

# Komputeralgebra rendszerek

## I. Bevezető és történeti áttekintés

Czirbusz Sándor

czirbusz@gmail.com

<http://compalg.inf.elte.hu/~czirbusz/>

Komputeralgebra Tanszék

ELTE Informatika Kar

D2.711A

2010-2011 őszi

# Tartalomjegyzék

- 1 Irodalom
- 2 Mi a komputeralgebra
- 3 Történeti áttekintés
  - Rendszerek, algoritmusok, alkalmazások
  - Az „ősidők”
  - A „rákészülés” időszaka
  - 1961-1966
  - 1966-1971
  - 1971-1981
  - 1981-1991
  - Azóta...
- 4 Egy üzleti és egy szabad szoftver
  - A Maple Project
  - A SAGE
- 5 Speciális és általános célú rendszerek
- 6 Előnyök, korlátok

# Irodalom I

## Könyvek



Járai Antal-Kovács Attila

*Komputeralgebra*

Informatikai Algoritmusok (sz: Iványi Antal)

Elte Eötvös Kiadó, 2004



Kovács Attila

*Komputeralgebra a tudományokban és a gyakorlatban*

Alk. Mat. Lapok 18, 1998



André Heck

*Introduction to Maple*

Springer-Verlag, 2003



Geddes-Czapor-Labahn

*Algorithms for Computer Algebra*

Kluwer Academic, 1992



F. Winkler

*Polynomial Algorithms in Computer Algebra*

Springer, 1996

# Irodalom II

## Internet

Néhány fontosabb internet-oldal:

Maple [www.maplesoft.com](http://www.maplesoft.com)

Sage [www.sagemath.org](http://www.sagemath.org)

Mathematica [www.wolfram.com](http://www.wolfram.com)

Wiki [Egy összehasonlítás](#)

# Mi a komputeralgebra I

## Komputeralgebrai- vagy szimbolikus-algebrai rendszerek

Szimbolikus számítások elvégzésére alkalmas számítógépes rendszerek (legtöbbször komoly numerikus és grafikus képességekkel). Az „algebrai” szó utal arra, hogy szimbolikus objektumokkal végzett műveletek algebrai eredetűek.

## A rendszerek alapfeladatai

- matematikai objektumok szimbolikus ábrázolása
- aritmetika ezekkel az objektumokkal

## Komputeralgebra mint tudományterület feladata

...erre az aritmetikára épülő hatékony algoritmusok keresése, elemzése és megvalósítása tudományos kutatásokhoz és alkalmazásokhoz. [JK]

# Mi a komputeralgebra II

## Egy próba–definíció

- 1 Algebrai struktúrákon futnak a programok.
- 2 Az eredmények pontosak, nincs numerikus hiba (egzakt aritmetika).
- 3 Az eredmények lehetnek formulák, matematikai objektumok.

# Mi a komputeralgebra III

## Magyarázat I

- 1 Gyakran célszerű egy kifejezést algebrailag egyszerűbb alakra hozni, majd utána numerikusan kiértékelni.
- 2 Az

$$x^2 + y^2 - 1 = 0$$

$$x - y =$$

egyszerű egyenletrendszer pontos megoldásai

$$\left\{ \left( \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right), \left( -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right) \right\}$$

# Mi a komputeralgebra IV

## Magyarázat II

- 3 Paraméteres problémáknál is formulát kapunk :

$$\int \frac{x}{x^2 - a} dx = \frac{\ln |x^2 - a|}{2}$$



# Szimbolikus vs. numerikus számítások I

## A Csebisev polinomok

### A Csebisev polinomok :

#### A Csebisev polinomok rekurzív definíciója

$$T_0(x) = 1; T_1(x) = x;$$

$$T_k(x) = 2 * x * T_{k-1}(x) - T_{k-2}(x), ha k \geq 2$$

# Szimbolikus vs. numerikus számítások II

## A Csebisev polinomok

A Csebisev polinomok értéke néhány  $k$  - ra

$k$	$T_k(x)$
0	1
1	$x$
2	$2 * x^2 - 1$
3	$4 * x^3 - 3 * x$
4	$8 * x^4 - 8 * x^2 + 1$

# Szimbolikus vs. numerikus számítások III

## A Csebisev polinomok

### FORTRAN program a Csebisev polinomok előállítására

```
REAL T(5)                                1  FORMAT(F5.2)
READ(5,1) X                               2  FORMAT(F9,4)
T(1)=1.0                                  END
T(2)=X
WRITE(6,2) T(1),T(2)
DO 10 N=3,5
T(N)=2.0*X*T(N-1)-T(N-2)
WRITE(6,2) T(N)
10 CONTINUE
STOP
```

# Szimbolikus vs. numerikus számítások IV

## A Csebisev polinomok

### Csebisev polinom előállítás ALTRAN-ban

```
PROCEDURE MAIN
ALGEBRAIC(X:4) ARRAY(0:4) T
INTEGER N
T(0)=1
T(1)=X
WRITE T(0),T(1)
DO N=2,4
T(N)=2*X*T(N-1)-T(N-2)
WRITE T(N)
DOEND
END
```

# Szimbolikus vs. numerikus számítások V

A Csebisev polinomok

Ugyanez MAPLE-ben

$T_0 := 1;$

$T_1 := x;$

**for**  $n$  **from** 2 **to** 4 **do**

$T_n := \text{expand}(2 * x * T_{n-1} - T_{n-2})$

**end do;**

# Rendszerek, algoritmusok, alkalmazások

E három, részben független tényező alakítja a komputeralgebra rendszerek fejlődését

**Rendszerek** Programozási nyelvek, programozási módszerek

**Algoritmusok** „Maga a komputeralgebra”

**Alkalmazások** Az algoritmusok alkalmazására kialakított rendszerek

Persze, a „vas”-ról se feledkezzünk el.

# Élet volt a komputerrek előtt is

A komputeralgebrában ma is kulcsfontosságú szerepük van :

- **Euklideszi algoritmus**

Legnagyobb közös osztó keresése, invertálás véges testekben

- **Kínai maradéktétel**

Homorfizmus „invertálása”

- **Newton interpoláció**

Egyenletek megoldás

# -1960

## Az első fecske

1953. H. G. Kahrmanian (Temple University) assemblyben  
UNIVAC I-re

- Az első magasszintű programozási nyelvek :  
FORTRAN(1958), Algol(1960)
- Lisp : John McCarthy alkotta meg 1958-ban a MIT-en,  
1960-ban publikálja.  
Ma is „aktív életet” él : AutoCad AutoLisp, Emacs, Scheme

[Wikipédia](#)



1961-1966

# A korai évek

## Az első rendszerek

- A M.I.T - en 1961–ben a SAINT (Symbolic Automatic INTegration)  
LISP-ben készült, heurisztikus algoritmusokra épült
- FORMAC az IBM-nél 1962-1964  
FORTRAN, elemi függvények manipulása (polinomok is)
- Bell Laboratories, ALPAK  
FORTRAN–ból hívható assembly rutinok
- PM –polynomial manipulation
- MATHLAB - az első interaktív program  
LISP alapú, polinom és racionális fv. kezelés

1966-1971

# Érlelődési fázis I

- SIN (Symbolic INtegration) 1966-1967, MIT  
már algoritmikus integrálást is használ, LISP alapú
- 1968, Stanford University, REDUCE, LISP alapú interaktív,  
fizikai kalkulációkra
- MATLAB-68
- Symbolic Mathematical Laboratory 1967  
grafikus kimenet CRT terminálra, több számítógépen  
„hálózatban”
- 1970 REDUCE-2  
Algol szerű RLISP-ben
- ALTRAN az ALPAK-ból

1966-1971

## Érlelődési fázis II

- A PM átdolgozásával keletkezik a SAC-1 (majd SAC-2, SAC/ALDES)
- CAMAL (CAMbridge ALgebra system)  
BCPL-ben készül, égi mechanikai feladatokra, relativitáselméleti számításokra
- SCRATCHPAD több rendszer tudását integrálja
- 1971 MACSYMA, már határértékszámítás, szimbolikus integrálás  
ez a „nagy ígéret”

1971-1981

## Az alkalmazási fázis

A rendszerek kikerülnek a laborokból mérnökökhöz, kutatókhoz.

- REDUCE a legelterjedtebbé válik hordozhatósága miatt
- MACSYMA erőteljes algoritmikus fejlesztés
- Megjelennek a speciális célú CAS-ek SAC/ALDES, SHEEP, TRIGMAN, SCHOONSCHIP
- Keletei fény Kievi Kibernetikai Egyetem, ANALITIK

1981-1991

# PC-k, új programnyelvek

- University of Waterloo : MAPLE (Geddes, Gonet) először B-ben, majd C-ben
- SMP –Symbolic Manipulation Program  
Stephen Wolfram, portábilis, szabály alapú
- MATHEMATICA, 1988 az első grafikus képességű rendszer + GUI !  
Stephen Wolfram, C alapú, saját nyelve az SMP-hez hasonló
- DERIVE  
PC-kre; csak interaktív környezet
- Cayley , GAP (Group Algorithms And Programming)  
csoportelmélet, a Mathematica illetve a Maple mintájára
- FORM, MACULAY, LiE, PARI

Persze, vannak, akik kimaradtak **Ez sem teljes**

# Tömegesedés

- Az internet elterjedése több frontos áttörés: gyors információáramlás, webmatek, a tudás mindneki számára elérhető
- Az üzleti rendszerek mellett széles körben elterjednek a szabad rendszerek. (Linux, GPL)
- A komputeralgebra bevonul az oktatásba, egyrészt mint segédeszköz (kalkulus, differenciálegyenletek, lineáris algebra), másrészt „saját jogán”.
- Kitágul a rendszerekbe épített matematika horizontja: logika, kombinatorika, geometria, univerzális algebra

# Rövid történet

- 1980–ban a Waterloo egyetemen K.O. Geddes, **Gaston Gonnet**, Morven Gentleman hozzák létre. Cél : egy kisebb komputereken is hatékony CAS létrehozatala.
- 1984–től a Watcom kezeli
- 1988–ban Geddes és Gonnet megalapítják a „Waterloo Maple Software, Inc.”-t. A Maple vezető CAS–sá válik
- 1990–válság, Gonnet távozik.
- Mérföldkövek Maple V, 8. Jelenleg : 14
- Autentikus lapok

# Jellemzők

- A kernel C-ben készült.
- A 10. verzió után a vizuális megjelenítés Java-ban történik
- A matematikai csomagok jelentős része saját(Algol W)
- Kétirányú kapcsolat Java, C, Fortran, VisualBasic felé
- Kliens–szerver megolodásként is működik
- Nem olcsó



# Történet

- Ambíció : a Magma, Mathematica, Maple ingyenes szabadforrású alternatíváját megvalósítani.
- Motiváció : Ki hisz egy olyan szoftver eredményének, amit az ára miatt nem tud megnézni?
- Alkotó : William Stein, Washington Egyetem
- 0.1. verzió: 2005. február 24.
- 1.0. verzió: 2006. február

# ellemzők

- Alapok : Python, objektumorientáltság.
- 150 szabadszoftvert integráltak hozzá
- Parancssoros (ipython) és GUI Web-böngészőben.
- Beépített Wiki
- Verziókontroll
- Upgradelés internetről
- Szabad szerverek érhetők el

# Speciális és általános célú rendszerek

## Általános célú rendszerek

Nagy matematikai apparátus, kellő lomhasággal.

## Speciális célú rendszerek

Speciális területre optimalizáltak

- SCHOONSCHIP, CAMAL, SHEEP, STENSOR
- Speciális matematikai területek
- [http://www.computeralgebra.nl/systems\\_and\\_packages/per\\_purpose/special/systems.html](http://www.computeralgebra.nl/systems_and_packages/per_purpose/special/systems.html)

# Előnyök, korlátok

## Előnyök

- Nagy mennyiségű algebrai számítás gyors elvégzése
- "Tévedhetetlen"
- Tud integrálni
- A matematika oktatás többszintű támogatása
- Kutatási munka felgyorsítása

## Korlátok

- A pontos aritmetika miatt gyakran exponenciálisan nő a számok és a kifejezések mérete
- Pszichológiai
- Numerikus problémák gyakran jobban kezelhetők „hagyományos” programozási nyelveken
- Bonyolultság, összetettség (általánosság  $\leftrightarrow$  hatékonyság)