Slavka SLAVOVA, Grozio STANILOV, Dobrich

Maple Visualisierung des Mathematischen Unterrichts

Computer-Algebra-Systeme (CAS) sind schon ein starkes Mittel für ein besseres Verständnis der Mathematik. Es gibt viele Argumente für ihre Benutzung in der mathematischen Ausbildung: mehr realistische Anwendungen, Analyse der reellen Angaben, ein besseres Verständnis von Funktionen, mehr Anziehung und Endeckung von Studenten, mehrfache Lösungsmethoden des Problems. Die geben notwendige Mittel in den Studentenhänden zur Entdeckung die grundlegende Konzeptionen, Regel und Muster für sich selbst, für Untersuchung schon bekannte Probleme, für die Anwendungen in der reellen Welt mit Computer Methoden, die eine große studentische Interesse motivieren. Das moderne Computer Software wie Maple kann und muss benutz werden, für die Entwicklung mittels komplizierte aber hoch qualitative Visualizierung und Animationen der abstrakten mathematischen Ideen. In dieser Arbeit beschreiben wir wie könnte man grundlegende Tatsachen von Maple zu den Studenten einbringen sollten, und demonstrieren wir die Benutzung von Maple für mathematisches Unterricht. Die Arbeit enthält eine Menge von Animationen und Visualisierungen. Wir diskutieren auch die Benutzung von Animationen als ein neuer Weg der mathematischen Ausbildung. Unsere Erfahrung lautet: die Studenten ziehen vor Computer Algebra zu benutzen, sie genießen ihre Arbeit in den Computer Sälen insbesondere bei Visualizierung und Animationen.

Unter dem Begriff der „Computeralgebra” versteht man heute den Grenzbereich zwischen Algebra und Informatik, der sich mit dem Entwurf, Analyse, Implementierung und Anwendung algebraischer Algorithmen befasst. Die Entwicklung von CAS geht bis in die vierziger Jahre des 20.Jahrhunderts zurück, als man kleinere Systeme zur Lösung spezieller Probleme einsetzte. Der eigentliche Aufschwung setzte aber erst ein, als man in der theoretischen Elementarteilchenphysik dazu überging, Formeln von Computern symbolisch bearbeiten zu lassen. Das Wort ’Algebra’ in CAS meint somit also die algorithmische Behandlung symbolischer algebraischer Ausdrücke, die zur Lösung von Gleichungen - oder allgemeiner von algebraischen Aufgabenstellungen notwendig sind. Der resultierende Algorithmus ist allerdings nach wie vor so komplex, dass bei einer Rechnung auf dem Papier wohl meist nach stundenlangem Rechnungen bei einem dann wahrscheinlich falschen Ergebnis enden würde. Computeralgebrasysteme sind die „Fast-Alleskönner" unter den Mathematikprogrammen. Sie zeichnen sich dadurch aus, daß sie Terme symbolisch bearbeiten können und die Bruchrechnung beherrschen. Sie können algebraische Terme faktorizieren und zusammenfassen sowie Gleichungen und Gleichungssysteme lösen. Dabei kann man einen Lösungsalgorithmus in seinen einzelnen Schritten darstellen, ohne auf einen einzigen Lösungsweg festgelegt zu sein wie bei vielen Lern- und Übeprogrammen zur Algebra. CAS können aber auch Ableitungs- und Stammfunktionen ermitteln. Sogar die Bestimmung von Taylorreihen oder die Lösung von einigen Differentialgleichungen ist mit solchen Systemen kein Problem. Auch die (zwei- und dreidimensionalen) graphischen Fähigkeiten der CAS sind beachtlich.

Im Jahre 1980 begannen an der kanadischen Universität Waterloo die Professoren Keith Geddes und Gaston Gonnet die Entwicklungs-arbeiten an Maple, um ein mathematisches Softwaresystem zu konstruieren, das sich sowohl in der Forschung als auch zur Ausbildung einsetzen läßt. Bereits nach nur drei Wochen wurde die erste funktionierende Version im

Dezember 1980 fertiggestellt. Im Laufe der nächsten Jahre wurde das System infolge mehrerer Förderprogramme weiterentwickelt und schon relativ früh an amerikanischen und europäischen Universitäten auf so verschiedenen Gebieten wie Mathematik, Informatik, Physik, Wirtschaft und Ingenieurwesen eingesetzt. Durch Ausprobieren und Verwerfen von Ideen wurden immer neuere und bessere Lösungsverfahren in Maple verwirklicht. Dieser Prozeß hält bis heute an.

Neben Maple gibt es weitere Computeralgebrasysteme, die bekanntesten sind Mathematica, Mathcad Plus, MuPad und Derive. Maple ist sehr bedienungsfreundlich, es ist ein mächtiges wissenschaftlich-mathematisches Computer-programm; seine Funktionen umfassen: symbolische Berechnungen; numerische Verfahren; zwei- und dreidimensionale Graphikfähigkeiten; mathematische Funktionen für Algebra, Analysis, diskrete Mathematik, Finanzmathematik, Lineare Algebra, Physik und Statistik; eine umfangreiche Programmiersprache. Darüber hinaus kann das Arbeiten mit Maple sehr viel Spaß bereiten. Maple beinhaltet eine komplette Umgebung zur Lösung mathematischer Problemstellungen mit Hilfe symbolischer und numerischer Verfahren sowie umfangreiche Hilfsmittel zur graphischen Darstellung und zur Animation der Ergebnisse.

1. Animationen. Unter eine Animation versteht man eine graphische Darstellung, die sich in Abhängigkeit von einer zusätzlichen Variablen, der animationsvariablen, ändert. Für diese Variable ist ein eigener Bereich festzulegen, in dem sie variiert. Einige Beispiele für zwei- und dreidimensionale Animationen sind in folgen angegeben. Mit dem Kommando *with(plots)* wird das *plots* Paket von Maple bereitgestellt, in dem die Anweisungen *animate* und *animate3d* enthalten sind.

1.1. Das Kommando **animate(f(x,t),x=a..b,t=c..d,Optionen)** definiert eine Animations-folge von Schaubildern einer Funktionenschar. Hier ist t der Scharparameter; der frames-Wert gibt die Anzahl der zu zeichnenden Schaubilder an. Hier sind einige Komandos:

```
animate(cos(x*t),x=-5..5,t=1..6,numpoints=200,frames=200);
animate([cos(x*t),x,x=-5..5],t=1..6,coords=polar,
numpoints=100,frames=100);
animate(cos(t*x)*exp(-x),x=-3..3,t=1..20,frames=20,
thickness=2);
animate([u*t,t,t=1..8*Pi],u=1..4,coords=polar,frames=100,
numpoints=200);
s:=t->100/(100+(t-Pi/2)^8): r:=t->s(t)*(2-sin(7*t)-
cos(30*t)/2):
animate([u*r(t)/2,t,t=-Pi/2..3/2*Pi],u=1..2,numpoints=200,
coords=polar,axes=none,color=green);
```

1.2. Das Kommando **animate3d(f(x,y,t),x=a..b,y=c..d,t=p..q,Optionen)** in 3D.

```
animate3d(cos(t*x)*sin(t*y),x=-Pi..Pi,y=-Pi..Pi,t=1..3,
color=cos(x*y));
animate3d(x*cos(t*u),x=1..3,t=1..4,u=2..4,coords=spherical);
animate3d(sin(x)*cos(t*u),x=1..3,t=1..4,u=1/4..7/2,
```

```
coords=cylindrical);
animate3d([x,y,(1.3)^x*sin(u*y)],x=1..3,y=1..4,u=1..2,
coords=spherical);
```

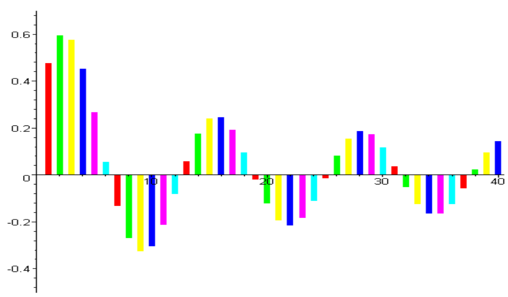
1.3. Das Kommando **animatecurve(f(x),x=a..b,Optionen)** in2D.

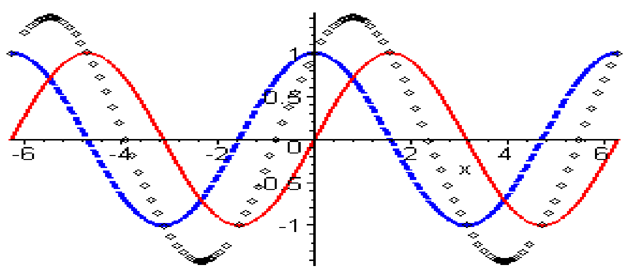
```
animatecurve(sin(x),x=-2*Pi..2*Pi,frames=50,color=blue,
thickness=2);
animatecurve({x^2-x,cos(x)},x=-Pi..Pi,frames=100,
color=green,thickness=2);
```

2. Graphische Darstellung. Hier steht der Befehl *plot* in 2D und *plot3D* in 3D zu Verfügung. Mit *orientation* gibt man die gewünschten Blickwinkel an, *style* definiert die Art der Darstellung, *shading* bestimmt die Einfärbung des Graphen und *grid* dessen Auflösung in x- und y Richtung.

2.1. Reele Folgen.

```
a:=n->sin(n/2)/sqrt(n); Folge:=seq(evalf(a(n),3),n=1..8);
plot([seq([n,t,t=0..a(n)],n=1..40)],thickness=6,
view=[0..40.5,-0.5..0.7]);
```



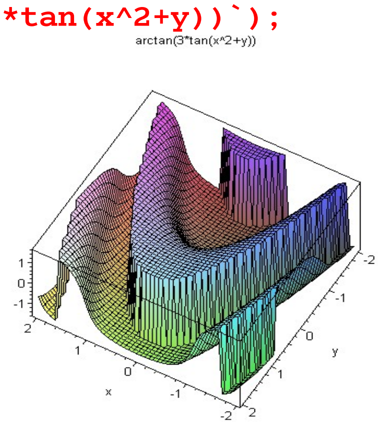


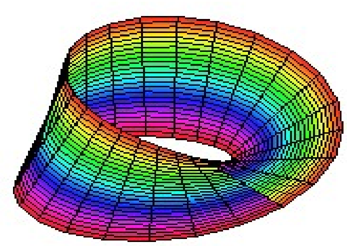
2.2. Funktionen in zwei Dimensionen.

```
plot(sin(x)/x,x=0..infinity);
plot([sin(4*x),x,x=0..2*Pi],coords=polar);
plot(sin(x)*sin(10*x)*sin(100*x),x=-6*Pi..6*Pi,color=navy);
plot(sin(x)*sin(10*x)*sin(100*x),x=-
0.5..0.5,thickness=2,color=navy);
plot([sin(x),cos(x),sin(x)+cos(x)],x=-
2*Pi..2*Pi,style=[line,line,point],linestyle=[1,3],color=[red,
blue,black],thickness=[2,3]);
plot([sin,cos,tan],-2*Pi..2*Pi,-
5..5,discont=true,color=[blue,green,coral],thickness=2);
plot(cos(cos(cos(cos(x)))),x=-Pi..Pi,color=green,
thickness=6);
implicitplot((x^2)/9+(y^2)/4=1,x=-3..3,y=-2..2,
thickness=2,color=brown);
```

2.3. Funktionen in drei Dimensionen.

```
f:=(x,y)->exp(x)*cos(y);  plot3d(f(x,y),x=-4..4,y=4..10);
plot3d(f(x,y),x=0..10,y=4..10,orientation=[64,87],
shading=none,color=sin(x)*cos(y));
g:=(x,y)->x*y/(x^2+y^2);
plot3d(g(x,y),x=-1..1,y=-1..1,axes=boxed,grid=[50,50]);
plot3d(g(x,y),x=-1..1,y=-1..1,axes=boxed,grid=[50,50],
orientation=[-50,20]);
plot3d([u*cos(v),-v*sin(u),u*v/4],u=-2*Pi..2*Pi,v=-
2*Pi..2*Pi,grid=[80,40],shading=zhue,orientation=[-45,45]);
plot3d([0.3*cos(u),(1+0.3*sin(u))*cos(v),(1+0.3*sin(u))*
sin(v)],u=0..2*Pi,v=0..2*Pi,scaling=constrained,axes=normal,nu
mpoints=2500,title=`torus`);
plot3d(arctan(3*tan(x^2+y)),x=-2..2,y=-
2..2,grid=[50,50],axes=boxed,orientation=[120,20],title=`arcta
n(3*tan(x^2+y))`);
```





```
mb:=[(2+s*cos(theta/2))*cos(theta),(2+s*cos(theta/2))*sin
(theta),s*sin(theta/2)]:
plot3d(mb,s=-1..1,theta=0..2*Pi,orientation=[-142,-145],
color=s);
```

2.4.Kurven im Raum. Kurven werden im Raum mit *spacecurve* aus dem Paket 'plots' gezeichnet.

```
spacecurve([t*sin(t),t*sin(t),t],t=0..6*Pi,
orientation=[65,81],axes=normal,shading=z);
f:=(x,t)->t*x*exp(-x^2);
spacecurve({seq([x,t,f(x,t),x=-5..5],t=1..6)},orientation=[-
102,82],shading=z,axes=frame);
```

Literatur

Heck, A.: Introduction to Maple, Springer Verlag New York 2003
Walz, A.: Maple 7, Oldenbourg Wissenschaftsverlag München 2002