In diesem Heft

Redaktion:
Benno Fuchssteiner, Waldemar Wiwianka
Universität - GH Paderborn
Postfach 1621, D - 4790 Paderborn
E-mail: benno@uni-paderborn.de
waldemar@uni-paderborn.de

Liebe Leserin, lieber Leser

Endlich liegt das dritte MathPAD-Heft vor. Leider ist MathPAD bisher nicht so häufig erschienen, wie ursprünglich geplant. Wir haben wohl einfach noch nicht genügend Routine, um die Arbeit, die in jedem Heft steckt, kontinuierlich mit unseren anderen Verpflichtungen in Einklang zu bringen. Aber vielleicht ändert sich das, so daß wir nach etwas Übung unsere steigende Leserzahl häufiger mit Berichten " erfreuen " können.
Stolz sind wir darauf, daß bei uns zwei Abonnementswünsche von uns völlig unbekannten Institutionen eingegangen sind. Leider konnten wir diesen Wünschen nicht entsprechen, da MathPAD nach wie vor kostenlos und in begrenztem Umfang an solche Interessenten abgegeben wird, die in der einen oder anderen Weise mit uns zusammenarbeiten. Die beiden Abonnementswünsche haben wir natürlich ehrenhalber in diese Kategorie aufgenommen, schon deshalb, weil wir von einer ökonomisch echt profitablen Situation selbst bei der hohen Zahl von **zwei** bezahlten Abonnements doch noch etwas entfernt sind.
Die stark wachsende Auflagenzahl zeigt allerdings, daß ein echtes Interessse an dem Heft besteht. Die Auflage dieser Ausgabe ist immerhin schon bei dreihundert angekommen, und das nächste Heft wird, sofern wir die technischen Schwierigkeiten meistern können, eine Auflage von nahezu 600 Stück haben. Dabei handelt es sich allerdings um ein Sonderheft zur DMV-Tagung. Neben einem kleinen lokalen Teil wird es hauptsächlich Besprechungen der gängigen Computeralgebra Systeme bringen.
Besonders dankbar sind wir wieder dem externen Autor, diesmal Paul Zimmermann von INRIA. Paul Zimmermann verdanken wir den Abdruck seines interessanten Vergleichs zwischen drei Computeralgebra Systemen. Herzlich möchten wir uns auch bei den Autoren unseres Fachbereichs bedanken, die nicht der MathPAD Arbeitsgruppe angehören, hier sind zu nennen H. Lenzing und W. Werner.

B.F. & W.W.

SUN macht das Fenster auf

Klaus Hering

Um die vollen Möglichkeiten eines grafikfähigen Arbeitsplatzes nutzen zu können braucht man ein System zur Fensterdarstellung, -gestaltung und -verwaltung.
Üblicherweise verwendete die Mehrzahl der SUN-Benutzer dazu `SunView`; diejenigen, die dem Trend folgten sich an Standards zu orientieren, benutzen `X-Windows`.

Nun gibt es das Fenstersystem, mit dem SUN beides unter einen Hut gebracht hat – `OpenWindows`.

`OpenWindows`, in der aktuell vorliegenden Version 2.0, bietet zunächst die üblichen Möglichkeiten der Manipulation mit Fenstern, wie das Öffnen/Schließen, Verschieben, Vergrößern, etc. derselben.

Das ist noch nichts Neues !

Was aber neu ist bei `OpenWindows`, ist die Tatsache, daß nun Applikationen sowohl von `SunView` als auch `X-Windows` nebeneinander laufen können. Dadurch hat sich das Angebot der zur Verfügung stehenden Tools wesentlich vergrößert, und der Anwender ist in der Lage sich das Beste aus beiden „alten“ Fenstersystemen auszusuchen.
So kann man etwa *xrn* benutzen um die Neuigkeiten aus aller Welt zu lesen und gleichzeitig den Previewer *dvipage* verwenden um sich LaTeX- Dokumente anzuschauen.

Daneben hat `OpenWindows` als Obermenge von `X-Windows` natürlich auch dessen Netzwerkfähigkeiten; man kann Applikationen auf einer entfernten Maschine rechnen lassen während die grafische Darstellung auf der eigenen Arbeitsstation geschieht.

Umgangsformen

Der Umgang mit `OpenWindows` ist nicht sehr verschieden vom Umgang mit `SunView`.
Zwar wird nun das Bewegen eines Icons/Fensters mit der linken statt wie gewohnt mit der mittleren Maustaste bewerkstelligt, aber es gibt keine grundlegenden Änderungen im Bereich der Fenstermanipulation.

Um eine Eigenschaft reicher geworden sind die sog. *pop up menus*, das sind Submenüs, die durch Aktivieren eines Hauptmenüpunktes erzeugt werden und wieder verschwinden sobald eine Aktion und/oder Programmeinstellung durchgeführt wurde.
Diese *pop up menus* wurden mit *push pins* ausgestattet. Wie an einer Pinwand kann man nun diese Submenüs „annageln“, d.h. fixieren und zur ständigen Verfügbarkeit auf den Hintergrund legen.

Abb. 1: Push pins am Beispiel eines Mail Tool Submenüs

Nützliche Programme

Das voreingestellte *Rootmenu*, das man bei `OpenWindows` zur Verfügung hat, enthält in der Grundausstattung einige hilfreiche Werkzeuge.

Zu diesen Programmen zählen

File Manager : ein Programm zur grafischen Darstellung der Dateistruktur und Directory-hierarchie; es erinnert ein wenig an die Oberfläche, wie sie vom Macintosh bekannt ist

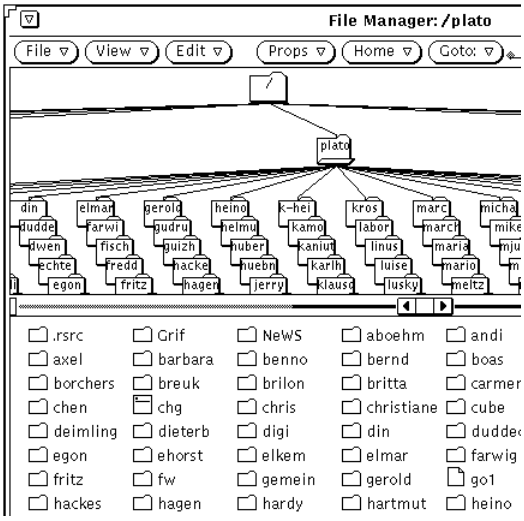


Abb. 2: File Manager — Darstellung von /user

Calender Manager : das Äquivalent zum *calentool*

Mail Tool : bequemes Programm zur Erledigung der elektronischen Post

Tape Tool : Programm, um die Bandstation anzusprechen : Sichern und Einlesen von Dateien und/oder Directories

Snap Shot : *VaDuLo* unter `OpenWindows`

Binder : ein nützliches Programm, das im Zusammenspiel mit **File Manager** das Ausführen der gebräuchlichsten Kommandos auf ein einfaches Mausklicken reduziert

Was ist zu tun, wenn man `OpenWindows` benutzen will ?

Um `OpenWindows` benutzen zu können muß man einige Vorbereitungen treffen. Dazu gehört die Erweiterung des Suchpfades und die Definition einiger Umgebungsvariablen. Sämtliche Änderungen erfolgen in der Datei *.cshrc* im Homedirectory des Benutzters.[1]

1. Environmentvariablen setzen :

   - setenv OPENWINHOME /local/'arch'/openwin
   - setenv LD_LIBRARY_PATH $OPENWINHOME/lib:/usr/lib

   Obige Vereinbarungen sind notwendig für die Installation von `OpenWindows`. Die folgenden Variablen sorgen dafür, daß die Manualseiten und der volle Umfang an Fonts zur Verfügung steht.

   - setenv MANPATH $OPENWINHOME/man:/usr/man
   - setenv FONTPATH $OPENWINHOME/lib/fonts:\
     /usr/lib/fonts/fixedwidthfonts

2. Pfad erweitern :

   $OPENWINHOME/bin
   $OPENWINHOME/bin/xview
   $OPENWINHOME/demo
   /usr/bin/X11

   müssen in den Suchpfad aufgenommen werden.

   Achtung : Die Reihenfolge innerhalb des Pfades ist wichtig. Die neuen Elemente müssen vor /bin und /usr/bin stehen !

Der Aufruf von `OpenWindows` erfolgt mit dem Befehl

openwin

Wie geht's weiter ?

An dieser Stelle muß noch ein wichtiger Punkt erwähnt werden, der die Entscheidung für `OpenWindows` positiv beeinflussen könnte.

SUN stellt mit der nächsten Betriebssystemversion nach SunOS 4.1.1 – letztere wird augenblicklich bei uns installiert – den Support von `SunView` ein. Daher muß sich der Benutzer über kurz oder lang auf ein anderes Fenstersystem umstellen.

`OpenWindows` ist sicher eine hochinteressante Alternative zu `X-Windows` und zudem noch mächtiger, da es eine Obermenge davon darstellt.

[1] Wer sich nicht sicher fühlt, kann sich damit natürlich vertrauensvoll an die Rechnerbetreuung wenden.

A comparison of *Maple V*, *Mathematica 1.2* and *Macsyma 309.6* on a Sun 3/60 (12 MB main memory, 30 MB swap)

Paul Zimmermann
zimmermann@inria.inria.fr

1 Initialization

The table below indicates the initialization time in seconds, the initial program size (SZ), the size of the part resident in main memory (RSS), and also the maximum number of processes that can be run simultaneously.

	Maple	Math[1]	Macsyma
load time	0.2 s	0.3 s	11 s
SZ (Ko)	528	3552	5952
RSS (Ko)	352	936	1720
maximum number	88	12	3

2 Integers

```
A2   2^8000/3^8000*15^8000/10^8000;
     s:=1:
D1   for i to 12 do s:=(s+2/s)/2 od:
D1   evalf(s^2-2);
     n:=500; s:=0:
D2   for i to n do s:=s+1/i od:
D2   evalf(s-log(n),50);
```

	Maple	Math	Macsyma
A2	65.4	214.1	303.0
D1	5.7	35.0	55.0
D2	3.0	5.6	33.5
Total	74.1	254.7	391.5

3 Reals

```
C1   s=0; Do[s=s+N[Sqrt[i],16],{i,100}]
C2   s=0; Do[s=s+N[Sin[i],16],{i,100}]
C3   s=0; Do[s=s+N[Log[i],16],{i,100}]
C4   s=0; Do[s=s+N[Sqrt[i],32],{i,100}]
C5   s=0; Do[s=s+N[Sin[i],32],{i,100}]
C6   s=0; Do[s=s+N[Log[i],32],{i,100}]
A3   N[Exp[Pi*Sqrt[163]],400]
L1   N[Sin[1],1000]
L2   N[Log[3],1000]
```

	Maple	Math	Macsyma
C1	0.30	1.00	5.38
C2	0.30	0.43	1.05
C3	0.37	0.42	1.23
Total	0.97	1.85	7.66
C4	2.0	2.3	48.7
C5	7.8	3.5	128.8
C6	7.8	3.3	78.5
Total	17.6	9.1	256.0
A3	2.9	8.8	254.9
L1	20.9	43.1	707.3
L2	9.9	39.8	2454.9
Total	33.7	91.7	3417.1

4 Control

```
B1   s:0; for i from 1 thru 5000 do
B1          if mod(i,2)=0 then s:s+i+1
            else s:s-i;
     fb(n) := if n<=1 then 1
              else 1+fb(n-1)+fb(n-2);
     nfb:entier(log(2000.)/
                log((1+sqrt(5.))/2));
B2   fb(nfb);
     f[n]:=n; s:0;
B3   for i from 1 thru 500 do s:s+f[n];
B4   for i from 1 thru 1000 do s:s+f[n];
```

[1] Math is a short-cut for Mathematica

mathPAD

	Maple	Math	Macsyma
B1	8.3	29.4	96.2
B2	2.8	11.5	24.8
B3	0.5	16.0	7.5
B4	1.0	42.3	15.2
Total	12.6	99.2	143.7

5 Symbolic calculations

```
A1  taylor(1/cos(z),z=0,80);
    f:=exp(cos(sqrt(1-x^4)));
D3  taylor(f,x=1/sqrt(2),8): evalf(");
    d:=1/(1+x^4);
A4  for i to 16 do d:=diff(d,x) od;
A5  for i to 16 do d:=int(d,x) od;
    p:=expand(product(product(x[i]-x[j],
                     i=j+1..5),j=1..5)):
D4  factor(p);
```

	Maple	Math	Macsyma
A1	28.6	124.8	268.2
D3	3.7	3.0	40.0
A4	0.5	2.1	6.0
A5	29.3	67.6	588.7
D4	5.9	9.8	50.0
Total	68.0	207.3	952.9

6 Arithmetic and Control

```
E1  ee=Sum[1/i!,{i,0,99}]
    CF[x_, n_Integer?Positive]:=
      Block[{xi,xp=x,r={}},
        Do[
          xi=Floor[xp];
          AppendTo[r,xi];
          xp=1/(xp-xi),
        {n}];
        Return[r]
      ]
E2  CF[ee,100]
```

	Maple	Math	Macsyma
E1	4.4	25.1	41.0
E2	0.7	1.2	80.1
Total	5.1	26.3	121.1

7 Taylor series expansions

```
    u[1](x) := log(1/(1-x));
    u[2](x) := x/(1+x);
    u[3](x) := exp(x)-1;
    u[4](x) := sqrt(1-x)-1;
    u[-3](x) := (1+x)^(1/3)-1;
    u[-2](x) := (1-x)*(1-x^2)*(1-x^3)*
                (1-x^4)*(1-x^5)-1;
    u[-1](x) := x+x^2+x^3+x^4+x^5;
    u[0](x) := sin(x);
    test(n) := block([t,i],
      t:x,
      for i:1 thru 8*n do
       (t:taylor(u[mod(i,8)](t),x,0,16)),
      t
    )$
F1  test(1);
F2  test(2);
```

	Maple	Math	Macsyma
F1	14.8	112.4	122.8
F2	35.8	357.7	370.0
Total	50.6	470.1	492.8

8 Differentiation

```
    test:=proc(nn)
       s:=0;
       for n from 1 to nn do
          d:=diff(sin(x)^n,x$n);
          s:=s+subs(x=0,d)*z^n;
       od;
       s
    end;
G1  test(50);
```

	Maple	Math	Macsyma
G1	28.2	308.5	1178.2

9 Modular factorization

$$a := 78 + 17x + 72x^2 - 99x^3 - 85x^4 - 86x^5 + 30x^6 + 80x^7 + 72x^8 + 66x^9 - 29x^{10} - 91x^{11} - 53x^{12} - 19x^{13} - 47x^{14} + 68x^{15} - 72x^{16} - 87x^{17} + 79x^{18} + 43x^{19} - 66x^{20} - 53x^{21} - 61x^{22} - 23x^{23} - 37x^{24} + 31x^{25} - 34x^{26} - 42x^{27} + 88x^{28} - 76x^{29} - 65x^{30} + 25x^{31} + 28x^{32} - 61x^{33} - 60x^{34} + 9x^{35} + 29x^{36} - 66x^{37} - 32x^{38} + 78x^{39} + 39x^{40} + 94x^{41} + 68x^{42} - 17x^{43} - 98x^{44} - 36x^{45} + 40x^{46} + 22x^{47} + 5x^{48} - 88x^{49} - 43x^{50};$$

```
I1  for p in {2,3,5,7} do
       Factor(a) mod p od:
I2  for p in {101,103,107,109} do
       Factor(a) mod p od:
I3  for p in {10007,10009,10037,10039} do
       Factor(a) mod p od:
```

	Maple	Math	Macsyma
I1	4.6	3.8	515.4
I2	13.8	14.1	1239.1
I3	53.4	22.9	231896.0
Total	71.8	40.8	233650.5

10 Factorization of composite polynomials

```
b[d_] := Expand[BernoulliB[d,x]*
                BernoulliB[d+1,x]]
```

```
    b10=b[10]
J1  Factor[b10]
    b20=b[20]
J2  Factor[b20]
    b30=b[30]
J3  Factor[b30]
    b40=b[40]
J4  Factor[b40]
```

	Maple	Math	Macsyma
J1	3.7	2.9	37.3
J2	18.0	22.8	561.9
J3	84.0	180.9	2678.6
J4	117.1	305.8	7059.3
Total	222.8	512.4	10337.1

11 Factorization of almost irreducible polynomials

```
test(d):=block(factor(bernpoly(x,d))
          ,factor(bernpoly(x,d+1)));
```

```
K1  test(25);
K2  test(45);
K3  test(65);
K4  test(85);
```

	Maple	Math	Macsyma
K1	4.8	6.9	138.8
K2	6.4	37.5	1086.1
K3	21.7	65.0	5251.6
K4	49.4	535.0	9913.0
Total	82.3	644.4	16389.5

12 Conclusions

The table below compares the efficiency of the three systems for each type of operation. The system with the value 1 is the fastest. A value k means k times slower.

	Maple V	Math 1.2	Macsyma 309.6
Integers	**1**	3.4	5.3
Reals 16 digits	**1**	1.9	8
Reals 32 digits	1.9	**1**	28
Reals 400-1000 digits	**1**	2.7	100
Control	**1**	8	11
Symbolic	**1**	3.0	14
Arithmetic/control	**1**	5	24
Taylor series	**1**	9	10
Differentiation	**1**	11	42
Factor (modular)	1.8	**1**	5700
Factor (composite)	**1**	2.3	46
Factor (irreducible)	**1**	7.8	200

CEBIT-Ein Messerundgang

Benno Fuchssteiner und Waldemar Wiwianka

Wie stehts doch zur CEBIT im Editorial der Wirtschaftswoche 12/91
Messen ... bieten eine ganz gute Gelegenheit .. den technischen Reifegrad von Nationen und deren ökonomische Zukunftschancen zu vergleichen. ... Deutschland soll eine Großmacht, dieses Jahrhundert ein europäisches werden? In den Messehallen von Hannover werden solche Prognosen als pures Wunschdenken entlarvt. Deutschland und mit ihm ganz Europa sind dabei, eine Computerkolonie der Japaner (und der Amerikaner zu werden). Computer und Informationstechnologie überhaupt werden zum wichtigsten Produktionsfaktor, sie entscheiden über die Wettbewerbsfähigkeit einer Volkswirtschaft. Für Deutschland und Europa sieht es da finster aus.
Das kann doch nicht wahr sein, das wissen wir nun besser! Also streben wir als erstes zum Stand eines deutschen Prestigeobjekts, dem Supercomputer *Suprenum*. Fehlanzeige, wir finden das Milliardending nicht, und auch intensives Blättern im 1961-seitigen Messekatalog hinterläßt keine Spur des verkaufsfertigen Fabelwesens. Also hilft nur noch Nachfrage beim *GMD*-Stand, wo man uns verschämt darauf verweist, daß man ja schließlich nicht allein dafür verantwortlich sei. Wofür eigentlich? Dabei haben wir bei der Nachfrage weder Kritik noch Ironie durchblicken lassen, warum auch? Und auch die Vermutung, daß der Suprenumstand nach Kalkar verlegt sei, kam uns erst später, da hatten wir aber schon zum Mittagessen etwas Wein getrunken.
So ausschließlich brauchen wir uns deshalb aber nun nicht dem *Sub*-computing zuzuwenden, denn später entschädigte uns dann *PARSYTEC* mit englischer Transputertechnologie etwas, aber auch nur etwas, für entgangene Gaumenfreuden des *Super*-computing. Trotzdem, als berichtenswert ausgewählt haben wir nur einige kleine und pfiffige Produkte, vornehmlich aus Übersee, dabei folgen wir aber nun bedenkenlos unserem eigenen Geschmack.

1 NEXT

Endlich da! auf dem deutschen Markt, der *NEXT*-Cube, oder dessen Nachfolgemodelle, vom legendären Apple-Gründer Steve Jobs. Vor zwei Jahren erlebte einer von uns die Markteinführung des Wundercubes im Medien-Spiegel der USA. Da wurde klar, um *APPLE* und *SUN* wars gleichermaßen geschehen. Rückblickend war es wohl eher ein Marketingereignis als der Durchbruch in eine neue technische Dimension am Arbeitsplatz. Der Cube, mit exzellentem Styling übrigens, dem Vernehmen nach von einem deutschen Designer, wurde in den USA, nachdem erst einmal der Qualm des ersten Medienrummels verraucht war, nur zögernd aufgenommen. Das, beziehungsweise die, Nachfolgemodelle sind heute, zwei jahre später, allerdings deutlich verbessert. In Abkehr von einer etwas merkwürdigen Marktphilosophie hat der Cube und seine Geschwister nun auch ein Floppy-Laufwerk, und das vorher obligatorische CD-Rom Laufwerk ist nun optional. Aber die entscheidenden Nachteile bestehen immer noch: zu geringes Software Angebot (ganze 96 Applikationen wie die FAZ zu berichten weiß), ein Betriebssystem, welches zwar gut ist, aber vom gängigen Standard abweicht. Natürlich, gute Technologie, aber doch nichts Außergewöhnliches. Auch der Preis stimmt, das Einsteigermodell, mit Studentenrabatt für etwas mehr als 8000 DM. Ob die angepeilten 5000 Einheiten in diesem Jahr allerdings verkauft werden, darf doch sehr bezweifelt werden. Fazit: *NEXT* hat einen langen Atem (wie der europäische Leiter Theo Webrans sagt) und braucht ihn wohl auch. Apple und Sun können beruhigt schlummern, einstweilen zumindest.

2 APPLE-Macintosh

Wir wollen es gleich zugeben, *APPLE* gehört unsere Liebe, deshalb wollen wir in vorbildlicher Unparteilichkeit bei Apple auch jeden kritischen Unterton meiden. Obwohl sicher Anlass gegeben wäre, denn umwerfend Neues gab es am Stand nicht. Wahrscheinlich tut sich da im Herbst etwas. Aber die Preise sind gesenkt worden, was ja zumindest etwas ist. *APPLE* nähert sich also mehr und mehr vernünftigen und erschwinglichen Hardwarepreisen, und es ist nicht mehr unbedingt teurer, einen guten Geschmack zu haben. Hinreißend immer noch der Laptop, diesmal schon wieder mit einem durch Hintergrundbeleuchtung verbesserten Bildschirm (aber da gibts mittlerweile andere Hersteller, die auch nicht schlafen, und deren Bildschirme inzwischen ebenso schön sind, siehe weiter unten). Ich warte mit dem Kauf des *Schlepptops* (wie er von einem Vertreter der Konkurrenz wegen seines Gewichts lieblos genannt wurde), allerdings noch bis er einen 68040-Prozessor hat, federleicht ist und die statischen Rams viel billiger sind. Aber haben täten" wir den Schlepptop doch schon gerne jetzt. Einer meiner Studenten hat einmal gesagt, ein Rechner sei nur soviel wert, wie der Drucker, den man sich dafür leisten kann. Stimmt eindeutig nicht, der Mac konnte und mußte einen immer für den zugehörigen *affordable* Imagewriter entschädigen. Das ändert sich nun: Der *Stylewriter* ist da. Klein, leicht und leise, ein 360dpi Tintenstrahldrucker mit Sheetfeeder (abnehmbar für die Reise) zum annehmbaren Preis. Nur Zeit muß man für ihn haben, er besteht einfach darauf daß man ihm lange, sehr lange, und liebevoll zuschaut wenn er sein Wunderwerk verrichtet. Da wird sich Canon, pardon Apple, noch etwas einfallen lassen müssen.

3 NEC

Was haben wir uns geärgert, daß wir das *APPLE*-Angebot für den Sonderpreis des CD-Rom Laufwerks nicht angenommen haben, denn inzwischen kostet es wieder über DM 1600. Wer sich mit uns geärgert hat, wird sich freuen über *NEC*'s neues tragbares Laufwerk CDR-35, für *dual use* nach dem Motto morgens Blechschrauben, abends Heavy Metal". Selbst wenn Sie wie wir, weder mit Heavy Metal was am Hut haben, noch Ihren Computer zum Entwurf von Blechschrauben benutzen, so werden sie die Idee doch begrüßen: ein tragbarer, gewöhnlicher CD-Player mit aufsteckbarem Computerinterface und den verschiedensten Treibern. Wahlweise Stromversorgung über Nertz oder Akku. Also auch etwas fur Altmodische (Beethoven statt Heavy Metal). Einziger Nachteil, etwas langsam beim Suchen (1,5 Sec. mittlere Zugriffszeit). Das Ganze zu einem recht guten Preis von ca. DM 1200. Kaufinteressenten sei aber ein Tip gegeben, der den Preis halbiert. Das billigste Angebot, was wir in MacUser gefunden haben ist 380 US Dollar.

4 TOSHIBA

Auf dem SUN Stand wurde er uns schon angekündigt, der *Sparc Laptop* von TOSHIBA. Es ist eine vollwertige Workstation, dieser Laptop; eigentlich müßte er ja Portable heißen, da sein Transportgewicht fast 8 kg beträgt und er außerdem keine Akkus hat. Aber trotzdem kann der Sparc LT gefallen; er bietet eine hohe Leistung, ist UNIX-fähig, vollständig kompatibel zu SunOS und hat ein LCD mit der gleichen Auflösung (1152 * 900) wie ein normaler Sun Bildschirm. Auf dem Stand konnte man dann auch mit dem Sparc LT spielen. Selbst OpenWindows 2.0 war verfügbar und wir konnten *richtig* mit dem Sparc LT arbeiten, wie wir es von unseren Sun Desktop Workstations gewohnt sind; einfach toll. Einen Haken hat die Sache, der Portable soll erst Ende +91 auf den deutschen Markt kommen und ca. 20.000 DM kosten.

Technische Daten

Prozessor:	SPARC 20 MHz
Hersteller des Chip-Satzes:	Cypress
	Mips: 13.2
Coprozessor:	Weitek WTL 3171
	1.5 MFLOPS
Speicher:	8 MB bis 40 MB ,
	654 kB Cache
Festplatte:	180 MB intern; bis
	2.8 GB exterm
Diskette:	3.5 1.44 MB
Bus-System:	1 * S-BUS, 1 * Modem-Steckplatz
Erweiterungen:	1 * Ethernet-Steckplatz,
	2 * RS232C, 1 * SCSI,
	1 * Parallel-Schnittstelle
Bildschirm:	
Model L10:	12monochrom,
	rückwärtig beleuchtet,
	TFD,
	1125 * 900 Pixel (45 ms)
Model E10:	Elektrolumineszenz mit weitem Sichtwinkel (2ms)
Betriebssystem:	OS/ASR4
	(aufbauend auf SunOS R4)
Betriebssystem-Erweiterungen:	
	SunView,
	MS-DOS Dateiverwaltungssystem
Gewicht:	7.9 kg (17.3 lbs)
Abmessungen:	340 * 375 * 98 mm
Preis:	ca. $ 13.000

Eine andere tolle Neuigkeit auf dem TOSHIBA Stand war der T3200SXC, ein Laptop, mit einem Farbbildschirm in revolutionärer neuer Technologie und der Leistung des 386-Prozessors, der mit 20 MHz getaktet ist. Neben dem Bildschirm mit seinen revolutionären Farb- und Grafikmöglichkeiten hat dieser Laptop -oder muß man auch ihn *Schlepptop* wegen seines hohen Gewichts nennen- aber noch mehr zu bieten. Beispielsweise verfügt er über eine 120 MB Festplatte und einen Hauptspeicher von bis zu 13 MB. Somit ist man bei der Wahl des Betriebssystems gut gerüstet. Man kann zwischen UNIX, OS/2 und DOS mit Windows 3.0 wählen. Selbst mit UNIX und X-Windows macht dieser Laptop eine gute Figur und dazu noch alles in Farbe. Da fragt man sich, warum soll der professionelle Anwender überhaupt einen Desktop kaufen?

Technische Daten

Prozessor:	Intel 386 SXC 20/8 MHz Sockel für Co-Prozessor 387
Speicher:	1 MB bis 13 MB
Festplatte:	120 MB, mittlere Zugriffszeit 19 ms
Diskette:	3.5 mit 1.44 MB / 720 kB
Bildschirm:	Hochauflösendes Farb-Display in TFT-LCD Technologie mit automatischer Helligkeits- regelung
Grafik:	VGA kompatibel + Toshiba Grafikmodus 640 * 480 Punkte 256 Farben
Bus-System:	AT-Steckplätze, RS232C, Anschluß für extern VGA Bildschirm
Betriebssystem:	MS-DOS4.01, MS-DOS3.3, MS-OS2/2.10 optional UNIX
Gewicht:	7.9 kg (17.3 lbs)
Abmessungen:	370 * 395 * 110 mm
Preis:	ca. 25.000 DM

5 Multimedia Workstation

Eine interessante Anwendung haben wir dann noch auf dem Sun Stand gesehen. Die Visualisierungs Komponenten der Firma APE. Hier hatte man eine Farb Sun Workstation mit einem Scan-Converter -der Brücke zwischen Computer-Grafik und TV-Video- verbunden. Als Eingangssignal dient RGB und als Videoausgang hat man VHS, S-VHS, RGB, Pal und noch vieles mehr zur Verfügung. An den Scan-Converter war ein Bildplattensystem angeschlossen. Es bietet die Möglichkeit bis zu 72000 Videobilder pro Bildplatte zu speichern. Bei der Aufzeichnung der Bilder kann man zwischen 25 Bilder/sec oder Einzelbildern wählen. Bei der Wiedergabe besteht die Möglichkeit zwischen 75, 50, 25 Bilder/sec oder EinzelBildern zu wählen. An das Bildplattensystem war dann ein Videorekorder und ein Videomonitor angeschlossen. Man konnte die Bilder, die auf der Sun generiert wurden, sowohl auf dem Videoschirm sehen, als auch auf eine Videokassette aufzeichnen. Die Steuerung dieser Prozesse wurde über eine Menüoberfläche auf der Sun Workstation dem Anwender einfach und transparent nutzbar gemacht. Aber wie alles im Leben, haben auch diese Visualisierungs Komponenten ihren Preis, sie sollen 120.000 DM kosten.
Anwendungen für dieses Visualisierungssystem haben wir reichlich, z.B. würden wir gern einen Soliton Film produzieren. Hierfür suchen wir aber noch einen Mäzen.

6 Sculpt3D

Zum Schlusss noch eine Softwareperle: Sculpt3D von Systematics. Das 3D-Graphikprogramm für den professionellen Layouter und CAD-ler, der einen Mac hat. Exzellente dreidimensionale Darstellung mit Lichteffekten nach der *ray-tracing*-Methode. Nicht einfach zu bedienen, denn etwas Darstellende Geometrie muss man schon verstehen, aber unübertroffen in der Qualität. In Zukunft auch mit Animation zu haben. Weitere Ausführungen ersparen wir uns, gesehen muss man es haben. Etwas prohibitiv nur der Preis, zumindest für Universitäten: ca. 5000 DM

Zum Titelbild: AUTOMATH gegründet??

Vor etwas mehr als einem Jahr kam die Idee auf, in Paderborn ein Institut für *Nichtlineare Dynamik, Automatisierung und Instrumentelle Mathematik* zu gründen. Diese Gründungsinitiative wird inzwischen von den Arbeitsgruppen Dourdoumas (Elektrotechnik), MathPAD (Mathematik), Lückel (Maschinenbau) und Rammig (Informatik) getragen. Allerdings wurde der Name, nicht aber die Aufgabenstellung, inzwischen geändert in *Institut für Automatisierung und Instrumentelle Mathematik*, kurz in *AUTOMATH*. Der ursprünglich vorhandene doppelte Bezug zur Mathematik klang nun doch etwas anstößig, denn wie bei so manchen Dingen im Leben gilt auch für die Mathematik, daß es zwar jeder tut, aber kaum einer auszusprechen wagt.

Die Hauptaufgabe des neuen Instituts ist die Entwicklung elektronischer Werkzeuge zur Simulation technischer Vorgänge und Dynamischer Abläufe, zum Entwurf mechanischer und elektronischer Systeme, sowie zur symbolischen und logischen Verarbeitung technisch-wissenschaftlicher Sachverhalte. Bei Betrachtung der beteiligten Arbeitsgruppen hoffen MathPAD-Leser natürlich, daß *AUTOMATH* sie eines Tages mit einem Chip zur symbolischen Verarbeitung beglückt. Doch vor den Preis haben die Götter natürlich die Gremien gesetzt.

Zur Freude der Gründer haben die beteiligten Fachbereiche inzwischen dem Projekt zugestimmt, sodaß zumindest die schon im Wintersemester 90/91 angelaufene *AUTOMATH*-Vorlesungsreihe nachträglich sanktioniert ist.

Die Graphik des Titelbildes beleuchtet die gegenwärtige Situation recht gut: *AUTOMATH* überwindet den freien Fall in die Bedeutungslosigkeit durch geschickte Ausnutzung der Eigendynamik und erzielt damit eine Kompensation vorhandener Beharrunskräfte. Kenner wissen allerdings, daß ein solch geostationärer Orbit eine delikate Gleichgewichtslage ist, die durch kleine Störungen letzten Endes zum spurlosen Verglühen führt. Nicht zu übersehen ist, daß zumindest zwei der symbolischen *AUTOMATH*-Gründer schon lethargisch und nahezu bewegungslos in der Kapsel des Fortschritts liegen. Das Gerücht, daß sie durch Verhandlungen mit Rektorat, KPF und FK in diesen Zustand versetzt wurden, ist allerdings übertrieben. Dies kann schon deshalb nicht wahr sein, weil sich ihr Gemütszustand nur unwesentlich von dem ihres freischwebenden Kollegen unterscheidet, der ja die Freiheit des Universums nur scheinbar genießt, wie seine maskenhaft roboterhaftige Haltung deutlich zeigt, und dessen unsichtbare Gedanken von argen Zweifeln bezüglich der Verwirklichung dieser Initiative geprägt sind. Auch er hat erfahren daß man selbst in der Freiheit des Alls den lethargischen Beharrungkräften der Erde nicht entrinnen kann. Freude ist ihm nur wegen einer neuen naturwissenschaflichen Erkenntnis anzumerken. Dachte er nämlich bisher, daß die Erzeugung gravitativer Beharrung homogen von der Erdmasse insgesamt ausginge, so weiß er nun, daß die Zentren dieser Naturkraft punktuell auf einige organisatorische Strukturen im ostwestfälischen Raum konzentriert sind.

Workshop Computer Algebra Systeme

Waldemar Wiwianka

Dies ist ein Bericht über den Workshop für Computer Algebra Systeme, der am 13. und 14. 12. 1990 in Essen im Institut für Experimentelle Mathematik stattfand. Der Workshop, an dem ca. 100 Wissenschaftler teilgenommen haben, wurde von Dr. Gerhard Schneider (Essen) organisiert. G. Schneider hat es geschafft, sechs hochkarätige internationale Referenten für die wichtigsten Computer Algebra Systeme einzuladen. Der Workshop war so organisiert, daß man nach jedem Vortrag, in dem sowohl Folien gezeigt wurden, als auch eine Demonstration des Computer Algebra Systems stattfand, viel Zeit hatte, um sich mit den Computer Algebra System intensiv zu befassen und Fragen an die Referenten zu stellen.

Tagungsprogramm

B. Trager (IBM Yorktown) : Scratchpad II
R. Maeder (ETH Zürich) : Mathematica
G. Schneider (Essen) : Cayley
J. Davenport (Bath) : Reduce
M. Monagan (Waterloo) : Maple
M. Schönert (Aachen) : GAP

Im einzelnen:

Scratchpad II

Scratchpad II ist ein Computer Algebra System, das sowohl einen Interpreter als auch einen Compiler zur Verfügung stellt. Man hat also die Möglichkeit, mittels der Programmiersprache und den abstrakten Datentypen Programme zu schreiben, sie zu kompilieren und dann im Interpretiermodus damit, wie mit einem Systemprogramm, zu arbeiten. Barry Trager ging in seinem Vortrag auf einige Besonderheiten von ScratchPad II ein. Z.B., daß die Integration immer eine Antwort liefert, d.h. entweder es existiert die Stammfunktion, dann wird sie ausgegeben, andernfalls existiert sie nicht. Desweiteren besitzt Scratchpad ein HyperTeX Helpsystem für

- Basic Commands
- Live Tutorials
- Browser

Für die Grafik bietet Scratchpad II Funktionen für das Plotten von 2-D und 3-D Grafiken an. Mittels des Grafik Windows und der dazugehörigen Buttons kann man die Grafik manipulieren, d.h. rotieren, den Standpunkt und die Perspektive ändern, die Farbe verändern und vieles mehr. Eine Besonderheit ist das Plotten von impliziten Funktionen. Hierbei werden alle Komponenten der Grafik gefunden. Die Algorithmen hierfür sind eine Kombination aus symbolischen und numerischen Algorithmen.

Scratchpad II läuft unter AIX auf den neuen WS RS6000, dem RT und IBM Mainfraims mit mindestens 16 MB Hauptspeicher. Außerdem werden ca. 70 MB Plattenspeicher benötigt. Wohl einer der größten Nachteile bei Scratchpad II ist der, daß es nur auf IBM Rechnern läuft.

Mathematica

Roman Maeder stellte Mathematica in Form eines Verkaufsvortrages vor. Nachdem er die Funktionalität an Hand von Beispielen vorgestellt hatte, wies er auf einige Besonderheiten von Mathematica hin.

- Die *Grafik* bedient sich adaptiver Algorithmen. d.h. wenn es erforderlich ist, wird die Anzahl der Stützstellen erhöht, um Unstetigkeiten oder Besonderheiten der Grafik besser herauszustellen.

- *Eindeutigkeit der Klammern* In Mathematica werden die Klammern nur für einen Zweck benutzt. Dieses hat den Vorteil, daß man bei der Eingabe das Multiplikationszeichen vergessen kann. Die Klammern werden für die folgenden Zwecke benutzt:
  [ ] für Funktionsaufrufe
  ( ) für Klammern in arithmetischen Ausdrücken
  { } für Listen.

- *Mathematisches Programmieren.* Die Definition einer Funktion kann durch Angabe einer Reihe von Regeln erfolgen. Beispiel: Hier wird eine Funktion *log* in Anlehnung an die Regeln für den

Logarithmus definiert.

$$log[x\_ y\_] := log[x] + log[y]$$

$$log[x\_\ n\_] := nlog[x]$$

Die Eingabe von

$$log[a\, b\, c^2\, d]$$

liefert dann

$$log[a] + log[b] + 2\, log[c] + log[d]$$

- Mathematica besteht aus ca. *200.000 Zeilen C-Code* und man hat 100 Mannjahre in die Entwicklung von Mathematica investiert.
- In der ersten Hälfte des Jahres '91 soll die neue, verbesserte *Version 2.0* verfügbar sein [1]

Maple

Michael Monagan berichtete, daß zur Maple Entwicklergruppe ca. 12 Personen gehören und daß im Maple Umfeld 5 PhD - und 25 Master - Thesen entstanden sind. Desweiteren hat Maple es bis heute auf 1500 Installationen gebracht, wovon 80 % im Akademischen Bereich zu finden sind. Im 1. Teil des Vortrages ging M. Monagan auf die Design Ziele von Maple

- kleiner Kern
- große Biblothek
- Platz- und Zeit-Effizienz

ein. Die Sprache von Maple ist als Programmiersprache für Mathematiker entwickelt worden. Wenn man z.B. eine Funktion definieren möchte, kann man es auf die folgende Art und Weise tun:

$$f := x \rightarrow sin(x)/x2$$

Ruft man nun $f(2)$ auf, so erhält man $\frac{sin(2)}{4}$. Monagan machte noch einmal auf die Tatsache aufmerksam, daß Maple aus 3 Teilen besteht, dem Kern, der Library und Iris. Iris beinhaltet den Parser, die Grafik und den Prettyprinter und hat ca. 20.000 Zeilen C-Code. Zum Kern gehören der Interpreter, die Speicherverwaltung, Datenstrukturen usw. und er besteht aus 16.000 Zeilen Code. Die Library enthält die Applikationen und hat zur Zeit 130.000 Zeilen Maple-Code. (Die neuen Library Pakete können sie im Artikel über Maple V auf Seite .. finden.)

[1] Der Bericht: *New Features in Version 2.0* kann bei mir via e-mail (waldemar@uni-paderborn.de) angefordert werden.

Den 2. Teil seines Vortrages leitete Monagan mit dem Satz

" Space is more scarce than time "

ein. Er erläuterte, daß dies sowohl für Systemprogramme als auch für Library Programme gilt. In diesem Teil ging er auf Einzelheiten von Maple ein. Er erläuterte sowohl einige Algorithmen (simplify, gcd), die unique representation, als auch die interne Darstellung der einzelnen Datenstrukturen (integer, rational, symbol, sum, sequenz, set). Desweiteren ging er auf die hash tables, die schnelle Substitution und die Garbage Collection, ein.

Reduce

Auch für Reduce hatte es die Tagungsleitung geschafft, einen sehr interessanten Referenten zu bekommen. Es war J.Davenport, den man durch seine vielen guten Artikel und Algorithmen aus dem Bereich der Computer Algebra kennt. In seinem Vortrag gab Davenport viele Quellen an, in denen man etwas über die Algorithmen findet, die von Reduce und anderen Computer Algebra Systemen benutzt werden. Ich gebe hier nur einige Beispiele an (was auch damit zusammen hängt, daß Davenport diese Folie sehr schnell wieder wegpackte).

- Springer LNCS 72: Hearn subresultant gcd
- Proc SYMSAC 81: p-adic (gcd, exgcd)
- Springer LNCS 162: Rich-Norman Methode
- Springer LNCS 102: Integration von algebraischen Funktionen
- Springer LNCS 432: Irena System

Das Paket zur Integration von algebraischen Funktionen kann man in Reduce mittels des Befehls
on algint dazuladen.
Das Irena System ist die Verbindung von Reduce und der Nag-Library für advanced numerics.

GAP

Martin Schönert verzichtete in seinem Vortrag auf die allgemeinen Worte zum Thema " Was ist Computer Algebra " und gab stattdessen einen ausführlichen Bericht über GAP. Dieser enthielt sowohl etwas über die Entstehung von GAP als auch Einzelheiten über die Programmiersprache (listen, sets usw.), die Kontrollstrukturen, die Library und viele spezielle Funktionen.

Der Umgang mit der Datenbank math (2.Teil)

Wend Werner

Ein aufmerksame Beobachter des Zeitgeschehens wird festgestellt haben, daß vieles fließt, daß kaum etwas so bleibt, wie es einmal war, und daß Dinge, an deren Vorhandensein man noch am Abend fest geglaubt hat, am Morgen plötzlich weg sind. So ist es auch bei uns : Der Zugang zur Datenbank **math** ist Spiel unbändiger Gezeitenkräfte und so, wie wir noch unlängst über ihn berichteten, funktioniert er bereits nicht mehr.
Darum kurz eine Zusammenfassung des derzeitigen Standes, obwohl man ruhig damit rechnen sollte, daß sich hier bald wieder etwas ändert. Also, **script** starten, einloggen auf **baldur** (Achtung: Das ist einer der Punkte, die sich ständig ändern und zwar immer, ohne daß man weiß, wann oder warum.), von dort aus die FIZ-Nummer anwählen, und nun, auch das gewiß technisch der letzte Schrei, meldet sich neuerdings das HRZ zu Wort und verlangt ein Password. (Wenn Sie schwören, daß das unter uns bleibt, es lautet *hrzuser*.) Sodann geht alles weiter, wie unlängst an dieser Stelle besprochen: Eingabe eines <**return**>, eines **x**, sodann Login-Identität und Password sowie schließlich eine 3. Die Tatsache, daß Sie sich jetzt hoffentlich demnächst im file **home** wiederfinden werden, ist eine andere Geschichte, die wir gleich besprechen werden. Nehmen wir für den Moment einmal an, Sie erhalten bereits eine entsprechende Meldung. Dann versetzt Sie die Eingabe

```
file math
```

in den Zustand, von dem wir bisher immer ausgegangen waren. Wie auch immer, wir sind bereit für die

Zweite Sitzung

Ein Bekannter hat es satt, die Bücher von *S. Lang* auf Englisch zu lesen. Er bittet Sie um eine Liste, die alle in Übersetzungen erschienenen Bücher dieses Autors samt der jeweils benutzten Sprache aufführt. Was wäre zu tun ? Sicherlich sollte man zunächst nach den Einträgen suchen, für die Serge Lang als Autor verantwortlich zeichnet. Zu diesem Zweck benutzen wir ähnlich wie (im 1. Teil) **/cc** zur Darstellung der AMS-Klassifikation das Suffix **/au**:

```
s Lang/au
```

Zur Erinnerung: `s` ist hier nicht die versehentlich klein geschriebene Abkürzung für “Serge” sondern bedeutet die Aufforderung, nach dem nachfolgenden Begriff in **math** zu suchen. Wir erhalten die überraschende Antwort:

```
L1 0 LANG/AU
```

Was geht hier vor ? Um die Antwort auf diese Frage etwas verständlicher machen zu können, sehen wir uns zunächst noch einmal einen typischen **math**-Eintrag an:

```
AN  645.46035 MATH
TI  How many wood would a woodchuck chuck ?
AU  Chuckoo, M. L.
SO  Proc.  US-Swed.  Semin.  in honour of
    Lars Gög, Lund /Swed., Irretrievable
    Results Ser.  12, 99-114 (1988).
DT  Conference article
LA  English
AB  [For the entire collection see Zbl.
    633.00013.]  The author presents a
    marvellous collection of superb and
    previously unreleased material on
    the theory of ill-posed problems.
        (J. Povar)
CC  65-02 Advanced exposition
    65C20 Models, numerical methods
    68K05 Miscellaneous appl.  of computers
UT  Ill-posed problems; Learning spaces;
    Zbl.  633.00013.
```

Dabei sollten Sie Ihr Augenmerk auf die am linken Rand erschienenen Großbuchstaben richten. Sie typisieren die verschiedenen Einträge auf der - sagen wir - Karteikarte, die einen Eintrag in der Datenbank **math** ausmacht. Schicken wir den Archivar auf die Suche, ohne ihm diesbezüglich genauere Angaben zu machen, so wird er bei seinen Nachforschungen nur die Felder berücksichtigen, die zum sogenannten **Basic index (/bi)** gehören. Es sind dies die Felder mit den Kennungen **TI**, **AB**, **CC**, **UT** sowie der **preview**-Indikator, den wir bereits im letzten Teil benutzt haben. In all diesen kann grundsätzlich nach allen vorkommenden Wörtern gesucht werden mit Ausnahme von einigen Stoppwörtern wie “the”, “or” oder “and”, die entweder zu oft vorkommen oder bereits eine andere syntaktische Bedeutung haben. Mit unserer letzten Anweisung hatten wir nun die Suche auf das Autorenfeld (**AU**) beschränkt. Und hier ist der Punkt: Sobald man den **search**-Befehl durch **/au** modifiziert, verliert er an Macht. Gesucht werden kann dann nämlich nur nach vollständigen Einträgen, also solchen, die untereinander durch ein Semikolon getrennt sind. Diese Eigenschaft teilt **AU** im übrigen mit allen anderen Datenfeldtypen mit Ausnahme derer, die zu **BI** gehören. Nun kann man freilich gar nicht immer so genau wissen, welche Einträge überhaupt auftreten. So kennen Sie vielleicht den vollständigen Namen eines Autors nicht, oder es kommt vor, daß das im Zentralblatt verwandte Kürzel für den Namen der Zeitschrift, an deren Jahresinhaltsverzeichnis Sie interessiert sind, etwas anders lautet als es vielleicht naheliegen würde. Für all diese Wiedrigkeiten im Leben des Datenbankbenutzers gibt es ein Gegengift: Das **e(xpand)** - Kommando. In unserem Fall benutzt man es so:

```
=> e lang, s.  /au
```

Als Antwort erhält man jetzt gewöhnlich diejenigen dieser vollständigen Einträge, die sich alphabetisch in der Nachbarschaft des expandierten Begriffs befinden:

```
E1    7          LANG, REINHARD/AU
E2    2          LANG, ROBERT/AU
E3    9   -->    LANG, S/AU
E4    2          LANG, S D/AU
E5    1          LANG, S R/AU
E6   85          LANG, SERGE/AU
E7    2          LANG, SHEAU DONG/AU
```

Dieses Mal hatten wir mehr Glück. Unser Lang findet sich unter **e3** und **e6**. Um diese Einträge zugänglich zu machen, ist ein **search** - Aufruf nötig.

```
=> 2 s e3 or e6
```

Dieser führt schließlich zu

```
L2   90   LANG, S"/AU OR LANG, SERGE"/AU
```

Um unser Ziel zu erreichen benötigen wir noch einige der anderen Datenfeldtypen. Der Übersichtlichkeit halber stellen wir an dieser Stelle gleich alle vor. Im einzelnen bedeutet

AN Accession Number (Die Nummer des Zentralblattreferats)
TI Das ist offensichtlich der Titel
AU Author(s), Editor(s)
SO Source
DT Document Type (z. B.: *B(ook)*, *C(onference)A(rticle)*, *P(re)P(rint)*, *J(ournal)*, *D(issertation)*)
LA Language (etwa *AZ(zerbaijani)*, *CA(talan)*, *EN(glish)*, *E(sperant)O*, *FR(ench)*, *RU(ssian)*)
AB Abstract Text (Das Zentralblattreferat)
CC Classification Codes (Die Klassifizierung nach dem AMS-Schema)
UT Uncontrolled Terms (“Key-Words and phrases”)
UP Diese Feldkennung bezieht sich auf den Zeitpunkt, zu dem der Eintrag in **math** vorgenommen wurde (also etwa 900224/UP für einen Update am 24.2.90). Dies ist ein wichtiger Parameter für den Fall, daß Sie zum Beispiel grundsätzlich nur an den Publikationen interessiert sind, die nach der Wiedervereinigung in die Datenbank aufgenommen worden sind.
PY Publication Year
RV Reviewer

In den Feldern **UP** und **PY** kann übrigens auch mit Hilfe der numerischen Operanden $<, <=, >, >=$ und $=$ gesucht werden (z.B. py=1988, 880101<=up<=890104 oder auch 1974-1980/py).
Um ein wenig mit unserer anfangs gestellten Aufgabe voranzukommen, benutzen wir jetzt das Feld **DT**, um die Bücher aus L6 auszusondern:

```
=> s l2 and b/dt
```

Das macht 31653 Bücher, davon 50 von Lang:

```
         31653   B/DT
    L3      50   L2 AND B/DT
```

Und da es ja unbedingt die Übersetzungen sein mußten:

```
=> s l3 not english/la
```

Es bleiben nur noch 9 übrig,

```
        621724   ENGLISH/LA
    L4       9   L3 NOT ENGLISH/LA
```

und die wollen wir samt Sprache alle haben:

```
=> d ti la 1-9
```

Wir verzichten auf die Wiedergabe des Ergebnisses.

In dem nun folgenden Teil der Sitzung wolen wir nachsehen, was die Routine SDI seit dem letzen Mal für uns getan hat. Da Sie gewiß vergessen haben, welches Profil wir gesichert hatten, sehen wir noch einmal nach:

```
=> d saved/s
```

Wir benutzen also den **display** - Befehl, um alle gesicherten ("saved") SDI-Profile ("/s") auszugeben. Die Antort besteht in unserem Fall in

```
NAME  CREATED    NOTES/TITLE
----  ---------  -------------------------
AP/S  24 AUG 90  SDI REQUEST FOR FILE MATH
```

Um zu sehen, was dieser Auftrag zum Ziel hatte oder um das gespeicherte Suchprofil in anderer Weise zu benutzen, kann man den **act(ivate)** - Befehl benutzen:

```
=> act ap/s
```

Es sind jetzt die alten Suchprofile wieder bekannt, ohne daß eine Suche bereits durchgeführt worden wäre (ähnlich also wie bei der im letzten Teil besprochenen Benutzung des Befehls **save** zusammen mit dem Suffix **/q**).
Wir benutzen nun ein weiteres Mal den **display** - Befehl, dieses Mal um nach den gesicherten Antworten auf unsere SDI-Suche zu fragen.

```
=> d saved/a
```

Wir erhalten so eine Mitteilung über die Anzahl,

```
NAME  CREATED    NOTES/TITLE
----  ---------  ------------------------
AP/S  24 AUG 90  18 ANSWERS IN FILE MATH
```

Für die tatsächlich gefundenen Antworten benutzt man ebenfalls den **act(ivate)**-Befehl:

```
=> act ap/a
```

Die Ausgabe lautet hier

```
L9       QUE  APPROXIMATIONSEIGENSCHAFT
L10      QUE  PROPRIETE?(1W)APPROXIMATION
              OR APPROXIMATION PROPERTY
L11      QUE  L1 OR L2
L12      QUE  L3 AND (46 OR 47)/CC
L13   18 SEA  L1 AND 901012/UP
```

Der Antwort können Sie übrigens auch entnehmen, wie man mit Hilfe von **/up** sowie der im letzten Teil besprochenen **save**-Funktion SDI manuell nachahmen könnte.

Haben Sie genug von der Approximationseigenschaft, oder sind Sie es ganz einfach leid, daß diese Sucherei ewig so weitergeht, so entledigen Sie sich der ganzen Angelegenheit durch

```
del ap/s
```

Ach ja, man sollte der Vollständigkeit halber noch hinzufügen, daß SDI überhaupt nur funktioniert, falls der login-file **home** heißt. Versucht man es von **math** aus, passiert viel Merkwürdiges und Unerklärliches. Es hat sich herausgestellt, daß man das am besten damit erklärt, daß auch beim FIZ nicht immer alle so genau wissen, was eigentlich gespielt wird.

Zum Abschluß dieses Beitrags noch einige allgemeine Bemerkungen und Ratschläge.
Beginnen wir mit der Bewältigung einer unangenehmen Situation, in die man bei der Arbeit auch mit dieser Datenbank kommen kann. Nehmen wir an, Sie rufen den SDI-Dienst auf, haben zuvor noch keine L-Zahl definiert, und schon ist es passiert: Sie stehen am Abgrund der Unendlichkeit. (Es hilft übrigens nichts, irgendwie die Verbindung zum FIZ zu unterbrechen: Versucht man sich gleich darauf wieder einzuloggen, so befindet man sich immer noch in derselben Schleife.) Das magische Zeichen für derartige Fälle lautet

```
$
```

Sie geben es ein und nehmen damit den letzten, verhängnisvollen Befehl wieder zurück.

Falls Sie eines Tages mit Ihrem Latein gänzlich am Ende sind, so sollten Sie die **help**-Funktion ausprobieren. Sie vermittelt Erklärungen für eigentlich alle wichtigen Begriffe und ist insgesamt, insbesondere auch was die Querverweise angeht, so gestaltet, wie man es von einer solchen Funktion erwarten würde.

Zu allerletzt noch ein unserer Meinung nach gravierender Nachteil dieser mathematischen Datenbank: Sollten Sie etwa nach einer bestimmten Gleichung suchen wollen, oder es interessiert Sie die Klasse der Hardy Räume H^p, so stehen Sie etwas auf verlorenem Posten. Gerade auf dem Gebiet mathematisch formalisierter Ausdrücke existieren in **math** keine erkennbaren Standards. Die einzige Möglichkeit besteht derzeit noch darin, ein Referat aus **math** herauszuholen, welches den Sie interessierenden formalen Ausdruck enthält, sich anzusehen, wie er dort aufgebaut wird und dann wenigstens nach all den Einträgen suchen zu lassen, die den Ausdruck in derselben Form enthalten. (Damit erhalten Sie mit einiger Sicherheit nicht alle). In Zukunft soll aber, wie gewöhnlich gut unterrichtete Kreise berichten, eine Umstellung der Referate auf TEXerfolgen.

A/UX - Unix auf dem Mac

Andreas Kemper

Das A/UX - Betriebssystem ist die Apple UNIX - Version für den Macintosh. A/UX verbindet die Leistungsfähigkeit des weitverbreiteten Betriebssystems UNIX mit den Vorteilen eines Macintoshs und seiner Benutzeroberfläche. Den Kern der A/UX-Oberfläche bildet der sogenannte A/UX - Finder. Diese dem Macintosh - Multifinder ähnliche Umgebung erlaubt die simultane Verwendung von Macintosh und UNIX - Fenstern in einer gemeinsamen Oberfläche.

1 A/UX - UNIX und mehr

A/UX stellt dem Benutzer eine vollständige, auf dem UNIX System V basierende, und um weitere BSD (Berkley Software Distribution) Elemente ergänzte, UNIX - Version zur Verfügung. Diese ist durch eine Anzahl weiterer Optionen ergänzt worden, die eine Einbindung des MacOS in die A/UX - Umgebung ermöglichen und die Benutzung der Standard-UNIX-Befehle und Tools vereinfachen. Eine auf A/UX basierende X - Umgebung ist von Apple erhältlich.

Das äußere Erscheinungsbild der A/UX - Oberfläche wird durch den Macintosh Desktop mit seiner Menüleiste am oberen Bildschirmrand geprägt. Mit Hilfe von Pull-Down-Menüs, Dialog-Fenstern und Icons ist der Benutzer in der Lage mit dem System zu kommunizieren, ohne komplizierte Befehlssequenzen zu erlernen. So kann durch die Verwendung einer Commando-Dialog-Box bei vielen Befehlen auf die Eingabe komplexer Flag-Optionen verzichtet werden. Statt dessen können die Optionen eines Befehles durch Anklicken von Buttons aktiviert werden. Um den Kommando - Interpreter des Systems ansprechen zu können, muß zunächst ein CommandShell - Fenster geöffnet werden. In diesem steht sodann eine UNIX - Shell zur Verfügung. A/UX stellt drei unterschiedliche Shell - Typen zur Auswahl: die Bourne Shell, die C Shell sowie die Korn Shell.

Ist der Macintosh mit einer Ethernet - Karte ausgerüstet, so kann die NFS (Network File System) - Fähigkeit der Software voll ausgenutzt werden. So ist der Zugriff auf die Filesysteme anderer UNIX - Rechner möglich. Ebenso verfügt A/UX über die Fähigkeit den Yellow Pages Mechanismus zu nutzen. Hierbei handelt es sich um eine verteilte Datenbank zur Vereinfachung der Administration in einem UNIX Netzwerk. Das Apple Protokoll AppleTalk wird ebenfalls von A/UX unterstützt, um die Verbindung zu anderen Macintosh Rechnern mit Hilfe der in jedem Macintosh vorhandenen Kommunikationssoftware zu ermöglichen. Da auch die Verbindung zwischen einem Macintosh und einem Apple Drucker auf dem AppleTalk Netzwerk basiert, ist somit die Verwendung der Apple Drucker LaserWriter und ImageWriter II aus der UNIX - Umgebung möglich.

2 A/UX und MacOS

Das Standard-Betriebssystem des Macintosh MacOS ist in A/UX nahezu vollständig integriert. Hierdurch können aus dem A/UX-Finder heraus eine große Zahl von Macintosh Applikationen gestartet und auf gewohnte Art und Weise genutzt werden. Die Arbeitsgeschwindigkeit der MacOS-Applikationen wird durch den Aufruf aus A/UX heraus nicht deutlich vermindert. Damit ist die Verwendung solcher Programme unter A/UX eine sinnvolle Methode, die Vorteile beider Betriebssysteme gleichzeitig zu nutzen. Leider gilt diese A/UX - Kompatibilität nicht für alle MacOS Applikationen gleichermaßen. So ist zum Beispiel der Einsatz der C-Entwicklungsumgebung THINK C in der Version 4.0 nicht möglich.

Zur Erhöhung der Kompatibilität mit gewöhnlichen Macintosh Applikationen stellt A/UX zwei verschiedene Betriebsmodi zur Wahl. Hierbei handelt es sich um den 24-Bit bzw. den 32-Bit Modus. Diese Bezeichnungen beziehen sich auf die internen Adressierungsmethoden der Applikationen im jeweiligen Modus. Grundsätzlich wird im 24-Bit Modus eine höhere Kompatibilität erreicht. Jedoch steht dem Benutzer in diesem Modus nicht die gesamte Funktionalität der A/UX - Oberfläche zur Verfügung.

A/UX erlaubt sowohl die Benutzung von Macintosh Files durch UNIX Programme als auch den Zugriff von Macintosh Applikationen auf die Daten des UNIX Filesystems. Im A/UX - Finder werden beide Filesysteme mit Hilfe von Icons dargestellt. Das Kopieren der Daten von einem Bereich zum anderen erfolgt durch einfaches Verschieben der entsprechenden Symbole. Um Daten des Macintosh - Filesystems durch ein UNIX Programm nutzen zu können, müssen diese zunächst in das UNIX Filesystem kopiert werden. In der anderen Richtung ist ein Zugriff direkt möglich, da die UNIX Partition unter A/UX als gewöhnliches Macintosh - Volume behandelt werden kann.. So ist es zum Beispiel möglich, während des Arbeitens im UNIX Bereich, Macintosh Texteditoren einzusetzen und somit deren Vorteile zu nutzen.

3 Softwareentwicklung

Zur Softwareentwicklung stehen der C Compiler *cc* sowie die Fortran Compiler *f77* und *efl* zur Verfügung. Diese werden von A/UX durch eine weitere Anzahl bekannter UNIX - Tools unterstützt. *Lint*, *awk*, *sccs* und *make* sind nur einige von ihnen. Durch die Verwendung der A/UX - Toolbox erhält der Entwickler von Applikationen Zugriff auf einen großen Teil der Macintosh Toolbox Routinen. Damit ist die Entwicklung von Macintosh Programmen, die gleichzeitig vom A/UX - Finder und vom MacOS gestartet werden können, möglich, ohne auf die Tools der UNIX Umgebung zu verzichten. Diesen Programmen steht nahezu die gesamte Funktionalität der Macintosh Toolbox zur Verfügung. Eine detailierte Beschreibung der Macintosh Toolbox Routinen findet sich in *Inside Macintosh I - V* .
Um die Kompatibilität der Programme mit A/UX zu gewährleisten, sind einige Einschränkungen zu beachten. So muß die Applikation mit dem Macintosh Multifinder verträglich sein, darf keinen direkten In - oder Output verarbeiten und nur die unter A/UX zur Verfügung stehenden Routinen der Toolbox benutzen. Außerdem dürfen die acht höchstwertigen Bits in Adressen nicht zur Speicherung zusätzlicher Informationen genutzt werden, wie dies in älteren Macintosh Programmen der Fall ist.

4 Installation

A/UX ist auf allen Rechnern der Macintosh II Familie sowie auf dem SE/30 lauffähig. Der Rechner sollte mindestens mit einem Hauptspeicher von 4 MB sowie mit einer 80 MB Festplatte ausgerüstet sein. Da A/UX ein vom Macintosh Filesystem unabhängiges Filesystem benötigt, ist es erforderlich, die Festplatte zu Beginn der Installation zu partitionieren. Zu diesem Zweck sollte zuerst der gesamte Datenbestand gesichert werden. Die Partitionierung erfolgt mit dem Tool Apple HD Setup. Das Programm gehört zum Lieferumfang der A/UX Software. Durch die Partitionierung der Platte wird diese logisch in mehrere unabhängige Teilbereiche oder Partitions aufgeteilt. Apple HD Setup enthält einige Standard-Partitionierungen für verschiedene Plattengrößen, die aus einem Menü ausgewählt werden können. Diese Partitionierungen können mit Hilfe graphischer Elemente verändert und den lokalen Gegebenheiten angepaßt werden. Vor der Auswahl der geeigneten Partitionierung sollte der Benutzer die Aufteilung des zur Verfügung stehenden Speicherplatzes gründlich planen, da eine spätere Änderung eine neue Partitionierung der Festplatte und eine nochmalige Installation erforderlich machen würde. Um A/UX installieren zu können, werden Bereiche für die konventionellen Macintosh-Daten, das A/UX-Root-Filesystem, den Swap-Bereich und eventuell weitere A/UX-User-Daten benötigt. Der Umfang dieser Bereiche ist von der Größe der zur Verfügung stehenden Platte, sowie vom späteren Einsatzbereich des Rechners abhängig. Vor dem Beginn der eigentlichen Installation muß zunächst ein System auf der nun leeren Macintosh Partition aufgespielt werden.
Die Installation der A/UX Software beginnt mit dem Kopieren mehrerer Dateien der beigefügten Disketten auf die Festplatte und dem Start des Installationsprogrammes. Zu Beginn ist die Eingabe einiger Parameter erforderlich. So wird unter anderem nach den SCSI - Nummern der beteiligten Geräte, dem Festplattentyp sowie dem Installationsmedium gefragt. Die weitere Installation kann von einer CD-ROM, von Disketten oder von einem Band erfolgen. Bei Benutzung einer CD-ROM wird hierfür etwa eine Stunde benötigt. Diese Phase der Installation läuft vollautomatisch ab.
Das A/UX System ist zur Zeit in der Version 2.0 auf einem Macintosh des Rechnernetzes der Mathematik installiert. Eine vollständige Dokumentation des Systems steht ebenfalls zur Verfügung.

5 Literatur

Zum Lieferumfang der A/UX Software gehören folgende Handbücher:

Road Map to A/UX
A/UX Installation Guide
Setting Up Accounts and Peripherals for A/UX
A/UX Essentials

Außerdem sind eine weitere Anzahl von Handbüchern für Benutzer, Systemadministratoren und Entwickler erhältlich.

Maple Version V

Waldemar Wiwianka

Endlich ist sie da, die neuste Version von Maple:

We are proud to announce the availability of new version of Maple version V (for those of you waiting for version 4.4, we have renamed it version V).

Mit diesen Worten wurde Maple V von Gaston Gonnet (dem "Vater "von Maple) angekündigt. Im Vergleich zur Version Maple 4.3, die uns bisher zur Verfügung stand, hat Maple V nun endlich eine 3-dimensionale Grafik, eine X Window Schnittstelle (eine SunView/Suntools Version ist fast fertig) und eine stark vergrößerte Library.

1 X Window Interface

Der große Unterschied zwischen Maple, das unter UNIX ohne X Window und mit X Window System läuft, ist das zusätzliche Maple Fenster für Ein - und Ausgabe. Es erlaubt beispielsweise *cut and paste, copieren , editieren usw.*, wie man es von anderen Fensterumgebungen (wie z.B. MacOS) gewohnt ist. Will man einen Befehl editieren, braucht man die Maus nur an die entsprechende Position bewegen und durch Klicken den Text Cursor plazieren. Desweiteren stehen Menus und Buttons zur Verfügung, mit denen man das Programm verlassen kann oder z.B. den Output im Maple Fenster verändern kann.

2 3-dimensionale Grafik

Maple V unterstützt nun 3-D Plots unter verschiedenen Software Plattformen, eingeschlossen das X Window System. Flächen können durch eine Funktion in 2 Variablen oder durch Parametrisierungen beschrieben werden. Das Koordinatensystem kann kartesisch, sphärisch oder zylindrisch sein. Die Fläche, die gezeichnet wird, kann aus Flächenstücken (Patches), Punkten oder als Drahtmodell, optional mit Hidden Line, gezeichnet werden. Eine Anleitung mit vielen Beispielen ist innerhalb von Maple mittels des Befehls:

$$help(plot3d)$$

verfügbar. Arbeitet man mit Maple unter dem X Window System, so generiert Maple V nach Aufruf von

$$plot3d(...)$$

ein separates Plot Fenster, indem man die Grafik manipulieren kann. Man kann z.B. den Blickwinkel verändern, den Hidden Line Algorithmus ein- oder ausschalten, die Grafik rotieren lassen, die Projektion verändern usw. . Alle diese Aktionen können per Maus und Menus vorgenommen werden.

3 Library

Hier wurden sowohl neue Library Funktionen, neue Library Paket und auch Funktionen in vorhandene Libraries hinzugefügt. Im folgenden möchte ich hier nur einige Beispiele angeben (entnommen aus: *New features that have been added to Maple for version V*)

<u>New library functions</u>

ellipsoid: surface area of an ellipsoid
evalr: evaluation over a range(s)
fnormal: floating point normalization
galois: computes the galois group of a (univariate) polynomial (up to degree 7)
history: a history mechanism (an alternative to Maple "s)
iratrecon: rational reconstruction
mellin: mellin transform
poisson: poisson series expansion
irreduc: irreducibility test of a (multivariate) polynomial
roots: computes the roots of a (univariate) polynomial
shake: real interval arithmetic (to Digits precision)
sturm: use the Sturm sequence to find the number of real roots in an interval
sturmseq: compute the Sturm sequence of a polynomial over Q or R
thiele: continued fraction interpolation
userinfo: generate user information
ztrans, invztrans: the Z-transform and it's inverse

C: Maple to C language translator
FFT: Fast Fourier transform
Hermite: computes the Hermite normal form of a matrix over a finite field
Irreduc: irreducibility test of a (univariate) polynomial mod p
MOLS: generates mutually orthogonal Latin squares
Nullspace: computes the nullspace of a matrix over a finite field
Primitive: tests if a univariate polynomial over a finite field is primitive
Randpoly: random univariate polynomial of degree k mod p
Randprime: random monic irreducible univariate polynomial of degree k mod p
Smith: computes the Smith normal form of a matrix over a finite field

New library packages

bianchi: 3-d Lie algebra classification package
cartan: special-purpose relativity tensor package–Cartan's formalism
debever: special-purpose relativity tensor package–Debever's formalism
finance: amortization and Black Scholes stock call option pricing
geom3d: 3-d geometry package
heap: priority queue data structure
liesymm: Lie symmetries package – obtains the determining equations for the Lie symmetries of a system of partial differential equations
logic: boolean algebra package
oframe: orthonormal frame package using the Ellis-MacCallum formalism
petrov: Weyl tensor classification package
plots: 2-d and 3-d special plots package
tensor: relativity tensor package

New functionality in library packages

linalg: linear algebra package
combinat: combinatorial functions package
numtheory: number theory package
stats: statistics package

4 Verfügbarkeit

Maple V ist zur Zeit auf allen SUN und MIPS Workstations verfügbar. Für DEC WS und den MAC[1] wird es gerade portiert. Danach will man Maple V für viele andere Systeme anpassen. Um nun auf unseren SUN WS mit Maple V zu arbeiten, ruft man einfach

$$maple$$

oder unter X Window (bzw. Open Windows)

$$xmaple$$

auf. Für alle, die noch nicht so vertraut sind mit X Window (oder Open Windows), ein kleiner Tip. Man kann *xmaple* auch auf einer leistungsfähigen WS starten und den Output auf der lokalen WS (server) anschauen. Dazu müssen folgende Befehle eingegeben werden: Auf der lokalen WS (hier kepler) gibt man den Befehl

$$xhost\ \ planck \qquad oder \qquad xhost\ +$$

ein.
Dieses bewirkt, daß *planck* bzw. bei + alle Rechner eine Verbindung zum X server (hier kepler) aufbauen können. Auf *planck* setzt man die Enviroment Variable *DISPLAY* auf den folgenden Wert

$$setenv\ \ DISPLAY\ \ kepler : 0$$

Nun kann man auf den Client (planck) Maple starten indem man *xmaple* aufruft.

Die komplette Beschreibung : *New features that have been added to Maple for version V*
kann bei mir angefordert[2] werden oder mittels des Befehles :

$$help(updates, v5)$$

bzw.

$$?updates, v5;$$

in der Maple Sitzung abgerufen werden.

[1] Die neue Mac Version soll auch den Notebook Mechanismus (wie bei Mathematica) enthalten.
[2] Z.B. per e-mail an waldemar@uni-paderborn.de

Briefe im Namen der Universität

Karsten Morisse

Mit dem LaTeX–Dokumentstil `letter_uni` ist es jetzt direkt möglich, ein oder mehrere Briefe im üblichen Uniformat zu schreiben. Die Anwendung funktioniert genauso, wie bei dem im LaTeX–Paket mitgelieferten Dokumentstil `letter`. Die Ausgabe ist so formatiert, daß sie einem Briefformular der Universität entspricht. Was leider fehlt ist die blaue Farbe des Unilogo's .

Zum Eintrag in die Felder im Briefkopf stehen verschieden Befehle bereit, so daß sich jeder Benutzer einen persönlichen Kopf generieren kann. Möglich sind die folgenden Kommandos:

- `\name{}` — Benutzername
- `\signature{}` — Unterschrift
- `\room{}` — Zimmernummer des Büro's
- `\mail{}` — Elektronische Mailadresse
- `\telephone{}` — Durchwahl innerhalb der Universität, d.h. ohne 60

Diese Kommandos sollten bei jedem Brief mit angegeben werden, da ansonsten der Briefkopf unvollständig ist. Optional sind die folgenden Kommandos:

- `\subject{}` — Betrifft – Zeile
- `\yref{}` — *Ihr Zeichen* – Referenz auf einen erhaltenen Brief
- `\ymail{}` — *Ihr Brief vom* – Referenz auf einen erhaltenen Brief
- `\myref{}` — *Mein Zeichen* – Eigene Kürzel
- `\date{}` — Beliebiges Datum. Fehlt dieses Kommando, so wird das aktuelle Tagesdatum eingesetzt.
- `\specialmail{}` — Erscheint über der Anschrift im Sichtfenster. Z.B. für *Eilbrief* oder *Luftpost*

All diese Kommandos erscheinen zwischen der Anschrift und dem eigentlichen Brieftext. Weiterhin gibt es noch zwei Kommandos, die am Ende eines Briefes wirken:

- `\cc{}` — Verteiler
- `\encl{}` — Anlagen

Als Beispiel ist nachfolgend ein Brief im LaTeX-Source und auf der nächsten Seite in der fertigen Version abgedruckt.

```
\documentstyle[11pt,german]{letter_uni}
% Als Option 'german' => Deutscher Brief
% ohne 'german' => Englischer Brief
\name{Karsten Morisse} \signature{K.Morisse}
\room{D2.316} \mail{kamo@uni-paderborn.de}
\telephone{3203}
\subject{Abbonement der Zeitschrift MathPAD}
\yref{BF \& WW} \myref{KM}
\ymail{13. Januar 1991}
\date{\today} \specialmail{Eilbrief}
\begin{document}
\begin{letter}
{MathPAD\\
Abteilung: Leserservice\\
Warburger Stra\ss e 100\\
4790 Paderborn
}
\opening{Sehr geehrte Damen und Herren,}
mit Begeisterung erwarte ich jede
Neuerscheinung Ihrer wunderbaren
Zeitschrift. Wie Sie es schaffen,
mit so geringem Umfang so umfassend
zu Informieren, bringt mich mit jeder
Ausgabe wieder auf's neue zum Erstaunen.
Aus diesem Grund bitte ich Sie, mir
MathPAD zuk\"unftig im Abbonement
zuzustellen. Somit verpasse ich keine
Ausgabe, und ich mu\ss\ mich nicht selbst
um den Erwerb des Journals k\"ummern.\\
Vielen Dank im voraus.
\closing{Mit freundlichen Gr\"u\ss en}
\end{letter}
\end{document}
```

Man kann in einem Textfile auch mehrere Briefe schreiben. Zu diesem Zweck ist nur der Teil zwischen `\begin{letter}` und `\end{letter}` mehrfach aufzuführen.

mathPAD

filmdraw : eine weitere Grafikroutine

Thorsten Schulze

In Kürze wird das vorhandene Grafikpaket *C_GRAPH*, welches aus den Programmen *accdraw* und *ddraw* besteht, um eine neue Routine, das Programm **filmdraw** erweitert. Dieses dient, wie der Name schon vermuten läßt, zum Erzeugen von Bildsequenzen. Zur Realisierung dieses Vorhabens wurde die vorhandene Routine *acc_draw.c* so modifiziert, daß jetzt durch einen einzigen Aufruf mehrere Grafiken hintereinander erzeugt werden können. Dazu werden wie bei den Programmen *accdraw* und *ddraw* zwei Dateien benötigt, zum einen die Parametrisierung **object.c** und zum anderen ein Eingabefile **object.inp**, die sich auf folgende Weise von den bisher verwendeten Dateien unterscheiden :

1.) In der Parametrisierung **object.c** müssen folgende Parameter vereinbart werden :

extern double t_value,bx1,by1,bz1;

Die Bedeutung dieser Parameter ergibt sich durch :

t_value : ein Parameter, von dem letztendlich alle Änderungen der einzelnen Grafiken untereinander abhängen werden.

bx1 : die in der Eingabedatei festgelegte x – Koordinate des Benutzerstandpunktes.

by1 : die in der Eingabedatei festgelegte y – Koordinate des Benutzerstandpunktes.

bz1 : die in der Eingabedatei festgelegte z – Koordinate des Benutzerstandpunktes.

Ferner ist es notwendig, drei Funktionen zu vereinbaren, die die Koordinaten des Benutzerstandpunktes **(bx1,by1,bz1)** an das Hauptprogramm übergeben. Diese Funktionen lauten :

double viewfx() { return(bx1); }

double viewfy() { return(by1); }

double viewfz() { return(bz1); }.

Durch die Vereinbarung dieser drei Funktionen ist es also möglich, den Standpunkt des Benutzers in Abhängigikeit vom aktuellen Wert für **t_value** zu verändern, indem beispielsweise

double viewfz() { return(bz1 - t_value); }

vereinbart wird.

2.) An das Eingabefile **object.inp** muß eine neue Zeile angehängt werden, die die Werte für den Bereich, den **t_value** durchläuft, sowie die Anzahl der zu zeichnenden Grafiken enthält, also eine Zeile der Form

a b c,

wobei **a,b** zwei double Werte seien und **c** ein positiver integer Wert sei. Beispielsweise würde die Zeile

0.0 1.0 20

dazu führen, daß 20 Grafiken erzeugt werden, wobei **t_value** den Bereich von **a** $= 0.0$ bis **b** $= 1.0$ in äquidistanten Schritten der Weite $\frac{b-a}{c-1} = \frac{1}{19}$ durchläuft.

Sind diese Vorbereitungen getroffen, so hat man zwei Möglichkeiten, das Programm zu aktivieren. Dabei ist zu beachten, daß der im folgenden mit **a)** bezeichnete Aufruf nur dazu verwendet werden sollte, sich einen groben Überblick über die später zu entwickelnde Bildsequenz zu verschaffen. Der im folgenden mit **b)** bezeichnete Aufruf dient dann zum Erzeugen des eigentlichen Films. Im einzelnen ergeben sich dann :

a) : **filmdraw object.c object.inp**

Durch diesen Aufruf werden die Grafiken auf dem Bildschirm erzeugt. Man hat dann jeweils

die Möglichkeit, die Grafiken mit dem **DUMP** - Button einzeln zu speichern, um sie später auszudrucken o.ä. . Der Name dieses mit **DUMP** erzeugten Rasterfiles setzt sich zusammen aus **object.pic** und der laufenden Nummer der Grafik, so erhält beispielsweise das sechste erzeugte Bild den Namen **object.pic6**.
Da dieser Aufruf nur dazu dient, sich einen groben Überblick über das Verhalten der zu plottenden Funktion zu verschaffen, werden höchstens 20 Grafiken erzeugt.

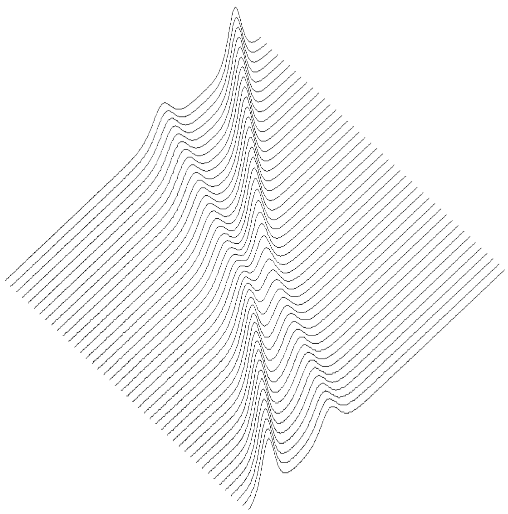


Abb. 1: 2-Soliton lösung der KdV

b) : filmdraw object.c object.inp bg

In diesem Fall werden die Grafiken nicht auf dem Bildschirm sondern intern erzeugt und folgendermaßen gespeichert :
Die erste Grafik wird in der Datei **object.anfang** komprimiert gespeichert, d.h. wir speichern jeweils die Anzahl der aufeinanderfolgenden Pixel mit derselben Farbe.
Für jede weitere Grafik speichern wir dann nur noch die Änderungen gegenüber der jeweils vorhergehenden Grafik in der Datei **object.fort**. Durch diese Speicherung schränken wir den benötigten Speicherplatz sehr ein, denn bei geeigneter Wahl des Parameters **t_value** und der Anzahl der zu zeichnenden Bilder kann man den Speicherplatz für jede Grafik (mit Ausnahme der ersten) auf 8 – 10 K einschränken, wohingegen die Speicherung eines einzelnen gesamten Bildes 128 K benötigen würde. Dabei setzen wir natürlich voraus, daß sich die zu zeichnende Funktion *gutartig* verhält, d.h. die Änderungen von Bild zu Bild sollten nicht zu groß sein.
Will man sich dann diesen Film **object** anschauen, so rufe man das Progamm

showfilm

auf, gebe den Namen des gewünschten Films an, also in unserem Beispiel **object**, und betätige den **LOAD** – Button, um die durch Aufruf **b)** erzeugten Dateien **object.anfang** und **object.fort** zu laden. Dananch betätige man den **DISPLAY** – Button, die Bildsequenz wird abgespult. Erhöhung des **repeat** – Wertes führt zu wiederholtem Abspulen des Films. Durch Verändern des **delay** – Wertes erzeugen wir kleine Pausen zwischen den einzelnen Bildern. Natürlich kann man diese Bilder auch einzeln abspeichern, indem man den **DUMP** – Button drückt und dann die Nummer der zu speichernden Grafik angibt. Der Name des Rasterfiles setzt sich ebenfalls aus dem Namen **object.pic** und der angegebenen Nummer des Bildes zusammen.

Um diese Routinen aufrufen zu können, muß man seinen Pfad in der Datei **.cshrc** um den Eintrag :

/user/gemein/C_GRAPH/FILM_DRAW

erweitern.

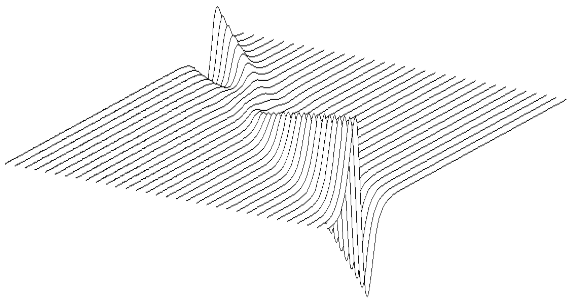


Abb. 2: Eigenvektor des Rekursionsoperators der KdV

Tagungsbericht: Computer Algebra in Engineering and Education

Karsten Morisse

Am 17. Dezember 1990 fand das Symposium *Computer Algebra in Engineering and Education* in der Technischen Hochschule Delft statt. Es nahmen ca. 120 Wissenschaftler aus den Niederlanden (ca. 85 %), Frankreich, England und Deutschland teil. Das Symposium bestand aus insgesamt 15 Vorträgen, die zum Teil in parallel abgehaltenen Sitzungen präsentiert wurden. Nebenbei waren mehrere Rechner bereitgestellt, auf denen man die Möglichkeit hatte, einige Computer Algebra Systeme auszutesten. Zur Verfügung standen:

Derive, Irena (Schnittstelle von Reduce nach NAG), Macsyma, Maple, Mathematica, Mathlab, muMath

Vorträge

Prof.Dr. A. Kerber (Universität Bayreuth): *Chemical Isomerism, the Basic Problem of Graph Theory and Combinatorial Theory of Enumeration, and a Program System for its Solution*
Der Vortrag behandelte das Problem der chemischen Isomerismen, d.h. die Existenz von chemisch verschiedenen Substanzen, die aus der selben Menge von Atomen bestehen. Das Problem führt in die Graphentheorie und die Theorie der kombinatorischen Aufzählung. Es entstammt der Tatsache, daß zu einer gegebenen chemischen Formel mehrere korrespondierende Molekulargraphen existieren können. Mittels Rechnerunterstützung können die Graphen konstruiert werden. Es wurde kurz das System MOLGRAPH vorgestellt, das diese Graphenkonstruktion ermöglicht.

Prof.Dr. F. Winkler (RISC-Linz): *Application of Computer Algebra to Problems in Technology*
Es wurden mehrere Anwendungsgebiete von Computer Algebra vorgestellt. Hierzu zählen: Robotics (Piano Movers Problem), Lösung nichtlinearer Gleichungen, Geometrische Modellierung (Behandlung geometrische Objekte wie z.B. Kurven, Oberflächen, Transformation von einer Darstellung in eine andere), Beweissysteme.

Prof.Dr.-Ing. E. Kreuzer (TH Hamburg): *Computer Algebra in Applied Dynamics*
Es wurden Gleichungen für die Bewegung von Mehrkörpersystemen betrachtet. Computer Algebra Methoden werden benutzt, um einen tieferen Einblick in die Systeme zu gewinnen.

Die Vorträge am Nachmittag waren in drei Sitzungen

Computer Algebra in Education,
Applications in Engineering,
Computer Algebra, Parallelism and Code Generation

zu je vier Vorträgen unterteilt. Wir haben die letzte Sitzung besucht.

Computer Algebra, Parallelism and Code Generation

Dr. J.L. Roch (INPG): *Parallel Models, Computer Algebra and Applications*
Es wurde ein Überblick über verschiedene parallele Algorithmen gegeben. Es wurde ausdrücklich betont, daß bei der Entwicklung von parallelen Algorithmen die spezifischen Eigenschaften des Problems berücksichtigt werden müssen. Der Titel versprach mehr, als der Vortrag hielt.

Dr. J.L. Roch (INPG): *PAC, an Environment for Parallel Algebraic Computations*
Der Vortrag behandelte das Algebra System PAC, welches die Ausführung paralleler Algorithmen erlaubt. Besonders herausgestellt wurde die Wichtigkeit einer guten Speicherverwaltung und einer dynamischen Lastverteilung. Außerdem wurden Laufzeiten für einige Probleme vorgestellt, so zum Beispiel 3 Stunden mit 32 Prozessoren für die die Berechnung der Hermiteschen Normalform einer 160×160 Matrix, oder 3 Stunden mit 16 Prozessoren für die Lösung eines dichten linearen Systems der Größe 700×700 über den reationalen Zahlen.

Drs. M.A. Fekken (TU Delft): *Parallel Computer Algebra on a Hypercube*
Es wurde ein Algebra System speziell für die Rechnung mit Poisson Reihen vorgestellt, welches auf einem Hypercube läuft. Eigenschaften des Hypercubes wurden recht ausführlich behandelt, so daß für das eigentliche Thema nur noch wenig Zeit blieb.

Dr. V.V. Goldman (UT): *Using Computer Algebra Systems to Generate Efficient Numerical Code*
Es wurden die Möglichkeiten der automatischen Programmgenerierung mittels Computer Algebra Systeme vorgestellt, insbesondere Programmcode für numerische Probleme.

Die Titel der von uns nicht besuchten Vorträge waren:

Computer Algebra in Education

Dr.B.N.P. Kamerich (KUN) *Integratie van gebruik van Maple in universitair algebra onderwijs*

Prof.Dr. F.H. Simons (TUE) *Computeralgebra en serviceonderwijs wiskunde*

Dr. F. van Schagen (VU) *Maple bij matrixrekening aan de VU*

Ir. R.E. Kooij (TU Delft) *Praktikum differentiaalvergelijkingen m.b.v muMath*

Applications in Engineering

Dr.Ir. E.J.H. Kerckhoffs (TU Delft) *Formula Manupulation in the Simulation of Fermation*

Dr.Ir. G. Hellings (Philips Nat. Lab.) *Calculation of PotentialDistributions for Energy Analysers*

Prof.Dr.Ir. P.G. Bakker (TU Delft) *Applications of Computer Algebra in Aerodynamics*

Ir. R.G.J. Janmaat (Philips Nat. Lab.) *Calculation of Symbolic and Numeric Transfer Functions of RLC-loudspeakers-crossover Filters*

Zentralblatt und Mathematical Reviews auf CD-ROM

Helmut Lenzing

Als Fremdkörper für unser homogenes SUN- und MACINTOSH-Netz haben W. Werner und E. Schalück eine MS-DOS-Umgebung erstellt, in welcher sie derzeit die Compact-Disc-Versionen der Datenbanken der Mathematical Reviews (MR) bzw. des Zentralblatts für Mathematik (Zbl) testen. Die MR-Version wird in Kürze durch die Universitätsbibliothek zugänglich gemacht; bei der Zbl-CD **CompactMATH** handelt es sich um eine Vorversion, die wir in Paderborn testen; die eigentliche Version (erstellt von FIZ Karlsruhe und Springer-Verlag) dürfte ab Mitte des Jahres vertrieben werden.

CompactMATH verfügt über eine sehr ansprechende menügesteuerte Benutzerführung (deutsch oder englisch); alternativ soll auch in direktem Kommando-Modus die Verwendung einer Teilmenge der auf dem MATH-host STN üblichen Messenger-Sprache möglich sein. (In der derzeitigen Test-Version von **CompactMath** war diese Teilmenge nahezu leer und daher kaum brauchbar.) Sie sehen hier den Bildschirm (englische Version) einer kombinierten Suche nach dem Beweis der Bieberbachschen Vermutung (siehe Abb. 1): Das Beispiel zeigt, daß bei menügesteuerter Suche – wie bei der direkten Benutzung der messenger-Suchsprache in der Datenbank MATH – eine Suche auf Felder (Autor, Titel, Abstract, ...) eingeschränkt werden kann, aber eine solche auch global möglich ist. Ferner sind (in Grenzen: ünd"geht immer) Boolesche Kombinationen möglich. Die rechte Spalte zeigt für die Komponenten der Suche und die kombinierte Suche die Trefferzahl an.

Bei der Wiedergabe derselben fällt wohltuend auf, daß Formeln als solche – und nicht als linearer Code – auf dem Bildschirm zu sehen sind bzw. im Druck erscheinen. Ferner wird — eine wesentliche Hilfe für schnelles Durchforsten großer Referatemengen — der jeweilige Suchbegriff durch Invertieren hervorgehoben (siehe Abb. 2).

Weder die on-line Suche in MATH noch die Recherche der MR-CD bieten ohne weiteres diese Möglichkeit. Zum Vergleich derselbe Text in MATH (siehe Abb. 3). Die Handhabung der MR-CD erfolgt ähnlich wie der online-Zugriff auf MATH im direkten Kommando-Modus.

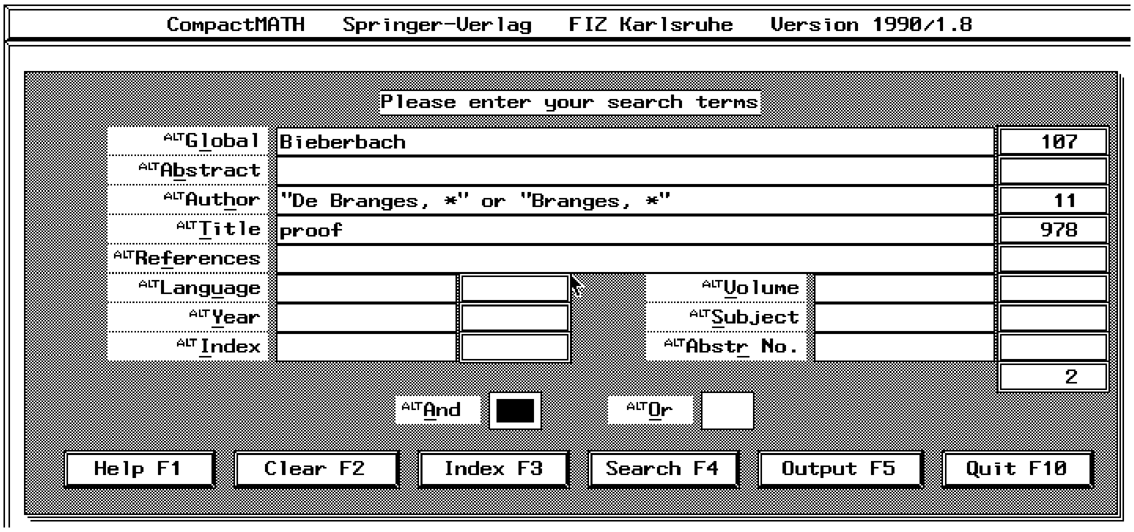
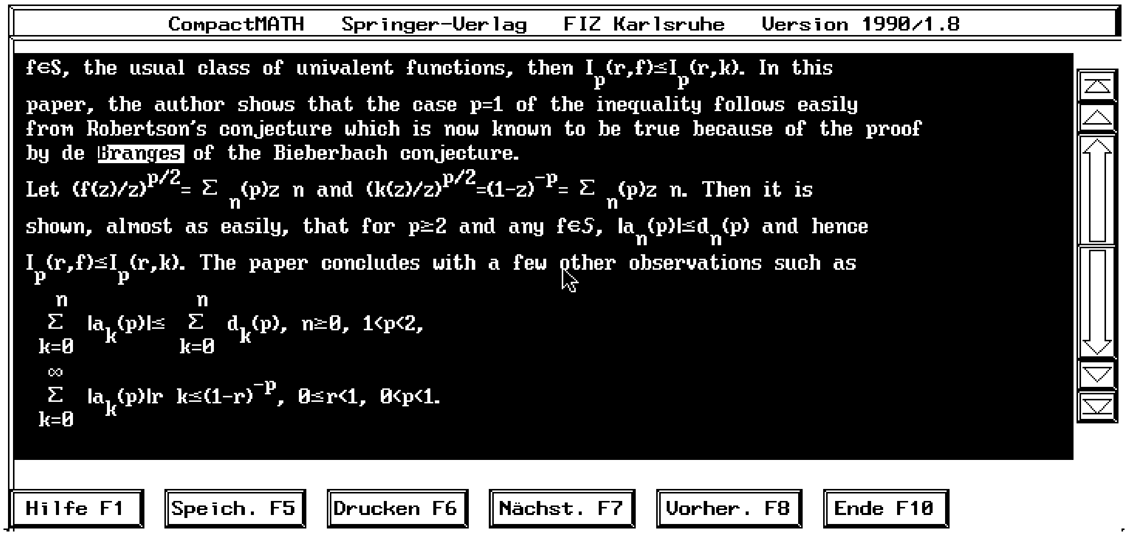


Abb. 1: Kombinierte Suche



CompactMATH Springer-Verlag FIZ Karlsruhe Version 1990/1.8

f∈S, the usual class of univalent functions, then $I_p(r,f) \le I_p(r,k)$. In this
paper, the author shows that the case p=1 of the inequality follows easily
from Robertson's conjecture which is now known to be true because of the proof
by de Branges of the Bieberbach conjecture.
Let $(f(z)/z)^{p/2} = \Sigma\ _n(p)z\ n$ and $(k(z)/z)^{p/2} = (1-z)^{-p} = \Sigma\ _n(p)z\ n$. Then it is
shown, almost as easily, that for p≥2 and any f∈S, $|a_n(p)| \le d_n(p)$ and hence
$I_p(r,f) \le I_p(r,k)$. The paper concludes with a few other observations such as

$$\sum_{k=0}^{n} |a_k(p)| \le \sum_{k=0}^{n} d_k(p),\ n \ge 0,\ 1<p<2,$$

$$\sum_{k=0}^{\infty} |a_k(p)|r\ k \le (1-r)^{-p},\ 0 \le r<1,\ 0<p<1.$$

Hilfe F1 | Speich. F5 | Drucken F6 | Nächst. F7 | Vorher. F8 | Ende F10

Abb. 2: Formeln; hervorgehobener Suchtext

Die Suchsprache ist hier nicht Messenger, sondern eine in ihrer Syntax und ihrer Mächtigkeit

- Sucheinschränkung auf Felder (Autor, Titel, Abstract,...),
- Boolesche Operationen,
- Trunkierung (Verwenden von wildcarts = Platzhaltern,...)
- Nachbarschaftssuche (Beispiele: zwei Suchbegriffe direkt aufeinanderfolgend oder im selben Absatz, ...)

vergleichbare Sprache. Formeln werden als linearer TEX-Code, genauer in AMS-TEX wiedergegeben. Dies ist hinderlich bei der direkten Beurteilung erzielter Treffer, ist jedoch ein großer Vorteil bei Import größerer Referatemengen auf die eigene Maschine und die dortige, bei TEX ja weitgehend systemunabhägige Weiterverarbeitung (Durchsehen unter texview"; Drucken).

In **CompactMath** kann man einzelne Referate ausdrucken aber – jedenfalls in der Testversion – nicht ganze Trefferserien auf einmal auf die Maschine übernehmen.
Die Datenbestände von Mathematical Reviews bzw. Zentralblatt sind zu umfangreich, um auf einer CD Platz zu finden (trotz des gewaltigen Speicherumfangs von ca. 600 MB pro CD). Durchsucht wird gleichwohl der gesamte Datenbestand, da der Gesamt-Index auf **einer** CD Platz findet. Zur Wiedergabe der Treffer muß dann natürlich die jeweils passende CD eingelegt werden. Stehen mehrere Laufwerke zur Verfügung, kann dieser lästige Schritt entfallen, da heutige CD-Software die Ansteuerung von bis zu vier CDs automatisch erledigt. Der Nutzer bemerkt dann von dieser Systemverwaltung nichts, aus seiner Sicht hat er es nur mit einer einzigen homogenen Datenbank zu tun.
PostScript: Die beiden Bildschirm-Ausdrucke hat Elmar Schalück produziert.

```
fin S, the usual class of univalent functions, then I(sub p) (r,f) <= I(sub p) (r,k).
In this paper, the author shows that the case p = 1 of the inequality follows easily
from Robertson's conjecture which is now known to be true because of the proof by de
Branges of the Bieberbach conjecture. Let (f(z) / z) sup(p / 2) = Sum a(sub n) (p)z(sup n)
and (k(z) / z) sup(p / 2) = (1 - z) sup(-p) = Sum d(sub n) (p)z(sup n). Then it is shown,
almost as easily, that for p >= 2 and any f in S, |a(sub n) (p)| <= d(sub n) (p) and hence
I(sub p) (r,f) <= I(sub p) (r,k). The paper concludes with a few other observations
such as Sum over (k=0,n)|a(sub k) (p)| <= Sum over (k=0,n) d(sub k) (p), n >= 0, 1 < p < 2,
Sum over (k=0,infinity)|a(sub k) (p)| r(sup k) <= (1 - r) sup(-p), 0 <= r < 1, 0 < p < 1.
```

Abb. 3: Text in MATH