, *N̲*, and *SEE* of lines 2 and 3. The first five functions respectively add, subtract, multiply, differentiate, and convert numbers to characters, while the sixth function provides the output display format of the superscripted variables with their coefficients. The last line provides temporary storage for the preceding member of the F series as well as a termination at the required N value. On an IBM System/360 Model 50 accessed through an on-line APL terminal the simple four-line program derived in twenty minutes the first 19 terms, i.e., terms through $N = 19$, of the F and G series. A sample of the output, the terms with $N = 11$, is shown in Fig. 3. A simple extension of the program could be used to evaluate these general results for any set of values of U, V, and W.

References

1. K. E. Iverson, *A Programming Language,* John Wiley and Sons, Inc., New York (1962); A. D. Falkoff and K. E. Iverson, *APL/360 Users' Manual,* IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y., August 1968.
2. J. L. Lagrange, *Ouvres* **IV**, 500 (1869).
3. A. D. Falkoff and K. E. Iverson, "The APL/360 Terminal System," *Research Report RC 1922,* IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1967; this report provides a brief introduction to APL expressions and their relation to mathematics.
4. P. Sconzo, A. R. LeSchack, and R. Tobey, *Astron J.* **70,** 269 (1965).
5. W. S. Brown, *Bell System Tech. J.* **42,** 2031 (1963).
6. G. E. Collins and J. H. Griesmer, *Research Note NC 641,* IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., 1966.

Received August 20, 1968