



Komputerowe rachunki symboliczne

Jerzy Karczmarczuk

Zakład Informatyki, Uniwersytet w Caen, Francja

1. Wstęp

Minęły czasy, gdy komputery służyły głównie do pracy numerycznej, tj. do przekształcania i generowania ciągów liczb obrazujących wyniki doświadczeń itp. Teraz są niezastąpione również w grafice, komunikacji, tworzeniu i analizie muzyki itd. Ponieważ to wszystko jest przetwarzaniem nadających się do formalizacji danych, fizykowi – od szkoły podstawowej, aż po noblistów (serio!) – mogą się także przydać do pracy z *formalnymi, symbolicznymi wzorami*. Możemy więc np. symbolicznie rozwiązywać równania, upraszczać formuły matematyczne w oparciu o ich *formalne* własności, różniczkować i całkować funkcje wyrażone wzorami, i tak dalej. Przykład: uruchamiamy jeden z popularniejszych systemów algebry komputerowej Maple i wpisujemy:

```
solve (a*x^2+b*x+c=0, x) ;
```

czyli: rozwiąż równanie $ax^2 + bx + c = 0$ względem x . a, b, c, x są *symbolami*, nie mają żadnej wartości numerycznej. Problem jest **strukturalny**, a nie liczbowy, program reprezentuje w pamięci znane nam równanie w postaci nadającej się do analizy, uruchamia swoje procedury, które przypominają znane ze szkoły algorytmy, i odpowiada:

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

w takiej właśnie, czytelnej postaci. Mając do dyspozycji symboliczne zmienne czyli dane, które określają nazwę zmiennej, np. tekst „ x ” itp., oraz mogą tworzyć i przetwarzać *złożone struktury danych* w rodzaju wielomianu $3z^3 - 8z - 7$, który jest reprezentowany jako para: nazwa zmiennej i *lista* współczynników, przy potęgach zero, 1, 2, ... *Współczynnik* to jest (jak powyżej) albo stała numeryczna, albo inna zmienna, albo w ogóle jakiś dowolny wielomian niezawierający zmiennej zasadniczej.

Tak więc przez rekurencję możemy utworzyć w komputerze dość skomplikowane wyrażenia, i to w *dowolnym* nowoczesnym języku programowania, który pozwala reprezentować w jakiejś danej *referencje* do innej danej, oraz tablice/listy

dowolnych danych, a przynajmniej pary danych. Ale języki „algebry komputerowej” się wyspecjalizowały, gdyż trzeba wiedzieć, jak reprezentować i realizować *algorytmy* pozwalające obliczać pochodne i całki funkcji symbolicznych, np. dzielić szeregi potęgowe, jak rozwiązywać równania wyższych stopni itp. Tak więc to nie sam język programowania decyduje o zastosowaniach symbolicznych, ale bogata biblioteka wspomagająca, często rozwijana przez wspólnotę użytkowników.

Wyrażenia matematyczne mogą być przedstawione zewnętrznie w postaci wzorów, albo w postaci graficznej, więc wyspecjalizowane języki symboliczne są także zwykle wyposażone w bogatą bibliotekę do drukowania wykresów 2- i 3-wymiarowych, choć niektóre korzystają z innych, niezależnych pakietów, takich jak Gnuplot (radzimy zaznajomić się z nim).

Oczywiście mamy do dyspozycji nie tylko wielomiany, ale i ułamki, i wektory, i formy różniczkowe, i symboliczne wyrażenia logiczne. Wyspecjalizowane pakiety dla fizyków zawierają procedury i wyrażenia ułatwiające pracę w kwantowej teorii pola czy teorii względności... Reprezentacja wewnętrzna wzorów jest zwykle dość przejrzysta dla specjalistów (listy złożone z list o odpowiedniej strukturze), ale jest ukryta przed oczami przeciętnego użytkownika. Nie będziemy więcej o tym mówić. Pewnym problemem jest niejednoznaczność reprezentacji wynikająca z faktu, że wiele różnych matematycznych wyrażeń może opisywać tę samą wartość, np. $(x + 1)(x - 1) = x^2 - 1$, i nie da się na ogół powiedzieć, która forma jest „lepsza” (krótsza, bardziej czytelna, czy wyrażająca pewne pożyteczne własności, tu np. wartości zer wielomianu?...), gdyż wszystko zależy od kontekstu. Zwykle jest możliwa transformacja między różnymi postaciami, ale czasami wygodnie jest dysponować nie jednym, a kilkoma różnymi pakietami algebry symbolicznej, które operują różnymi konwencjami, mniej lub bardziej przydatnymi w konkretnych zastosowaniach.

Czasami pakiety algebry symbolicznej służą nie po to, by bezpośrednio wspomóc nasze zrozumienie jakichś wzorów, ale aby – po czasami bardzo skomplikowanych przekształceniach – wygenerować *program numeryczny* w takich językach jak Fortran czy C.

2. Jeszcze kilka przykładów

2.1. Upraszczenie wyrażeń i rozwiązywanie równań

Wpiszmy w MuPADzie, innym systemie zbliżonym nieco do Maple'a

```
f := x/(x + y) + y/(x + y) - sin(x)^2 - cos(x)^2:  
f = simplify(f)
```

gdzie := przypisuje wyrażenie po prawej zmiennej *f*, a dwukropek kończy instrukcję, blokując jednocześnie automatyczne drukowanie wartości. Wyrażenie

`f = simplify(f)` jest *równaniem*, a nie przypisaniem. MuPAD oblicza i drukuje wartość obu stron:

$$\frac{x}{x+y} - \sin(x)^2 - \cos(x)^2 + \frac{y}{x+y} = 0$$

Wydruk jest dwuwymiarowy, ale w darmowej wersji systemu formatowanie używa standardowych czcionek, co daje wynik mniej elegancki niż w Maple. Możliwe jest także generowanie wydruków w formacie umożliwiającym powtórne wczytanie przez system, a także w formacie LaTeXu, w celu umieszczenia wyniku np. w artykule lub książce.

Jeśli Axiom, jeszcze inny system, dostanie wyrażenie: `solve(x^3-1)`, zareaguje w sposób nieoczekiwany:

```

2
[x= 1,x + x + 1= 0]
Type: List Equation Fraction Polynomial Integer

```

czyli znajdzie jedną wartość *explicite*, a dwie pozostałe zostawi w postaci równania. Dobrze znający Axiom spojrzą na **typ** wyrażenia i stwierdzą, że system „chciał” podać rozwiązanie w postaci wyrażenia wymiernego, co nie było w pełni możliwe. Jeśli natomiast zażądamy `radicalSolve(x^3-1)`, system odpowie:

```

+---+ +---+
- \|- 3 - 1 \|- 3 - 1
[x= -----, x= -----, x= 1]
2 2
Type: List Equation Expression Integer

```

i tutaj widzimy różnicę między Maplem a Axiomem, który liczy bardziej na użytkownika odnośnie do określenia dziedziny rozwiązań. Procedura `radicalSolve` zezwala na używanie pierwiastków. Czasami forma: $(x+y)/2$ bądź $x/2+y/2$ zależy od żądanego *typu*: czy chcemy mieć ułamek złożony z wielomianów, czy odwrotnie, wielomian złożony z wyrażeń wymiernych. Nie miejmy złudzeń: maszyna nie zastąpi naszej wiedzy matematycznej, a zwłaszcza nie będzie w stanie określić formy, w której chcemy mieć wynik, jeśli jest wiele możliwości.

2.2. Różniczkowanie i całkowanie

Praktycznie wszystkie pakiety potrafią różniczkować wyrażenia, a większość potrafi obliczać proste całki, rozwiązywać symbolicznie nieskomplikowane równania różniczkowe itp. Maxima oblicza i (niektóre) całki nieoznaczone, np. `integrate(x/(1+x^3), x)` dostarcza:

$$-\frac{\text{LOG}(x^2 - x + 1)}{6} + \frac{\text{ATAN}\left(\frac{2x - 1}{\text{SQRT}(3)}\right)}{\text{SQRT}(3)} + \frac{\text{LOG}(x + 1)}{3}$$

i także oznaczone, wynik wyrażenia `integrate(1/(1+x^2), x, 0, 1)` jest równy $\frac{\pi}{4}$. Gdy czegoś się nie da scałkować do końca, np. przy rozwiązywaniu równania różniczkowego $y'(x) + 3xy = \sin(x)/x$: `ode2('diff(y,x)+3*x*y = sin(x)/x, y, x)`, wynik zawiera symbol całki (tutaj sformatowaliśmy go czytelniej...):

$$y = e^{-\frac{3x^2}{2}} \left(\int \frac{e^{\frac{3x^2}{2}} \sin(x)}{x} dx + C \right)$$

Możemy również obliczać granice funkcji i przeprowadzać rachunki z analizy wektorowej: obliczać gradienty, rotacje itp. Ba, niektóre rachunki dają się przeprowadzać w przestrzeni, której liczba wymiarów jest także zmienną symboliczną!

2.3. Macierze

Oto reakcja MuPADu na wyrażenie `x:=matrix([[a,b],[c,d]]), 1/x;`:

$$\begin{array}{c} \begin{array}{cc} +- & -- \\ | & | \\ +- & -- \\ | & | \\ +- & -- \\ | & | \\ +- & -- \end{array} \begin{array}{cc} | & | \\ | & | \\ | & | \\ | & | \\ | & | \\ | & | \\ | & | \end{array} \begin{array}{cc} d & b \\ - a d + b c & - a d + b c \\ c & a \\ - a d + b c & - a d + b c \end{array}, \begin{array}{cc} +- & -- \\ | & | \\ +- & -- \\ | & | \\ +- & -- \\ | & | \\ +- & -- \end{array} \begin{array}{cc} | & | \\ | & | \\ | & | \\ | & | \\ | & | \\ | & | \\ | & | \end{array} \begin{array}{cc} b & a \\ - a d + b c & - a d + b c \\ a & d \\ - a d + b c & - a d + b c \end{array} \end{array}$$

Potrąfimy więc obliczać wyznaczniki, odwracać macierze, a także wykonywać inne operacje macierzowe, np. diagonalizacje, użyteczne w mechanice i innych gałęziach fizyki.

3. Pakiety algebry symbolicznej, komercyjne i dostępne publicznie/darmowo

Programów pozwalających operować symbolicznymi wzorami jest na rynku kilkadziesiąt. Większość to programy komercyjne, niektóre bardzo wyspecjalizowane, służące w konkretnych dziedzinach, np. w diagnostyce programów przemysłowych.

słowych, ale sporo uniwersalnych, które mogłyby się przydać w szkole, ale używane są rzadziej, gdyż kosztują sporo. Na początku były to pakiety akademickie, wymagające dużych komputerów, ale wkrótce okazało się, że takie systemy, jak muMath czy Derive (ten ostatni nadal jest sprzedawany przez Texas Instruments), mogą być także instalowane na domowych komputerkach wyposażonych w procesory 8-bitowe. Lawina ruszyła.

Oto kilka nazw programów komercyjnych, pozostawimy Czytelnikom odszukanie ich; redakcja *Fotonu* oferuje wszelką pomoc w ich lokalizacji i odpowieździ na konkretne pytania dotyczące ich używania:

- **Maple.** Jeden z najpopularniejszych w środowiskach akademickich, w związku z czym ciągle rozwijany; dodatkowe pakiety są tworzone na uniwersytetach i innych szkołach, i często dostarczane darmowo. Klasycznie wyglądający język programowania, wygoda graficzna. Podstawowa biblioteka rachunków symbolicznych bywa czasami używana z wnętrza innych programów, np. Matlabu, systemu służącego głównie do obliczeń numerycznych.
- **Mathematica.** Chyba najbardziej reklamowany system na świecie. Powstał w środowisku akademickim, ale jeden z jego autorów postanowił zrobić na nim pieniądze i dzieła dokonał (choć nie obyło się bez konfliktów). Bardzo dobra grafika, sporo dokumentacji. Według autorów i niektórych użytkowników jest to system dobrze dostosowany do psychiki debiutantów, ale zaawansowane zastosowania wymagają sporo studiów.
- **Reduce.** Jeden z najstarszych systemów algebraicznych, kiedyś bardzo popularny wśród fizyków. Był kiedyś rozprowadzany darmowo, ale jego twórca, Anthony Hearn i inni autorzy musieli wydać sporo pieniędzy, aby system przeżył, stał się więc komercyjny... Posiada sporo wyspecjalizowanych pakietów, rachunki tensorowe, równania różniczkowe itd. Nie posiada własnej grafiki, używa systemu Gnuplot, ale ciągle użyteczny. Głównie dla tych, którzy mają już z nim pewne doświadczenie.
- **Magma.** Jeden z nowszych systemów, w którym pojęcie *struktury matematycznej*, jako obiektu o sformalizowanych własnościach, jest bardzo ważne, co pozwala na konstruowanie algorytmów, które „patrzą” bardziej na elementy formalnej matematyki niż na konstrukcję danych informatycznych. Do tych systemów prawdopodobnie należy przyszłość, choć początkujący będą mieli więcej informacji do opanowania...
- **Derive.** Prosty w użyciu, interakcyjny. Popularny w licznych szkołach w Stanach Zjednoczonych. Dysponuje nieskomplikowaną grafiką.

- MuPAD. Podobnie jak w Magma, typy obiektów i struktury matematyczne (pierścienie, ciała itp.) są bardzo ważne. Prostszy niż Magma, MuPAD jest systemem dość popularnym w Niemczech, także w szkołach; powstał na uniwersytecie w Paderborn. **Ważne:** Prof. Fuchssteiner i jego współpracownicy oferują także wersję darmową. Niezła grafika, choć ustępująca niektórym innym systemom.

Podkreślmy, że to jest zaledwie kilka wśród mnóstwa! Systemów darmowych jest także sporo, niektóre nie ustępują systemom sprzedawanym, a nawet pod pewnymi względami je przewyższają.

- MuPAD. Jak już stwierdziliśmy, ten pakiet jest dystrybuowany darmowo, ale w ograniczonej wersji, mniej „przyjaznej” niż wersja komercyjna. Podstawowy pakiet rachunkowo/algebraiczny jest ten sam, tylko interfejs jest mniej wygodny, więc szkoły są zachęcane, aby kupić wersję kompletną, a darmowa może być ściągnięta przez uczniów (uniwersytet w Paderborn wydaje licencje osobiste). Składniowo trochę zbliżony do Maple’a. Aktualna wersja może być dystrybuowana razem ze Scilabem, pakietem do obliczeń numerycznych, który był już przez nas polecany na łamach *Fotonu*.
Miroslaw Majewski napisał książkę „MuPAD dla niecierpliwych”, polska wersja została wydana przez Bibliotekę Nauczyciela Matematyki.
- Axiom. Niegdyś komercyjny, rozwijany intensywnie przez bardzo liczną ekipę (głównie akademicką), stał się darmowy. Jest to pierwszy system (nazywający się ówczesnie Scratchpad), w którym własności obiektów matematycznych zostały ujęte nowocześnie i formalnie. Zarówno Magma jak i MuPAD zostały nim zainspirowane. Aktualna wersja pod Windows ma kłopoty z grafiką, ale to się zmieni (wymaga *X-Window System*, prace nad dostosowaniem są w toku). Axiom dobrze współpracuje z edytorem zwanym TeXmacs, który pozwala na drukowanie wyników w postaci elegancko formatowanej, podobnie jak Maple.
- Maxima (niegdyś komercyjny o nazwie Macsymba). Jeden ze starszych i najbardziej popularnych systemów, któremu dano nowe życie. Prostszy niż Axiom, o bardzo bogatej dokumentacji. Nie posiada własnej grafiki, ale jest dystrybuowany razem z Gnuplotem, z którym współpracuje w sposób przejrzysty i automatyczny.
- Inne: GAP i Macaulay2 są bardziej profesjonalne, dla matematyków. Niektórzy matematycy, przyzwyczajeni do innych systemów, korzystają z tych jako z bibliotek specyficznych algorytmów.

Co wybrać? Nauczyciel, który zamierza na serio poświęcić trochę czasu metodom rachowania symbolicznego i zachęcić uczniów do skorzystania zeń, ma aż nad-

miar możliwości. Ważne, aby wybrać jakiś system i go *dobrze* opanować. Nic nie stoi na przeszkodzie, aby przetestować kilka jednocześnie, ale w przypadku pracy z grupami uczniowskimi lepiej jest używać jednego, żeby się nie pogubić. Sądzi-
my, że aktualnie najwygodniejszym systemem z opisanych jest MuPAD, między innymi dlatego, iż istnieje do niego polska książka. Z drugiej strony, Maxima dysponuje niektórymi pakietami interesującymi dla fizyka, np. analizą wymiarową.

4. Grafika i programowanie

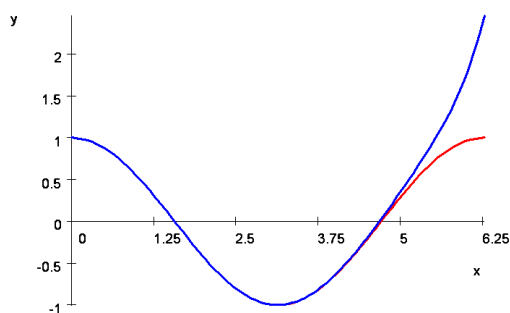
4.1. Grafika

Aczkolwiek wykresy są z konieczności oparte na wartościach liczbowych wyrażeń matematycznych, a nie symbolicznych, grafika stanowi bardzo ważny element omawianych pakietów i często stanowi o ich sukcesie, gdyż pozwala sprawdzać interakcyjnie niektóre wzory, a poza tym każdy wie, że jeden dobry rysunek może zastąpić tysiąc słów. Zwykle dysponujemy możliwościami wykresów 2- i 3-wymiarowych oraz wieloma stylami i opcjami rysowania; często także animacją.

Poniższy wykres został wygenerowany przez MuPAD instrukcją:

```
plotfunc2d(cos(x), scos(x), x=0 .. 2*PI, LineWidth=10);
```

gdzie `scos` jest procedurą przybliżającą funkcję cosinus szeregiem potęgowym. Konstruujemy ją w następnym rozdziale. Pakiety algebraiczne (a także Gnuplot) potrafią tworzyć pliki w różnych formatach, np. w kodzie PostScriptu, co pozwala na dołączenie wykresów o bardzo dobrej jakości do tekstów drukowanych, nie tylko na ekranie.



4.2. Programowanie

Opisane pakiety nie są tylko „kalkulatorami”, ale stanowią pełne języki programowania, umożliwiające konstrukcję programów iteracyjnych, procedur przepro-

wadzających analizę rozwiązań itp. Znaczna część bibliotek tych pakietów jest konstruowana właśnie w tych językach: Maple'a, MuPADu, Axiomu itd. Oto procedura obliczająca przybliżenie funkcji $\cos(x)$ przez szereg potęgowy:

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots, \text{ aż do wykładnika równego 12.}$$

```
scos:=proc(x) local f,y,s;
begin
f:=1; s:=-1; y:=1;
for n from 2 to 12 step 2 do
f:=f*x*x/(n*(n-1)); y:=y+s*f;
s:=-s;
end_for;
return(y);
end_proc;
```

Ten urwany szereg może być przedstawiony symbolicznie, ale także użyty do wykresu sprawdzającego jakość przybliżenia, jak powyżej. W tym wypadku znaleźliśmy jawną postać rozwinięcia, ale często musimy operować funkcją bardziej skomplikowaną. Wtedy można zażyczyć rozwinięcia przez sam system, np. Maxima pozwala na otrzymanie szeregu powyżej przez `taylor(cos(x), x, 0, 12)`. Szereg ten może być następnie przetwarzany przez nasze własne (albo biblioteczne) procedury, aby otrzymać np. przybliżenie funkcjami wymiernymi.

Aby móc sprawnie programować, zwykle konieczna jest znajomość wewnętrznej struktury wyrażeń, a w każdym razie możliwość stwierdzenia, czy mamy do czynienia z wielomianem (sumą), ułamkiem, funkcją specjalną itp., oraz możliwość ekstrakcji pojedynczych zmiennych z dowolnego wyrażenia. To przekracza ramy niniejszego artykułu.

5. Możliwe zastosowania fizyczne

Na ten temat można napisać całe książki, fizyka była głównym motorem tworzenia pakietów symbolicznych w latach sześćdziesiątych i siedemdziesiątych. Autorem jednego z nich, Schoonschipu jest późniejszy laureat Nagrody Nobla z fizyki, M. Veltman. Reduce i Mathematica również powstawały na terenie opartym przez fizyków. Istniejące biblioteki algorytmów zawierają więc tysiące procedur użytecznych w teorii względności czy fizyce cząstek. Elementarnych zastosowań jest mniej, ale np. Maxima posiada przynajmniej:

- Pakiet do analizy wymiarowej, pozwalający sprawdzić spójność wzorów względem użytych jednostek, albo np. znaleźć taką kombinację ładunków, sił, czasu itp., która jest bezwymiarowa.

- Pakiet do algebry i analizy wektorowej (gradienty, laplasjan itp.; różne układy współrzędnych), użyteczny w przetwarzaniu wzorów z mechaniki i elektrodynamiki. Inne systemy, jak MuPAD czy Reduce, również zawierają zbliżone pakiety.

Być może w przyszłości, w miarę zapotrzebowania, poświęcimy kilka stron *Fotonu* na konkretne przykłady, na razie kilka sugestii dla początkujących.

- Systemy symboliczne nadają się również do prostych, interakcyjnych obliczeń numerycznych, np. do rozwiązywania układu równań. Wynik może być podany w postaci ułamków wymiernych, co często jest korzystniejsze niż wartości zmiennoprzecinkowe. W MuPADzie wystarczy np. napisać
`solve([2*x+4*y-3=0, 3*x+y=8]);`
 żeby otrzymać
`{[x = 29/10, y = -7/10]}`.

Omawiane pakiety często używają list i podobnych struktur danych o dowolnej liczbie elementów. Jest to bardzo wygodne do przechowywania danych statystycznych, a następnie do obliczania średnich, dyspersji itp.

- Wbudowane procedury obliczania pochodnych są użyteczne do różniczkowych metod szacowania błędów doświadczalnego, więc i pracownia doświadczalna skorzysta z metod algebry komputerowej.
- Całkowanie równań różniczkowych pozwala generować i wykreślać trajektorie wynikające z równań ruchu, np. przy symulacji układu planetarnego.
- Wspomniana już analiza wymiarowa. Możemy np. przypisać $v:=3.01*\text{cm}/\text{sec}$, gdzie cm , sec są symbolami. Następnie wprowadzić $m:=100*\text{cm}$, itd. Wzory wyrażone liczbowo *razem* z jednostkami podadzą nam wymiar wyrażenia i dostaniemy ostrzeżenie, że przytrafiła się pomyłka, gdy np. wymiar będzie nie jednomianem, a wielomianem względem którejś jednostki, jeśli popełnimy głupstwo w rodzaju $v+a*v*v$, a a jest bezwymiarowe...
- Procedury wizualizacji danych: wykresy 2- i 3-wymiarowe, szybkie przetwarzanie formalnych wzorów w informację graficzną, są po prostu niezastąpione.
- Pakiety algebry symbolicznej pozwalają *ładnie formatować wzory*, a czytelność formuł matematycznych jest z pedagogicznego punktu widzenia dość zasadnicza.

Oczywiście omawiane systemy nie zastąpią nam znajomości wzorów i technik ich przekształcania, podobnie jak kalkulatory nie zastąpią całkowicie znajomości tabliczki mnożenia. Pozwolą jednak skoncentrować uwagę uczniów na *sensie*

przetwarzanych wyrażeń, a mniej na ich czysto składniowych, banalnych aspektach.

6. Referencje

Podajemy zaledwie kilka, wybranych dość losowo.

- [1] Wykaz systemów algebry komputerowej: research.mupad.de/CAIN/SYSPACK/.
- [2] Porównanie różnych systemów:
page.axiom-developer.org/zope/mathaction/RosettaStone.
- [3] *Newsgroup* poświęcona rachunkom symbolicznym: sci.math.symbolic.
- [4] System MuPAD, URL: research.mupad.de/.
- [5] Maxima: maxima.sourceforge.net/.
- [6] Axiom. page.axiom-developer.org/zope/mathaction/FrontPage.
- [7] Richard H. Enns, *Computer Algebra Recipes for Mathematical Physics*, Springer (2005).
- [8] Dla użytkowników pakietu Mathematica – pakiet przykładów fizycznych:
library.wolfram.com/infocenter/Courseware/4706/.
- [9] Gnuplot. www.gnuplot.info/.