POSTĘPY ASTRONOMII
Tom XXVI (1978). Zeszyt 1

PROGRAMY KOMPUTEROWE DO SYMBOLICZNYCH PRZEKSZTAŁCEŃ ALGEBRAICZNYCH

Część II

ZASTOSOWANIA I PERSPEKTYWY

ANDRZEJ KRASIŃSKI

Centrum Astronomiczne im. Mikołaja Kopernika, PAN (Warszawa)

MAREK PERKOWSKI

Instytut Automatyki Politechniki Warszawskiej (Warszawa)

СИМВОЛИЧЕСКИЕ АЛГЕБРАИЧЕСКИЕ ПРОГРАММЫ ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

Часть II

ПРИМЕНЕНИЕ И ПЕРСПЕКТИВЫ

А. Красиньски, М. Пэрковски

Содержание

Статья кратко описывает некоторые виды применения алгебраических программ в математике, физике и астрономии. Более подробно рассматриваются программы для символического интегрирования, общей теории относительности, квантовой электродинамики и небеской механики. Упоминаются некоторые другие виды применения. Приводятся также ссылки на литературу (это только избранная, а не полная библиография). Дискутируются возможные применения таких программ в будущем. В статье представлена основная идея и некоторые черты общих систем алгебраической манипуляции и обрассматривается возможность запрограмирования новой системы такого типа в Польше. Кратко представлено влияние алгебраических программ на математику и науку об искусственном интеллекте. Сравниваются некоторые критерии оценки полезности и эффективности разных программ и системов. Наконец перечисляются некоторые большие алгебраические системы общего назначения и рассматриваются перспективы их введения в Польше.

SYMBOLIC ALGEBRAIC COMPUTER PROGRAMS

Part II

APPLICATIONS AND PERSPECTIVES

Abstract

Examples of definite applications of algebraic programs made in mathematics, physics and astronomy are briefly described. The topics treated in more detail are: symbolic integration, general relativity, quantum electrodynamics and celestial mechanics. Programs dealing with some more topics are listed from the literature (this is a small selection of references, not a bibliography). Possible future applications of such programs are discussed. The underlying idea and some capabilities of general purpose algebraic manipulation systems are discussed and possibilities of creating new ones in Poland are considered. The impact of algebraic programs on matematics and the science of artificial intelligence is briefly presented. A comparison of criteria of usefullness and efficiency of various programs and systems is given. Finally, a few large general purpose systems for algebraic manipulation are mentioned and the perspectives of their use in Poland are discussed.

---

4. PRZYKŁADY ZASTOSOWAŃ PROGRAMÓW ALGEBRAICZNYCH

W poprzedniej części naszego artykułu podaliśmy jedno tylko uzasadnienie używania komputerów do obliczeń algebraicznych: zdjęcie z człowieka ciężkiej, bezmyślnej pracy wymagającej dużej niezawodności rachunkowej. Istnieje jednak jeszcze kilka innych typowych sytuacji, w których komputery wykonujące obliczenia algebraiczne bywają bardzo użyteczne lub nawet niezastąpione (Roskies 1974):

1) Sprawdzenie długich i skomplikowanych rachunków wykonanych uprzednio przez człowieka

2) Wykonanie rachunków tak skomplikowanych, że żaden człowiek nie zechciałby robić ich ręcznie

3) Sprawdzenie wyników programu numerycznego. Program numeryczny operuje na danych liczbowych od samego początku, kumulując błędy rachunkowe, które następnie trzeba szacować rozmaitymi metodami. Program algebraiczny otrzymuje wynik w sposób analityczny i podstawia dane liczbowe dopiero w ostatnim kroku, co umożliwia znacznie pewniejszą ocenę błędu

4) Sprawdzenie nietrywialnych przypadków szczególnych twierdzeń, które zamierzamy dopiero udowodnić. Bez takiego sprawdzenia hipotetyczna teza twierdzenia jest zbyt niepewna, aby warto było wkładać znaczniejszy wysiłek w jej dowodzenie. Sprawdzanie „ręczne" bywa zaś bardzo pracochłonne.

Programy algebraiczne są obecnie stosowane w wielu dziedzinach nauki i techniki. Autorom wiadomo o programach zastosowanych w teorii względności (Barton i Fitch 1971;

Payne 1972 i 1976; d'Inverno 1975; Cohen i in. 1976), teorii cząstek elementarnych (Campbell 1970 i cytowane przez niego prace), elektrodynamice kwantowej (Hearn 1966 i 1971; Roskies 1974), renormalizacji różnych teorii pola (Calmet i Perrottet 1971), mechanice kwantowej (Boys i in. 1956 – jeden z pierwszych historycznie programów algebraicznych; Fletcher i Reeves 1963; Reeves 1966), fizyce statystycznej (Mc Cormack 1975), fizyce plazmy (Bers i in. 1976 – w pracy tej autorzy cytują kilkanaście wcześniejszych zastosowań), krystalografii (Brugger 1965), geofizyce (Nylan i Chapman w materiałach wydanych przez Petricka 1971), mechanice nieba (Jeffreys 1971, przykłady omówione w osobnym paragrafie), analizie i syntezie układów elektrycznych (Goldberg 1959) i elektronicznych (Neidelman 1967), chemii kwantowej (Wactlar i Barnett 1964; Schnupp 1968; Heldmann i Schnupp 1968), teorii sterowania (wg Tobeya 1966 i Sammet 1966 i 1968), teorii optymalizacji (Stoutenmeyer 1975), teorii gier (wg Tobeya 1966), ekonometrii (Tobey 1966), oraz o programach ogólnego przeznaczenia, stanowiących w zasadzie języki specjalne przystosowane do łatwego programowania obliczeń algebraicznych.

Lista wykorzystanych zastosowań jest jednak znacznie większa, na świecie (głównie w USA) pracuje się bowiem nad tą dziedziną bardzo intensywnie. Liczba prac sięga już setek. Niestety, większość najcenniejszych pozycji (zwłaszcza amerykańskich raportów zawierających opisy i listingi programów) jest niedostępna w Polsce. Przeglądy programów algebraicznych i ich zastosowań zawierają prace Sammet (1966), Tobeya (1966), Campbella (1970), Hearna (1971), Bartona i Fitcha (1972a,b) oraz materiały konferencji wydane przez Bobrowa (1968) i Petricka (1971).

W niniejszym artykule ograniczymy się do omówienia wybranych przykładów zastosowań „algebry komputerowej", mając nadzieję, że na tej podstawie astronomowie sami dostrzegą nowe potrzeby i możliwości.

I. ZASTOSOWANIA W MATEMATYCE

Największa liczba publikowanych prac związanych z rachunkami algebraicznymi na komputerach dotyczy obliczania całek (Moses 1971a). Jest to problem trudny z tego powodu, iż jest z samej natury rzeczy niealgorytmiczny. Nie istnieje, jak w przypadku różniczkowania, jednoznaczny operacyjny przepis na całkowanie. Całka ze znanej funkcji może być nową, nieznaną dotąd funkcją, zdefiniowaną przez tę właśnie całkę. Dlatego nie ma gwarancji, że dowolna całka da się w ogóle obliczyć symbolicznie. Poszukiwanie metod całkowania możliwie najszerszych klas funkcji jest głównym przedmiotem współzawodnictwa programistów w nowej dziedzinie wiedzy, powstającej na pograniczu matematyki i badań nad sztuczną inteligencją. Prace te doprowadziły już do skutecznych programów całkujących, nadających się do praktycznego użytku. Stosują one dość zaawansowane metody, jak np. analiza heurystyczna celów i środków służących do ich osiągania (Ernst i Newell 1969), dekompozycja i planowanie procesu całkowania (Slagle 1963), rozkład na ułamki proste i całkowanie funkcji wymiernych (Moses 1971a), całkowanie metodami Rischa* (Moses 1971a), metoda całek konturowych i residuów (Campbell i in. 1976).

*Risch (1970) opracował silne algorytmy całkowania wyrażeń należących do pewnych klas funkcji zespolonych.

Wiele publikacji dotyczy też działań na wielomianach (Abramov 1974) oraz upraszczania algebraicznego wyrażeń i sprowadzania do postaci normalnych (Moses 1971b; Hearn 1976). Część zagadnień z tych grup (a mianowicie operacje algorytmiczne) można by uznać za rozwiązane, ponieważ istnieją już programy wykonujące szereg operacji na wielomianach oraz bardzo skuteczne programy upraszczające. Ich agorytmy mogą być jednak konstruowane na wiele sposobów – trwają więc poszukiwania najefektywniejszych algorytmów, zapewniających minimalny czas działania programu i minimalne obciążenie pamięci maszyny. Ponadto z zagadnieniami tymi łączą się ściśle zagadnienia matematycznie nierozstrzygalne, dla których trzeba poszukiwać metod heurystycznych (patrz Moses 1971b), lub zagadnienia jeszcze nie rozwiązane. Należą do nich problemy istnienia postaci normalnej i sprowadzania do niej dla pewnych klas funkcji (Moses 1971b), faktoryzacji i upraszczania wielomianów wielu zmiennych (Jordan i in. 1966; Collins 1971) i inne.

Do problemów w pełni rozwiązanych teoretycznie i efektywnie zaprogramowanych można zaliczyć np.: różniczkowanie, rozwinięcia na szeregi Taylora, działania algebraiczne na liczbach zespolonych, macierzach i tensorach, obliczanie kolejnych wyrazów szeregów danych wzorami rekurencyjnymi, podstawianie. Wszystkie te metody zostały zaprogramowane w systemie MACSYMA (Martin i Fateman 1971; Bers i in. 1976).

W toku są prace nad szeregami Fouriera, transformatami Laplace'a, Fouriera i Z* (cytowane za Sammet 1968), a także obliczaniem granic (Laurent 1973) i rozwiązywaniem układów równań algebraicznych wyższych stopni (Moses 1966).

Osobną grupę zagadnień stanowi rozwiązywanie równań różniczkowych. Istnieją programy do rozwiązywania równań określonych typów (Tobey 1966; Moses 1971a), programy wykorzystujące ogólne metody asymptotyczne (Rink i Guru 1976) i metody stosujące rozwinięcia na szeregi Taylora (Ball i Berns 1966), a także programy konwersacyjne (Farlow 1976).

Kilka innych zastosowań w matematyce omawia artykuł Tobeya (1966).

II. ZASTOSOWANIA W FIZYCE

W fizyce programy algebraiczne są najbardziej intensywnie wykorzystywane w teorii względności i elektrodynamice kwantowej. W teorii względności stosuje się je głównie do obliczania tensorów krzywizny (tzn. współczynników koneksji oraz tensorów Riemanna, Ricciego, Einsteina i Weyla). Jest to praca nadzwyczaj ciężka i nudna, której wykonanie, wraz ze sprawdzeniem wyników, w trudnych przypadkach zajmuje kilka miesięcy nawet wprawnemu rachmistrzowi. Komputer potrafi wykonać ten sam rachunek w kilka minut, i to bez błędów (jeśli nie było błędu w programie...). Zatem nawet jeśli ktoś musi najpierw samodzielnie napisać program, włożony w to czas zwraca się już po dwukrotnym jego użyciu. Polecamy to uwadze wszystkich zainteresowanych nowymi zastosowaniami.

Z technicznego punktu widzenia liczenie tensorów krzywizny wymaga użycia algorytmów działań algebraicznych i różniczkowania w zastosowaniu do szerokiej klasy skomplikowanych wyrażeń funkcyjnych oraz obliczania wyznaczników i macierzy odwrotnych do danych, wszystko to z równoczesnym bardzo skutecznym upraszczaniem wyników. Programy te wymagają też niekiedy wykonywania działań algebraicznych i różniczkowych na wielomianach i

*Transformaty Z są stosowane m.in. w teorii sterowania do rozwiązywania równań różnicowych.

funkcjach wymiernych. Dla potrzeb teorii względności powstały też programy wykonujące rachunki przybliżone, tzn. zaniedbujące potęgi pewnego małego parametru wyższe niż zadeklarowany rząd przybliżenia (F r i c k 1977) oraz programy badające typ algebraiczny tensora Weyla (B a r t o n i F i t c h 1972a; d'I n v e r n o 1975). Drugi z tych problemów wymaga rozwiązywania równań algebraicznych do czwartego stopnia włącznie (a ściślej: badania koincydencji ich pierwiastków), przy czym współczynniki równań są złożonymi wyrażeniami funkcyjnymi. Ostatnio powstają również programy „drugiej generacji", które same potrafią ułożyć algorytm obliczeń w oparciu o skondensowany zapis tensorowy (np. program taki „rozumie" zapis $R^i_{k;i}$, gdzie R^i_k jest danym tensorem, i w oparciu o ten zapis wylicza $R^i_{k;i} = R^i_{k,i} + \Gamma^i_{si} R^s_k - \Gamma^s_{ki} R^i_s$, wykonując wszystkie potrzebne sumowania, por. H ö r n e f e l d t 1977).

W elektrodynamice kwantowej programy algebraiczne wykorzystuje się do manipulacji na macierzach gamma przy liczeniu grafów Feynmana dla różnych procesów. Rachunki te wymagają od programu niewielkiego zakresu umiejętności (dodawanie, mnożenie, różniczkowanie i porządkowanie wielomianów oraz obliczanie kilku typów całek), połączonego jednak z możliwością przekształcania wielkich wyrażeń (sumy kilkudziesięciu tysięcy składników). W tej właśnie dziedzinie zastosowań istnieje szczególnie efektowny przykład użyteczności programów algebraicznych (R o s k i e s 1974). Wyliczenie momentu magnetycznego elektronu z dokładnością do szóstego rzędu wymaga obliczenia 40 grafów Feynmana. Obliczenie numeryczne jednego grafu trwa 20–30 godz., zaś błąd numeryczny w szóstym rzędzie przybliżenia wynosi ok. 5%. L e v i n e i R o s k i e s opracowali program algebraiczny (w języku ASHMEDAI, który sami stworzyli), obliczający jeden graf w ciągu 2 min. – analitycznie, a więc z dowolną dokładnością. Jeden z systemów obliczeń symbolicznych, wykorzystywanych w elektrodynamice kwantowej, SCHOONSHIP (S t r u b b e 1974) jest dostępny w systemie CYFRONET. Nie mamy jednak informacji o tym, czy był w Polsce używany.

Dla potrzeb elektrodynamiki kwantowej poczyniono też próby uproszczenia procedury wejścia w sposób godny uwagi wszystkich użytkowników. Mianowicie, właściwe postawienie matematyczne problemu bywa nie mniej trudne od wykonania samego rachunku, chociaż jest to na ogół procedura algorytmiczna. Byłoby więc rzeczą użyteczną, gdyby program potrafił nie tylko przekształcać wyrażenia analityczne, ale również samodzielnie konstruować dane początkowe w oparciu o słowną instrukcję, tak jak student rozwiązujący zadania domowe, np.: „wylicz wszystkie poprawki czwartego rzędu do elastycznego rozpraszania elektronów na fotonach" (H e a r n 1971). Próby tego rodzaju nie osiągnęły na razie sukcesu, umożliwiającego ich szersze wykorzystanie praktyczne (C a m p b e l l i H e a r n 1970; C a l m e t i P e r r o t t e t 1971).

Dalsze przykłady zastosowań programów algebraicznych w elektrodynamice kwantowej zawierają artykuły C a m p b e l l a (1970) oraz B a r t o n a i F i t c h a (1972a).

Ponadto programy algebraiczne były stosowane w fizyce do następujących zagadnień:

1) Dopasowywanie krzywych zadanej rodziny do danych punktów doświadczalnych metodą najmniejszych kwadratów (T o b e y 1966). Program, napisany w języku FORMAC, przyjmował jako dane: wartości zmiennej niezależnej x_i, wartości zmiennej zależnej $y_i(x_i)$ wraz z przypisanymi im wagami oraz rodzinę funkcji $f(x, \{a_j\})$ numerowaną parametrami a_j z ustalonymi przez programistę wartościami początkowymi a_j. Obliczenia tego typu są w zasadzie numeryczne (chodzi o znalezienie numerycznych wartości a_j), lecz aby program numeryczny mógł rozpocząć pracę, należy najpierw wyliczyć wartości funkcji $f(x, \{a_j\})$ dla zadanych

wartości x_i oraz wartości pochodnych $\frac{\partial f}{\partial a_j}(x_i)$. Zwykle ten rachunek przygotowawczy był robiony „ręcznie" lub kodowany w programie numerycznym, co wymagało wiele czasu i wysiłku człowieka. Omawiany program wykonuje cały symboliczny rachunek przygotowawczy, znakomicie ułatwiając pracę programisty – numeryka.

2) Obliczanie elementów macierzowych w metodzie wariacyjnej rozwiązywania równania Schrödingera dla układów wieloelektronowych (Reeves 1966 i prace tam cytowane). Program ten był wielokrotnie wykorzystywany przez chemików do liczenia funkcji falowych elektronów w molekułach.

3) Znajdowanie postaci równań ruchu Newtona w dowolnych współrzędnych krzywoliniowych (Howard 1967). Problem ten jest tak prosty, że właściwie nikt nie zechce tracić czasu na pisanie danych dla komputera, tym bardziej że wydruk z programu był w dość egzotycznej notacji typu FORTRANu. Autor programu napisał go nie z myślą o użytkownikach, lecz o postępie sztuki programowania. Program posługiwał się rachunkiem tensorowym. Niestety, autor nie wiedział, że w tym czasie istniały już programy wykonujące rachunki tego samego typu na znacznie trudniejszych przykładach, dla potrzeb teorii względności (np. Fletcher i in., 1967). Niepotrzebne dublowanie cudzej pracy jest jednym z typowych zjawisk w nowo powstających dziedzinach wiedzy. Przykłady innych podamy w par. 5.

4) Obliczanie współczynników elastyczności w kryształach (Brugger 1965).

Zainteresowanych dalszymi zastosowaniami w fizyce odsyłamy do bibliografii Sammet (1968) oraz artykułu Bartona i Fitcha (1972a).

III. ZASTOSOWANIA W ASTRONOMII

Spośród zastosowań astronomicznych najczęściej cytowane w literaturze jest znajdowanie wzorów na kolejne wyrazy szeregów f i g w problemie dwu ciał (np. Barton i Fitch 1972a). Są to szeregi przedstawiające funkcje $f(\tau)$ i $g(\tau)$ dane wzorem:

$$x(\tau) = f(\tau)\,\vec{x}_0 + g(\tau)\,\dot{\vec{x}}_0,$$

gdzie: $\tau = t - t_0$, $\vec{x}(\tau)$ oznacza aktualne położenie badanego obiektu, $\vec{x}_0 = \vec{x}(t = t_0)$, $\dot{\vec{x}}_0 = \frac{d}{dt}\vec{x}(\tau)|_{t=t_0}$. Posługując się rozwinięciami funkcji $f(\tau)$, $g(\tau)$ i $\vec{x}(\tau)$ w szeregi Taylora oraz równaniami ruchu Newtona zapisanymi w postaci:

$$\ddot{\vec{x}}(\tau) = -\mu(\tau)\,\vec{x}(\tau),$$

a także związkami $\dot{\mu} = -3\,\mu\sigma$, $\dot{\sigma} = \varepsilon - 2\,\sigma^2$, $\dot{\varepsilon} = -\sigma(\mu + 2\varepsilon)$,
gdzie: $\sigma = \dot{r}/r$, $\varepsilon = (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)/r^2 - \mu$, można pokazać, że wyrazy szeregów f i g wyrażają się następującymi wzorami rekurencyjnymi (Stumpff 1959):

$$f_0 = 1,\ g_0 = 0$$

$$f_{n+1} = \dot{f}_n - \mu_0\, g_n$$

$$g_{n+1} = \dot{g}_n + f_n$$

gdzie: $\mu_0 = \mu(t_0)$, (f_n i g_n są funkcjami t_0, kropka oznacza różniczkowanie po t_0).

Problem dwu ciał został po raz pierwszy postawiony w ten sposób przez Lagrange'a, który podał jawne wzory na pierwsze pięć wyrazów obu szeregów. Mimo prostej postaci wzoru rekurencyjnego obliczenia te komplikują się gwałtownie z każdym krokiem rachunku i od czasów Lagrange'a aż do lat sześćdziesiątych XX w. nie posunięto się poza rząd $n = 7$. Jest to więc dobre pole do działania dla komputerów. Wzory na f_n i g_n do $n = 27$ zostały otrzymane w roku 1965 (Sconzo, Le Schack i Tobey 1965) za pomocą języka komputerowego FORMAC (Sammet i Bond 1964).

Obliczanie szeregów f i g nie jest jednak najszczęśliwszym przykładem zastosowania programów algebraicznych, ponieważ rachunek ten sprowadza się do wyliczania całkowitych współczynników szeregu, a to może być wykonane również przez program numeryczny (Veltman 1972). Klasycznym polem do zastosowania programów algebraicznych w astronomii są rozmaite schematy rachunkowe mechaniki nieba do obliczania rzeczywistych torów planet, księżyców i komet, zwyczajowo zwane teoriami. Tu również istnieje efektowny przykład pożytku z programów algebraicznych. Jeden ze schematów, zwany teorią Delaunaya ruchu Księżyca, w swoim oryginalnym sformułowaniu (Delaunay 1860) zawierał obok członów wiekowych również człony periodyczne w funkcji Hamiltona dla równania ruchu Księżyca. Należało te drugie wyeliminować przez rozwiązywanie równań Hamiltona metodą kolejnych przybliżeń. Delaunay wybierał jeden człon periodyczny, zaniedbując pozostałe, rozwiązywał równania Hamiltona, a następnie przez odpowiednio dobraną transformację kanoniczną otrzymywał nowy hamiltonian – już bez pierwszego członu periodycznego. Powtarzając ten rachunek dla kolejnych członów periodycznych dochodziło się w końcu do hamiltonianu stałego (w danym rzędzie przybliżenia), dla którego rozwiązanie jest trywialne. Następnie należało odtworzyć szereg opisujący aktualne położenie Księżyca, będący rozwinięciem względem 4 małych parametrów: mimośrodów orbit Księżyca i Ziemi, stosunku promieni orbit Księżyca i Ziemi oraz sinusa kąta między płaszczynami tych orbit. Z powodu powolnej zbieżności szeregu, dla celów praktycznych rachunek musiał być ciągnięty do wysokich rzędów przybliżeń. Był on tak skomplikowany, że wyliczenie pierwszych 9 rzędów przybliżeń zajęło Delaunayowi 20 lat pracy, a więc niemal całe aktywne życie naukowe dostępne człowiekowi, zaś opublikowane wyniki zajmują 2 tomy po 400 stron. Przy użyciu komputera ten sam wynik został uzyskany w ciągu pół roku, wliczając w to czas pisania programu. Przy okazji autorzy (Deprit i in. 1970a,b) wykryli w rachunkach Delaunaya kilka pomyłek (sygnalizowanych wcześniej w literaturze, lecz bez wskazania źródeł błędu, Deprit i in. 1970b) oraz zauważyli możliwość znacznego ulepszenia metody i poprawienia zbieżności szeregu (możliwość tę zrealizowali w innej pracy, dotyczącej ruchu sztucznych satelitów, Deprit, i in. 1970c). W konkluzji stwierdzili, że poświęciwszy temu problemowi pół roku zamiast 20 lat nie czują się do niego przykuci na całe życie, co daje im „cudowną wszechstronność".

Dwa różne szeregi opisujące tę samą funkcję (w przypadku mechaniki nieba – tor planety) muszą być oczywiście równoważne, tzn. musi istnieć algorytm pozwalający na wyliczenie każdego wyrazu drugiego szeregu na podstawie pewnej (skończonej lub nieskończonej) liczby wyrazów pierwszego. W tym sensie każde dwie „teorie" mechaniki nieba muszą być sobie równoważne. Zdarzały się jednak sytuacje, że istniały na ten temat wątpliwości. Na przykład

Deprit (1969) wysuwał wątpliwości co do tego, czy teoria Hori (1966) jest równoważna jego własnej, tzn. wolna od błędów, ponieważ swojej był pewien, zaś nie udawało się mu znaleźć jawnych wzorów przekształcających elementy szeregu Hori w jego szereg. Rozstrzygnięcie tej kwestii umożliwiły dopiero programy algebraiczne (Campbell i Jeffreys 1970). Wykonały one rachunki, dla których wykonania ręcznego jakiś człowiek musiałby znów złożyć ofiarę z całego swojego życia.

Innym przykładem skomplikowanego rachunku w mechanice nieba jest wyliczanie tzw. operatorów Newcomba (Newcomb 1895; Plummer 1960), czyli współczynników szeregu przedstawiającego siłę zaburzającą, z jaką jedna planeta oddziałuje na drugą. Małymi parametrami są tu mimośrody orbit i kąty nachylenia ich płaszczyzn. Rachunek (policzenie wszystkich wyrazów szeregu do 8-rzędu), który zajął Newcombowi kilka miesięcy, komputer wykonał w 5 min. (Gerard i in. 1965), przy czym w ciągu następnych 5 min. dociągnął rachunki do 12 rzędu. Operatory wyższych rzędów mogą być wyliczone bez większych przeszkód. Dzięki zastosowaniu programu udało się rozszerzyć metodę Newcomba na kilka przypadków szczególnych, które przedtem wymagały osobnego traktowania (np. przypadek orbit prawie współmiernych oraz przypadek, gdy orbita planety zaburzającej leży wewnątrz orbity zaburzanej).

Programy do wyliczania szeregów perturbacyjnych mechaniki nieba były również stosowane do obliczania torów pojazdów kosmicznych (Klumpp i Kalan 1970, cytowane za Bartonem i Fitchem 1972a).

Szereg innych zastosowań programów algebraicznych w mechanice nieba omawiają prace Jeffreysa (1971), Le Schacka i Sconzo (1968) oraz Bartona i Fitcha (1972a,b).

Na koniec wspomnimy o pracy Gustavsona (1966), która jest poświęcona czysto formalnemu problemowi istnienia całek pierwszych dla równań ruchu Hamiltona układu o n stopniach swobody w równowadze. Metody użyte w tej pracy są „w duchu" bardzo podobne do stosowanych w mechanice nieba: chodzi o konstruowanie kolejnych transformacji kanonicznych i rozwiązywanie równań ruchu Hamiltona w coraz wyższych rzędach przybliżeń. Problem ten wynikł jednak z badań nad strukturą galaktyk, a więc praca ta jest jedynym znanym nam przykładem zastosowania programu algebraicznego w astronomii do problemu nie związanego bezpośrednio z mechaniką nieba.

*

* *

Poprzestaniemy na tych przykładach już dokonanych zastosowań, bowiem omówienie wszystkich wymagałoby raczej książki niż artykułu, i to książki wnikającej w niepowiązane ze sobą szczegóły wielu różnych dziedzin nauki i techniki. Spróbujemy natomiast dać kilka wskazówek co do możliwości przyszłych zastosowań programów algebraicznych.

5. MOŻLIWE PRZYSZŁE ZASTOSOWANIA

Przede wszystkim programy algebraiczne warto stosować tylko wtedy, gdy istotnie zależy nam na otrzymaniu ścisłego wyniku, zaś grube oszacowania nie są zadowalające. Nieprzypadkowo podane przykłady zastosowań astronomicznych dotyczą niemal wyłącznie mechaniki

nieba. Jest ona jedynym bodaj działem astronomii w pełni zasługującym na miano nauki ścisłej, gdzie nawet kilkuprocentowe rozbieżności teorii z obserwacją wymagają wyjaśnienia. Gdy zgodność z dokładnością do czynnika 100 bywa zadowalająca, zaś do czynnika 10 – zdumiewająco dobra (co często można przeczytać w pracach astrofizycznych, a zwłaszcza kosmologicznych), niepotrzebne są ścisłe przekształcenia wzorów.

Warunek powyższy jest tylko konieczny, ale nie wystarczający. Dane wejściowe do programu muszą być przepisane na kartach lub wprowadzone do komputera przez konsolę. Jeśli są to wielkie, skomplikowane wzory, na których chcemy wykonać krótki ciąg przekształceń, ryzyko popełnienia błędu przy przepisywaniu jest zbyt wielkie, zaś zużyty na przepisywanie czas – porównywalny z czasem „ręcznego" wykonania rachunku. Typowe zagadnienie do zaprogramowania to takie, w którym dane początkowe są niewielkie, zaś wyniki pośrednie komplikują się w trakcie obliczeń. Wyliczanie tensorów krzywizny z tensora metrycznego oraz schematy perturbacyjne mechaniki nieba spełniają ten warunek idealnie.

Zagadnienia nadające się do zaprogramowania można podzielić, z technicznego punktu widzenia, na algorytmiczne i niealgorytmiczne. Wszelkie problemy algorytmiczne mogą być zaprogramowane już w tej chwili. Trudności i przeszkody należą tu głównie do sfery rzemiosła (częściowo również polityki, gdy chodzi o zakup komputera lub języka) i ich pokonanie leży w granicach możliwości każdego użytkownika, który zada sobie trud opanowania odpowiedniego języka oraz zdobycia do niego dostępu na komputerze. Kilkoma odpowiednimi komputerami oraz językami dysponujemy również w Polsce, o czym będzie jeszcze mowa dalej. Dla przykładu podamy kilka zagadnień w teorii względności, czekających na zaprogramowanie. Są to: przeliczanie tensorów z jednego układu współrzędnych do drugiego, wyliczanie tensorów krzywizny w różnych tetradach form zewnętrznych, wyliczanie obrazów spinorowych różnych wielkości (to zagadnienie musi być połączone z „nauczeniem" programu rachunku na liczbach zespolonych), wyliczanie lewej strony równań Einsteina w formalizmie Newmana-Penrose'a, wyliczanie lewej strony równań pola innych teorii grawitacji (np. Einsteina-Cartana).

W przypadku problemów niealgorytmicznych stosowane są dwa podejścia. Jednym z nich, uważanym przez większość specjalistów za rozwiązanie idealne, są systemy konwersacyjne, łączące szybkość i nieomylność komputera z doświadczeniem, intuicją i zdolnością do stawiania celów i rozpoznawania postaci, właściwymi człowiekowi. W systemach tych wszystkie działania algorytmiczne są wykonywane przez komputer, zaś człowiek rozpoznaje typy formuł sygnalizowanych na monitorze, dobiera eleganckie podstawienia i nakazuje maszynie cele dalszych przekształceń za pomocą klawiatury i pióra świetlnego. Systemy takie wymagają dość dużej pamięci (ok. 60 k słów, to jest 250 k bajtów) i mogą być realizowane na maszynach klasy IBM 360/370, R32 i R40, a więc również w warunkach polskich. Ponieważ w systemy konwersacyjne mogą być włączane, omówione dalej, programy heurystyczne – systemy te mają bardzo duże możliwości, a ich jedyną wadą jest stosunkowo prędkie męczenie się człowieka przy współpracy z maszyną.

Drugie podejście oparte jest na tzw. programowaniu heurystycznym, stosowanym w badaniach nad sztuczną inteligencją. W podejściu tym opracowano ogólne metody rozwiązywania szerokich klas zadań: wspomagane heurystycznymi dyrektywami poszukiwanie metodą prób i błędów w przestrzeni wzorów i ich transformacji, dekompozycję zadań, analizę celów i środków, automatyczne dowodzenie twierdzeń, samouczenie się programów (zagadnienia te omawiają monografie Nilssona (1971), Ernsta i Newella (1969), Popova i Firdmana (1976)). Powyższe metody zostały wykorzystane np. w programach całkujących i rozwiązujących równania różniczkowe (Slagle 1963; Moses 1971a), ogólnych systemach symbolicznych (np. w języku PROLOG – Bień 1974), programach faktory-

zujących wielomiany (C l a y b r o o k 1976). Ponadto programowanie heurystyczne wykorzystuje metody tworzone ad hoc dla wąskich klas zadań przy współpracy wybitnych specjalistów z danej dziedziny. Ich doświadczenie przekazywane jest programom bądź bezpośrednio, w formie „wykładu", bądź dzięki wykorzystaniu automatycznego protokołowania i generalizacji ich długotrwałej pracy w systemach konwersacyjnych.

Przykładem programu wyposażonego w możliwość uczenia się od człowieka jest program tworzący hipotetyczne modele przestrzenne wielkich cząsteczek na podstawie dostępnych danych doświadczalnych, ograniczających liczbę dopuszczalnych konfiguracji, ale nie wyznaczających żadnej jednoznacznie (C a m p b e l l 1970). Intuicja dobrego chemika odgrywa tu wielką rolę, toteż program przedstawia człowiekowi swoje hipotezy, zaś człowiek odrzuca niewłaściwe – wyjaśniając, dlaczego były złe – i sugeruje lepsze.

Podczas gdy metody ogólne w przypadku np. programów całkujących wymagają maszyn o bardzo dużych pamięciach (ok. 250 k słów) i dużego wkładu pracy, to wykorzystanie drugiego sposobu, z włączeniem pewnych ogólnych technik programowania heurystycznego, jest dla odpowiednio wybranych zagadnień możliwe również w naszych warunkach.

6. SYSTEMY ALGEBRAICZNE OGÓLNEGO PRZEZNACZENIA

W programach liczących tensory krzywizny stosunkowo niewielką część stanowią procedury specyficzne dla teorii względności, jak tworzenie symboli Christoffela lub form koneksji, czy obliczanie tensora Riemanna, natomiast większość objętości (np. w obecnej wersji naszego programu ok. 80%) zajmują procedury pomocnicze: tłumaczenie z jednej notacji na drugą, operacje na macierzach, różniczkowanie, organizowanie wydruków, upraszczanie algebraiczne, a więc operacje o bardzo wszechstronnych możliwych zastosowaniach. Operacje te, z nieznacznymi zmianami lub w ogóle bez zmian, mogą być przenoszone do wielu zupełnie różnych programów (lub też zapożyczone z całkiem innych programów).

Procent operacji pomocniczych nie zawsze jest aż tak wysoki, lecz w każdym programie algebraicznym stanowią one znaczną część, przy czym z reguły każda pomocniczna procedura może być zastosowana w dużej liczbie zupełnie różnych programów (np. blok upraszczania algebraicznego musi być wykorzystany w każdym programie algebraicznym; różne programy wymagają jedynie różnych stopni uniwersalności tej procedury). Fakt ten został już dostrzeżony i właściwie wykorzystany w świecie. W USA powstało kilka dużych systemów obliczeń symbolicznych, zdolnych do wykonywania ogólnych operacji matematycznych przy minimalnym zatrudnieniu użytkownika – np. w systemia MACSYMA policzenie całki, rozwiązanie równania algebraicznego dowolnego stopnia lub znalezienie wielomianu charakterystycznego macierzy wymagają wprowadzenia do maszyny danych oraz jednowyrazowej instrukcji. Jeżeli ktoś chce napisać swój własny program wykonujący obliczenia specjalnego typu i ma do dyspozycji cały system, nie jest już zmuszony do czasochłonnego samodzielnego uruchamiania programów pomocniczych, lecz po prostu je wywołuje, lub nawet są one wywoływane automatycznie.

Najbardziej rozwinięte z istniejących systemów to MACSYMA (M a r t i n i F a t e m a n 1971) zainstalowany w MIT i dostępny w wielu ośrodkach w USA i w Kanadzie za pośrednictwem sieci ARPA oraz SCRATCHPAD implementowany w IBM (G r i e s m e r i J e n k s 1971). MACSYMA wymaga 512 k słów pamięci. Systemy takie rozwinęły się z programów dla specyficznych zastosowań. Na przykład system GRAD-ASSISTANT Fletchera (F l e t c h e r i in. 1967) powstał na bazie programu do obliczeń tensora krzywizny, napisanego w języku LISP.

Nasz program również mógłby stać się zaczątkiem systemu ogólnego przeznaczenia, gdyby do pracy nad nim zechciała włączyć się większa liczba użytkowników o urozmaiconych potrzebach. Jest to chyba najważniejsze z możliwych przyszłych zastosowań programów obliczeń algebraicznych w matematyce i fizyce. Podkreślamy, że jest to przyszłe zastosowanie, bowiem w chwili obecnej duże systemy zajmują tak wielki obszar pamięci maszyny i działają tak nieoszczędnie, że próby zastosowania ich do poważnych problemów często kończą się niepowodzeniem: wyczerpanie dostępnej pamięci maszyny powoduje przerwanie realizacji programu. Ponieważ jest oczywiste, że większa ogólność systemu jest okupywana jego mniejszą efektywnością w konkretnym zastosowaniu, w chwili obecnej bardziej przydatne praktycznie będą małe, wyspecjalizowane systemy dostosowane do wąskich klas zagadnień. Tworzenie takich systemów będzie bardzo ułatwione, gdy użytkownicy skomunikują się ze sobą i części już istniejących systemów zostaną skopiowane w nowych. Autorzy dysponują już w tej chwili sprawdzonymi i poprawnie działającymi procedurami upraszczania algebraicznego, różniczkowania, liczenia wyznaczników i odwracania macierzy dowolnego stopnia oraz tłumaczenia wzorów z notacji infixowej typu FORTRANu na prefixową i z powrotem. (Procedury te mogą wymagać jedynie usprawnień typu optymalizacji wykorzystania pamięci maszyny w celu obniżenia kosztów wykonania programu, które na razie są bardzo wysokie. Najskuteczniejszym środkiem prowadzącym do tego celu jest napisanie całego programu bezpośrednio w assemblerze maszynowym – jest to wdzięczne zadanie dla zawodowych programistów).

Dużą przeszkodą w rozwoju programów algebraicznych jest fakt, że wszystkie one zostały uruchomione na komputerach, które przez producentów były przeznaczone i specjalnie przystosowane do rachunków numerycznych. Z tego powodu programy i systemy algebraiczne musiały być naginane do obcego dla nich narzędzia. Zapotrzebowanie na takie systemy było zbyt małe, aby opłacało się uruchamiać systematyczną produkcję specjalnej aparatury. Stoimy jednak na progu zmiany i w tej dziedzinie. W USA i ZSRR skonstruowano już pierwsze komputery, zaprojektowane specjalnie dla celów obliczeń symbolicznych (Smith i in. 1971; Korpela 1976; Tarasevich 1977). Posiadają one sprzętową realizację podstawowych operacji używanych w programowaniu symbolicznym i algebraicznym. Dla potrzeb programów algebraicznych buduje się też specjalne urządzenia wejścia/wyjścia maszyn cyfrowych (kilka z nich omawia Sammet 1966).

7. PROGRAMY ALGEBRAICZNE Z PUNKTU WIDZENIA ICH TWÓRCÓW

Do tej pory zajmowaliśmy się korzyściami, jakie przyniosło programowanie algebraiczne fizyce i naukom pokrewnym. Współpraca matematyków i programistów z fizykami jest jednak korzystna dla obu stron. W „epoce przedkomputerowej" matematycy interesowali się tylko istnieniem rozwiązań dla pewnych problemów, czasami poświęcając trochę uwagi również sposobom praktycznej konstrukcji rozwiązań, czyli algorytmom. Z zasady natomiast nie interesowała ich efektywność danego sposobu, tzn. czas i wysiłek, jakiego on wymaga. W przypadku komputerów efektywność (w której wysiłek mierzony jest wielkością zaangażowanej pamięci maszyny i czasu jej pracy) jest sprawą najwyższej wagi: alternatywa między nieefektywnym i efektywnym algorytmem bywa równoważna alternatywie między niemożnością i możliwością rozwiązania jakiegoś problemu na maszynie. Fakt ten skierował uwagę matematyków na problemy efektywności. Na przykład, w dziedzinie algorytmów algebraicznych jednym z głównych problemów tego typu jest w tej chwili znalezienie zadowalająco efektywnego algorytmu na wyznaczanie największego wspólnego podzielnika

dwu wielomianów wielu zmiennych (Barton i Fitch 1972b). Dla potrzeb programów całkujących opracowano też metody rozpoznawania, które funkcje pozwalają się scałkować w taki sposób, aby wynik nie zawierał symbolu całki. Pełne rozwiązanie tego problemu byłoby interesującym osiągnięciem matematycznym (Barton i Fitch 1972b).

Nauka o sztucznej inteligencji otrzymała ciekawe pole sprawdzania swych koncepcji teoretycznych i dziedzinę, z której wynikają nowe problemy. Badania nad sztuczną inteligencją i programowaniem algebraicznym są związane tymi samymi ośrodkami, językami programowania i ludźmi, i jak dotychczas silnie na siebie oddziaływały. Jako przykład można wymienić program SAINT Slagle'a (1963), który był pierwszym programem obliczającym całki nieoznaczone, a jednocześnie wprowadził do sztucznej inteligencji nowe fundamentalne idee dekompozycji zadań. Innym przykładem jest samouczący się program faktoryzujący wielomiany wielu zmiennych (Billy i Claybrook 1976). Jednocześnie rozwój programów algebraicznych prowadzi do powstawania nowych języków symbolicznych ogólnego przeznaczenia, struktur danych, sposobów organizacji pamięci i mechanizmów obliczeniowych.

8. UŻYTECZNOŚĆ I EFEKTYWNOŚĆ PROGRAMÓW ALGEBRAICZNYCH

Jak wspomnieliśmy, istnieje obecnie dla niektórych zagadnień (np. dla liczenia tensorów krzywizny) po kilka różnych programów. Są one napisane przez różnych autorów, w oparciu o różne formalizmy, różne metody programowania i różne języki programowania. Nic więc dziwnego, że ich sprawność jest bardzo różna. Powstaje zatem naturalne pytanie: które języki i które programy są najlepsze? Wśród zawodowych programistów przyjęło się kryterium, że ten program jest lepszy, który wykonuje swoją pracę w krótszym czasie i przy mniejszym wykorzystanym obszarze pamięci maszyny. Drugie kryterium jest rzeczywiście istotne, ponieważ zlekceważenie go może w ogóle uniemożliwić wykorzystanie programu. Kryteria te, traktowane zbyt rygorystycznie, są jednak bardzo zwodnicze mimo ścisłej wymierności. Program X może być szybszy i oszczędniejszy od programu Y nie dlatego, że jest mądrzej pomyślany, lecz dlatego, że jest prymitywniejszy i mniej „umie", a więc ma mniej możliwości do sprawdzenia po drodze. Bywa i tak, że program X okazuje się szybszy i sprawniejszy od Y, lecz użycie X wymaga od użytkownika znajomości treści programu, mozolnej wstępnej obróbki danych w celu przystosowania ich do potrzeb programu oraz dopisania do programu licznych, specjalnych instrukcji. Bywa też, że pewne operacje matematyczne (np. różniczkowanie) nie dają się wprost przetłumaczyć na język programu, lecz trzeba je symulować za pomocą sztucznych konstrukcji (tak jest np. w systemach SCHOONSHIP i ASHMEDAI (Roskies 1977), gdzie różniczkowanie jest symulowane przez zastępcze operacje algebraiczne). Wreszcie, program X może znacznie przewyższać inne sprawnością przy testowaniu go na prostych problemach lub na problemach, do których został specjalnie przystosowany. Przy bardziej skomplikowanych i nietypowych testach ów ogólny faworyt może okazać się w ogóle niezdolny do wykonania zadania. Dlatego rzeczywistym kryterium przydatności programu jest zakres problemów, które potrafi on praktycznie rozwiązać oraz łatwość i wygoda jego użycia. Kryteria te są trudniej wymierne, lecz właśnie one są ważne dla użytkownika, który korzystając z programów rzadko lub jednorazowo (takich będzie zawsze większość) da pierwszeństwo programowi, wymagającemu od niego mniej wstępnej nauki.

Programowanie obliczeń algebraicznych jest dość młodą i mało jeszcze rozwiniętą dziedziną działalności naukowej. Ma to kilka konsekwencji:

1) Dalekie jeszcze od doskonałości programy bywają bardzo kosztowne w użyciu. Na przykład, policzenie tensorów krzywizny dla ogólnej metryki Robertsona-Walkera przy użyciu pierwszej wersji programu napisanego przez autorów (w systemie CYFRONET) kosztowało ok. 7500 zł. Usprawnianie programu powoduje jednak szybką obniżkę tych kosztów. Angielski program CLAM, zainstalowany w CYFRONECIE, będący przykładem programu starannie zoptymalizowanego* liczy tensory krzywizny dla metryki Bondiego (B o n d i, v a n d e n B u r g i M e t z n e r 1962) za 119 zł, podczas gdy „ręczne" liczenie problemu tej klasy musiałoby zająć człowiekowi od kilku tygodni do kilku miesięcy**.

2) Błędy w dużych programach i systemach bywają trudne do odnalezienia i ujawniają się czasem dopiero po kilku latach skutecznego działania programu. Budzi to niekiedy nieufność kandydatów na użytkowników. W związku z tym należy podkreślić, że komputer „myli się" w zupełnie inny sposób niż człowiek. Błędy popełnione przez człowieka są chaotyczne, przypadkowe i nieprzewidywalne. Ten sam człowiek może dwa razy wykonać to samo obliczenie — raz poprawnie, drugi raz z błędami. Jest to główne źródło dręczącej niepewności, czy uzyskany przez człowieka wynik jest poprawny. Komputer realizujący program z błędami „myli się" zawsze w tym samym miejscu i w ten sam sposób. Skutki tych pomyłek bywają też z reguły całkiem inne niż w przypadku człowieka: wynik takiego programu jest najczęściej jawnie bezsensowny, tzn. w ogóle nie ma formy wzoru matematycznego, lecz jest niezrozumiałym dla laika napisem. Z tych powodów błędy w programach raczej nie bywają groźne w skutkach, występują zresztą bardzo rzadko***. Przeciwnie, literatura obfituje w przykłady błędów rachunkowych ludzi, wykrytych przez komputery (np. d'I n v e r n o 1975).

3) Społeczność naukowa jest nieoswojona z programami algebraicznymi, wielu ludzi nie wie o ich istnieniu, wielu innych nie rozumie i nie docenia ich znaczenia. Programy algebraiczne bywają oceniane jedynie pod kątem ich przydatności dla rachunków numerycznych, co jest jawnym nieporozumieniem. Niektórzy relatywiści starszej generacji są fanatycznymi wielbicielami „ręcznych" rachunków, zaś w programach algebraicznych widzą niebezpieczeństwo dehumanizacji nauki oraz odciągnięcia ludzi od pracy naukowej. Nieporozumienie dotyczy w tym przypadku definicji pojęcia „nauka". Istotą pracy naukowej nie są rachunki, lecz myśl stojąca za nimi. Rachunki są jedynie bardzo uciążliwym, choć nieuniknionym, narzędziem przetwarzania myśli. Nie jest więc istotne, kto te rachunki wykona, jeśli tylko zostaną one wykonane rzetelnie. W rzeczywistości sprawa zastosowania komputerów wygląda całkiem odwrotnie: człowiek, uwolniony od ciężkiej, ogłupiającej, wielomiesięcznej pracy nad rachunkami, będzie mógł przeznaczyć swój czas na pracę prawdziwie twórczą, w której komputery nie potrafią go zastąpić. Będzie mógł skupić się na generowaniu pomysłów, ich realizację i sprawdzenie pozostawiając maszynie. Życie ludzkie zostanie w ten sposób wzbogacone i przedłużone, zaś pole do popisu dla jednostek mało twórczych — znacznie zawężone. Może to mieć wyłącznie pozytywne konsekwencje.

4) Nie istnieją jeszcze ustalone kanały porozumiewania się różnych grup specjalistów od programów algebraicznych ani powszechnie przyjęte formy współpracy****, ponieważ wciąż

---

*Program CLAM napisany jest w języku assemblerowym maszyn CDC o nazwie COMPASS.

**Metryka Bondiego jest zwyczajowo używana jako testowy problem dla różnych programów, ponieważ jest najszerzej znanym przykładem trudnego, skomplikowanego rachunku w teorii względności.

***Błąd może pozostawać niewykryty przez dłuższy czas tylko wtedy, gdy znajduje się w rzadko używanej gałęzi programu.

****Na kontynencie amerykańskim rolę tę w pewnej mierze spełnia *Special Interest Group for Symbolic and Algebraic Manipulations (Amer. Comp. Mach.)*.

nowe grupy powstają żywiołowo. Skutkiem tego jest nie tylko niepotrzebne dublowanie pracy zrobionej przez kogoś innego, ale również wielość i rozmaitość stworzonych systemów. Na ogół każdy specjalista od programów algebraicznych zna dobrze tylko jeden system (często stworzony przez niego samego) i tylko jeden język programowania, i nie ma ochoty uczyć się innych. W imię interesów dalszego rozwoju nie należy ograniczać tej rozmaitości, bowiem dzięki swobodnej konkurencji przyszłość wyłowi systemy rzeczywiście najlepsze. Godnym ubolewania zjawiskiem jest jednak zaślepienie i fanatyzm znacznej części autorów, odnoszących się z lekceważeniem do cudzych osiągnięć, których nie rozumieją. Roskies (1977) podkreślał wielką skuteczność systemu ASHMEDAI i jego wyższość nad REDUCE, a także powszechniejsze od SCHOONSHIPu zastosowanie (w dwu ostatnich punktach mylił się zresztą). Veltman (1972), autor SCHOONSHIPu, stwierdził, że system ASHMEDAI jest używany przez jednego tylko człowieka. W tym samym artykule (dowcipnym, lecz aroganckim) Veltman stwierdził, że nie widzi żadnego praktycznego pożytku z programów algebraicznych w fizyce teoretycznej, jedyny bowiem sens ich tworzenia dostrzega w postępie wiedzy komputerowej, i tylko te programy go interesują, które wynikły z problemów komputerowych. Z najwyższym lekceważeniem wyraził się też o programach wykonujących tłumaczenia z notacji typu programowego na normalną matematyczną, odmawiając im wszelkiego znaczenia. Takie uwagi świadczą o zupełnym nierozumieniu istoty rzeczy. Autorzy nie znający LISPu wyrażają się o tym języku z lekceważeniem, podkreślając jego nieoszczędne wykorzystanie pamięci maszyny i twierdzą (fałszywie), jakoby oparte na nich systemy były bezużyteczne w praktyce. Znawcy LISPu podkreślają jego wielką uniwersalność i łatwość, z jaką programy mogą być modyfikowane, kontrastując ją z prymitywizmem i sztywnością struktur pamięci takich języków, jak FORTRAN. Jest rzeczą zabawną, w jak podobnych słowach różni autorzy wypowiadają sprzeczne ze sobą zdania: znawcy mechaniki nieba twierdzą, że programy algebraiczne zostały najwcześniej zastosowane i są najintensywniej wykorzystywane w mechanice nieba, inni twierdzą dokładnie to samo o elektrodynamice kwantowej. Dla czytelnika wynika stąd jedno zalecenie: wszelkie informacje o programach i systemach muszą być przyjmowane z bardzo ograniczonym zaufaniem i, w miarę możliwości, weryfikowane u strony przeciwnej.

9. RÓŻNE JĘZYKI PROGRAMOWANIA ALGEBRAICZNEGO

W naszym artykule poświęciliśmy wiele uwagi programom w języku LISP i systemom algebraicznym opartym na tym języku, ponieważ mamy już w nich pewne doświadczenie, zaś innymi nikt się dotychczas w Polsce, wg naszej wiedzy, nie zajmował. Z tego powodu dalsze rozwijanie programów w LISPie byłoby najmniej kłopotliwe. Dla pełniejszego zobrazowania sytuacji w świecie musimy jednak poinformować czytelników, że inne języki (w tym wymienione uprzednio) skutecznie konkurują z LISPem na tym polu. Część z nich jest dostępna w Polsce, lub może być sprowadzona. W systemie CYFRONET dostępne są języki: SNOBOL (Gimpel 1976) SCHOONSHIP, PROLOG (Bień 1974) i oparty na FORTRANIE system FORPREP/ALMAP (Machura 1975). Doświadczenie w ich wykorzystaniu jest jednak nikłe lub żadne, toteż trudno obecnie coś stwierdzić o ich praktycznej użyteczności. Stosunkowo łatwo mogą być zainstalowane na różnych maszynach systemy oparte na FORTRANie – ALTRAN, SAC–1, lub na LISPie – REDUCE. Autorzy niektórych z tych programów wyrażają w publikacjach gotowość udostępnienia ich innym użytkownikom (Cohen, Leringe, Sundblad 1976). Najkorzystniejsze byłoby zainstalowanie któregoś z tych języków na dostępnych w Polsce maszynach o dużej pamięci: IBM – 360/370, R40 lub R32.

Wybór odpowiedniego języka jest ciągle sprawą otwartą. Zagadnienia złożone wymagają, ze względu na efektywność, specjalnych dopasowanych do nich języków; z drugiej strony szereg zadań można skutecznie rozwiązywać w systemach ogólnych – wydaje się zatem, że należy rozwijać oba typy systemów dla zaspokojenia potrzeb jak największej grupy użytkowników. Ciekawe porównanie kilku systemów obliczeń symbolicznych, napisanych w różnych językach, zawiera artykuł Cohena, Leringe'a i Sundblada (1976). Wynika z niego, że największe nadzieje rokują systemy ALTRAN i SAC–1 oparte na FORTRANie. Systemy REDUCE (oparty na na LISPie) i SYMBAL nie ustępują im jednak wyraźnie, toteż ostateczne rozstrzygnięcie jest dopiero sprawą jutra.

W przyszłości systemy i programy tego typu powinny być tworzone przy współpracy fizyków i innych użytkowników, reprezentujących realne i konkretne potrzeby, z matematykami, specjalistami od sztucznej inteligencji i programistami oferującymi współczesne narzędzia programowania o potencjalnie bardzo dużych możliwościach. Byłoby dobrze, gdyby polscy fizycy i astronomowie włączyli się aktywnie do tego nowego nurtu badań. „Bazę" w postaci szybkich komputerów o dużej pamięci mamy bowiem w Polsce wystarczającą. Posiadanie na miejscu własnego systemu oraz grupy stale zajmujących się nim ludzi zapewnia największą sprawność w reagowaniu na potrzeby użytkowników oraz najlepsze wykorzystanie dostępnych możliwości technicznych. Ponadto, wobec wciąż małego zaawansowania prac w tej dziedzinie istnieją duże szanse, że opracowując cały system samodzielnie stworzy się coś zasadniczo lepszego od systemów już istniejących.

*

* *

Autorzy składają podziękowanie Doc. Drowi G. Sitarskiemu oraz Drowi M. Pindorowi za pomoc w dotarciu do niektórych źródeł informacji, oraz Drowi M. Pindorowi za udostępnienie końcówki komputera CDC należącej do Wydziału Fizyki UW.

LITERATURA

Abramov, C. A., 1973, Algoritmy i algoritmičeskije jazyki, 6, 88.
Ball, W. E., Berns, R. R., 1966, Commun. Assoc. Comp. Mach., 9, 626.
Barton, D., Fitch, J. P., 1971, Commun. Assoc. Comp. Mach., 14, 542.
Barton, D., Fitch, J. P., 1972a, Rep. Progr. Phys., 35, 235.
Barton, D., Fitch, J. P., 1972b, Computer J., 15, 362.
Bers, A., Kulp, J. L., Karney C. F. F., 1976, Comp. Phys. Commun., 12, 81.
Bień, J. St., 1974, *Opis systemu PROLOG*, preprint, UW, Wydz., Mat.
Bobrow, D. G. (ed.), 1968, *Symbol Manipulation Languages and Techniques*, North-Holland.
Bondi, H., van den Burg, M., Metzner, A., 1962, Proc. Roy. Soc. London, A 269, 21.
Boys, S. F., Cook, G. B., Reeves, C. M., Shavitt, I., 1956, Nature, 178, 1207.
Brugger, K., 1965, J. Appl. Phys., 36, 768.
Calmet, J., Perrottet, M., 1971, J. Comp. Phys., 7, 191.
Campbell, J. A., 1970, Comp. Phys. Commun., 1, 251.
Campbel, J. A., Hearn, A. C., 1970, J. Comp. Phys., 5, 280.
Campbell, J. A., Jeffreys, W. H., 1970, Celest. Mech., 2, 467.
Campbell, J. A., Kent, J. G., Moore, R. J., 1976, BIT (Nordisk Tidskrift for Informationsbehandling), 16, no 3, 241.
Claybrook, B. G., 1976, Artificial Intelligence, 7, 203.
Cohen, H. I., Leringe, O., Sundblad, Y., 1976. Gen. Rel. Grav., 7, 269.

Collins, G. E., 1971, Commun. Assoc. Comp. Mach., 18, 515.
Delaunay, C., 1860, *Theorie du mouvement de la Lune*, Gauthier-Villars, Paris, Vol. 1.
Deprit, A., 1969, Celest. Mech., 1, 12.
Deprit, A., Henrard, J., Rom, A. 1970a, Astron. J., 75, 747.
Deprit, A., Henrard, J., Rom, A., 1970b, Science, 168, 1569.
Deprit, A., Henrard, J., Rom, A., 1970c, Celest. Mech., 2, 166.
d'Inverno, R. A., 1975, Gen. Rel. Grav., 6, 567.
Ernst, G. W., Newell, A., 1969, GPS: *A Case Study in Generality and Problem Solving*, ACM Mono, Academic Press, New York.
Farlow, S. J., 1976, Comp. and Math. with Applic., 3, 15.
Fletcher, J. G., Clemens, R. W., Matzner, R. A., Thorne, K. S., Zimmerman, B. A., 1967, Astrophys. J. Lett., 148, L91.
Fletcher, R., Reeves, C. M., 1963, Computer J., 6, 287.
Frick, I., 1977, *8-th Int. Conf. on Gen. Rel. and Grav.*, Waterloo (Ontario), s. 372.
Gerard, J. M., Izsak, I. G., Barnett, M. P., 1965, Commun. Assoc. Comp. Mach., 8, 27.
Gimpel, J. F. 1976, *Algorithms in SNOBOL 4*, Wiley, New York.
Goldberg, S. M., 1959, *Solution of an electrical network using a digital computer*, M. S. Thesis, M.I.T.
Griesmer, J. H., Jenks, R. D., 1971, *Proceedings of the 2-nd Symposium on Symbolic and Algebraic Manipulation*, ACM, SIGSAM, New York, s. 42.
Gustavson, F. G., 1966, Astron. J., 71, 670.
Hearn, A. C., 1966, Commun. Assoc. Comp. Mach., 9, 573.
Hearn, A. C., 1971, Commun. Assoc. Comp. Mach., 14, 511.
Hearn, A. C., 1976, *A new REDUCE model for algebraic simplification*, preprint, Centre de Physique Theorique, Marseille.
Heldmann, G., Schnupp, P., 1968, J. Comp. Phys., 3, 208.
Hori, G., 1966, Publ. Astron. Soc. Japan, 18, 287.
Hörnefeldt, L., 1977, *8-th Int. Conf. on Gen. Rel. and Grav.*, Waterloo (Ontario), s. 187.
Howard, J. C., 1967, Commun. Assoc. Comp. Mach., 10, 543.
Jeffreys, W. H., 1971. Commun. Assoc. Comp. Mach., 14, 538.
Jordan, D. E., Kain, R. Y., Clapp, L. C., 1966, Commun. Assoc. Comp. Mach. 9, 638.
Klumpp, A., Kalan, G., 1970, *Apollo Guidance, Navigation and Control Memo No E-1543.*
Korpela, G., 1976, SIGSAM Bull., 10, no 3, 30.
Laurent, J. P., 1973, Artificial Intelligence, 4, 69.
Le Schack, A. R., Sconzo, P., 1968, Astron. J., 73, 217.
Machura, M., 1975, *FORPREP ALMAP – system manipulacji algebraicznej.* Prace COPAN nr 222.
Martin, W. A., Fateman, R. J., 1971, *Proceedings of the 2-nd Symposium on Symbolic and Algebraic Manipulation*, ACM, New York, s. 59.
McCormack, F. J., 1975, Comp. Phys. Commun., 9, 1.
Moses, J., 1966, Commun. Assoc. Comp. Mach., 9, 634.
Moses, J., 1971a, Commun. Assoc. Comp. Mach., 14, 527.
Moses, J., 1971b, Commun. Assoc. Comp. Mach., 14, 548.
Neidelman, L, D., 1967, Commun. Assoc. Comp. Mach., 10, 167.
Newcomb, S., 1895, Astron. Papers Amer. Ephemeris, 5, 301.
Nilsson, N. I., 1971, *Problem-solving methods in artificial intelligence*, McGraw-Hill, New York.
Payne, A. D., 1972, Comp. Phys. Commun., 4, 100.
Payne, A. D., 1976, Comp. Phys. Commun., 12, 145.
Petrick, R. (ed.), 1971, *Proc. 2-nd Symp. on Symbolic and Algebraic Manipulation*, ACM, New York.
Plummer, H. C., 1960, *An introductory treatise on dynamical astronomy*, Dover Press, p. 172.
Popov, E. V., Firdman, G. R., 1976, *Algoritmičeskije osnovy intellektualnych robotov i iskustvennogo intellekta*, „Nauka", Moskva.
Reeves, C. M., 1966, Commun. Assoc. Comp. Mach., 2, 276.
Rink, R. A., Guru, B. P., 1976, BIT (*Nordisk Tidskrift for Informationsbehandling*), 16, 161.
Risch, R. M., 1970, Bulletin of AMS, 76, 605.
Roskies, R., 1974, *Algebraic calculation by computers.* Invited talk at the APS Meeting, Salt Lake City, Utah (preprint, Univ. of Pittsburgh).
Roskies, R., 1977, informacja prywatna.

Sammet, J. E., Bond, E., 1964, IEEE Trans. Electr. Comp., 13, 386.
Sammet, J. E., 1966, Commun. Assoc. Comp. Mach., 9, 555.
Sammet, J. E., 1968, *Proc. of the IFIP Working Conference on Symbol Manipulation Languages*, Bobrow, D. C., (ed.), North Holland Publ. Comp. Amsterdam, 358.
Schnupp, P., 1968, Int. J. Quant. Chem., 2, 599.
Sconzo, P., Le Schack, A. R., Tobey, R., 1965, Astron. J., 70, 269.
Slagle, J. R., 1963, J. Assoc. Comp. Mach., 10, 507.
Smith, W. R., Rice, R., Chesley, G. D., Leiliotis, T. A., Lundstrom, S. F., Callhan, M. A., Gerald, L. D., Kuk, T. G., 1971, *Spring Joint Comp. Conf.*, s. 601.
Stoutenmeyer, D. R., 1975, Trans. on Math. Software, 1, 147.
Strubbe, H., 1974, Comp. Phys. Commun., 8, 1.
Stumpff, K., 1959, *Himmelsmechanik*, Band I, 2 Auflage, VEB Deutscher Verlag der Wissenschaften, Berlin, s. 291.
Tarasevich, S. V., 1977, *8-th Int. Conf. on Gen. Rel. and Grav., Waterloo*, (Ontario), s. 330.
Tobey, R. G., 1966, Commun. Assoc. Comp. Mach., 9, 742.
Veltman, M., 1972, Comp. Phys. Commun., 3 (Suppl.), 75.
Wacilar, M. D., Barnett, M. P., 1964, Commun. Assoc. Comp. Mach., 7, 704.