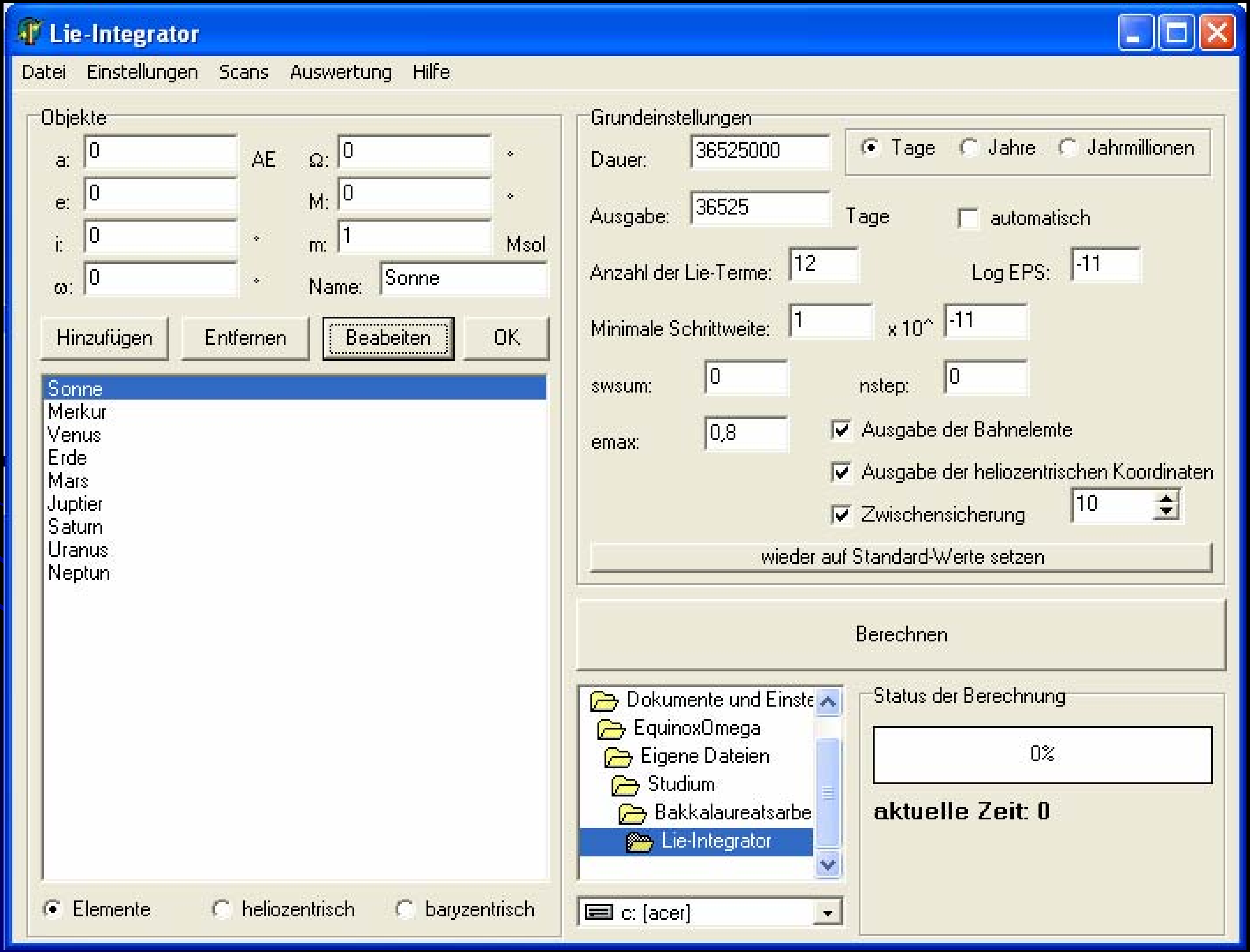
Der Lie-Integrator



Sophus Lie

- geb.: 17. 12. **1842** in Nordfjordeid
- gest.: 18. 2. **1899** in Oslo

- Er studierte verschiedene Naturwissenschaften an der Universität von Oslo bis er zur Mathematik fand.
- 1869: erste Publikation
- **1871-1872**: Entwicklung der **Lie-Integration** zur Lösung von Differentialgleichungen
- Arbeiten über Geometrie, Transformationsgruppen und Kommutator-Algebra
- Lehrte in Leipzig und Oslo

- Er neigte zu Depressionen und zerstritt sich mit seinen Kollegen.

Lie-Integration

- Ist ein Verfahren zum numerischen Lösen von Differentialgleichungen

$$\cancel{\int f(z)dz}$$

$$z = e^{t*D}\xi$$

Der Lie-Operator

ist ein linearer Differentialoperator

$$D = \theta_1 \frac{\partial}{\partial z_1} + \theta_2 \frac{\partial}{\partial z_2} + \ldots + \theta_n \frac{\partial}{\partial z_n}$$

Wobei: $D^2(f) = D(D(f))$

Lie-Reihen

$$L(z,t) = e^{t*D} f(z)$$

$$L(z,t) = \sum_{i=0}^{\infty} \frac{t^i}{i!} D^i f(z)$$

$$L(z,t) = f(z) + t*D*f(z) + \frac{t^2}{2!} D^2 * f(z) + \ldots$$

Methode zum Lösen von DGL

- Anwendbar auf Systeme von Differentialgleichungen erster Ordnung

$$\frac{dz_i}{dt} = \theta_i(z_1, z_2, \ldots, z_n)$$

- Mit gegebenen Anfangsbedingungen numerisch beliebig genau lösbar

$$z_i(t=0) = \xi_i$$

- Der Lie-Operator ist in folgender Form zu verwenden:

$$D = \sum_{i=1}^{n} \theta(\xi_1, \xi_2, \ldots \xi_n)_i \frac{\partial}{\partial \xi_i}$$

- Somit erhält man die Lösungen der DGLs:

$$z_i = e^{t*D} \xi_i$$

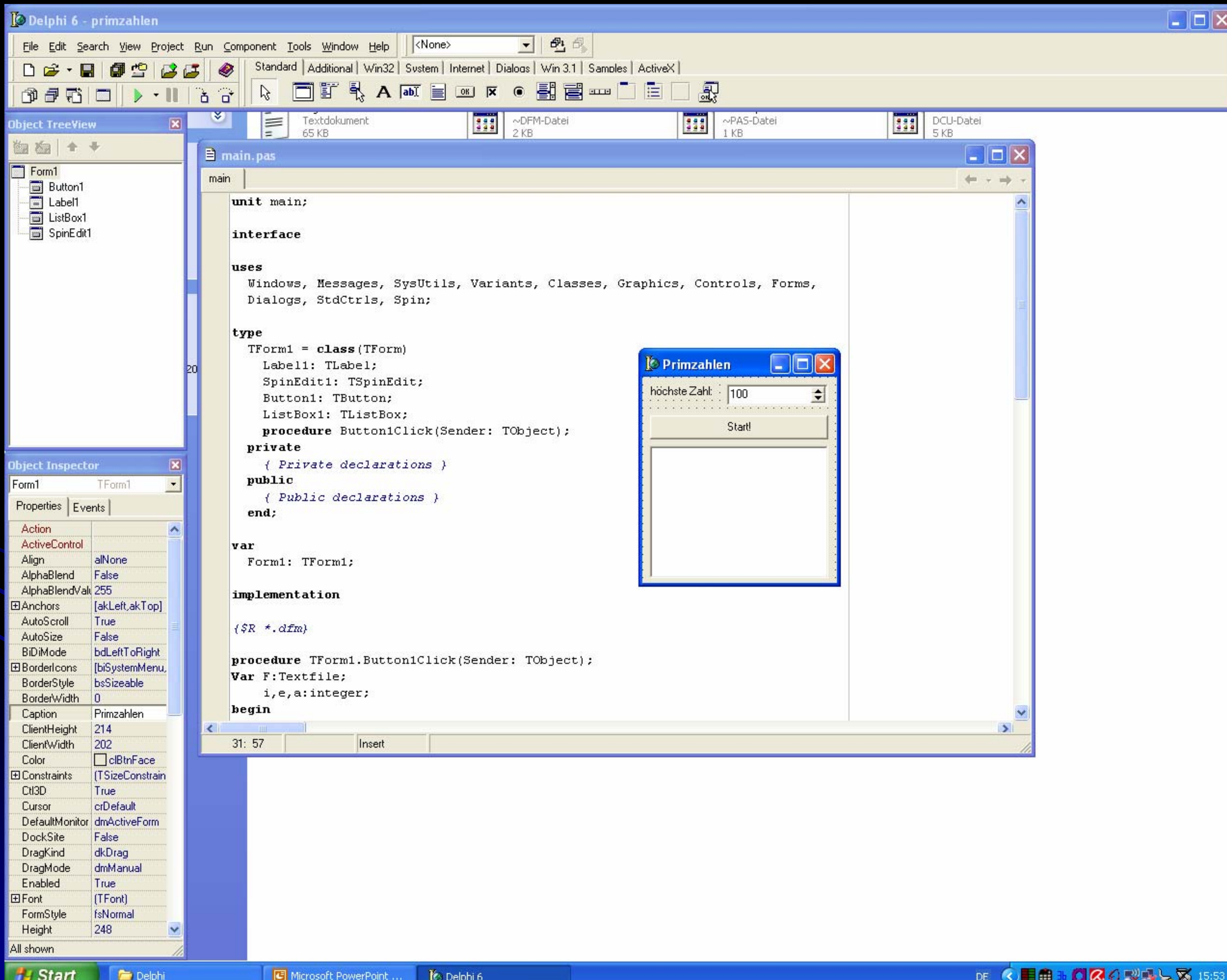
$$z_i = \xi_i + t*D(\xi_i) + \frac{t^2}{2!} D^2(\xi_i) + \frac{t^3}{3!} D^3(\xi_i) + \ldots$$

Für weitere Details ➔ Astronomische Rechenmethoden und Informatik

Vorteile

- Schrittweite änderbar
- Hohe Genauigkeit
- Relativ kleine Rundungsfehler
- Verhältnismäßig schnelle Berechnungen

Delphi 6


Delphi 6 - primzahlen
File Edit Search View Project Run Component Tools Window Help
Standard Additional Win32 System Internet Dialogs Win 3.1 Samples ActiveX
Object TreeView
Form1
Button1
Label1
ListBox1
SpinEdit1
Textdokument 65 KB
~DFM-Datei 2 KB
~PAS-Datei 1 KB
DCU-Datei 5 KB
main.pas
main
unit main;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Spin;
type
TForm1 = class(TForm)
Label1: TLabel;
SpinEdit1: TSpinEdit;
Button1: TButton;
ListBox1: TListBox;
procedure Button1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
Var F:Textfile;
i,e,a:integer;
begin
31: 57
Insert
Primzahlen
höchste Zahl:
100
Start!
Object Inspector
Form1 TForm1
Properties Events
Action
ActiveControl
Align alNone
AlphaBlend False
AlphaBlendVal 255
Anchors [akLeft,akTop]
AutoScroll True
AutoSize False
BiDiMode bdLeftToRight
BorderIcons [biSystemMenu,
BorderStyle bsSizeable
BorderWidth 0
Caption Primzahlen
ClientHeight 214
ClientWidth 202
Color clBtnFace
Constraints (TSizeConstrain
Ctl3D True
Cursor crDefault
DefaultMonitor dmActiveForm
DockSite False
DragKind dkDrag
DragMode dmManual
Enabled True
Font (TFont)
FormStyle fsNormal
Height 248
All shown
Start
Delphi
Microsoft PowerPoint ...
Delphi 6
DE
15:53

Besonderheiten

- Delphi = Visual Pascal
- Es gibt eine 80-Bit Gleitkommavariable **extended** : $3,6*10^{-4951}$ bis $1,1*10^{4932}$
- dynamische Arrays
- graphische Oberfläche
- geht direkt in Assembler
- Kombination von objektorienierter und strukturierter Programmiersprache
- viele vorgefertigte Komponenten

Ein einfaches Programm

In Delphi

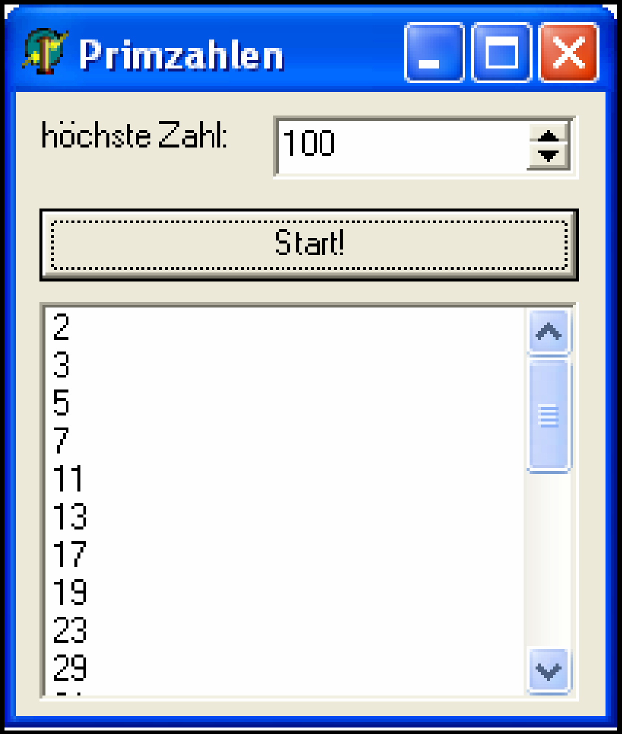
```
procedure TForm1.Button1Click(Sender: TObject);
Var F:Textfile;
   i,e,a:integer;
begin
assignfile(F,'ergebnis.txt');
rewrite(F);
for i:=2 to spinedit1.Value do
begin
a:=0;
for e:=2 to i-1 do
begin
if (i mod e)=0
then
inc(a);
end;
if a=0
then
begin
writeln(F,inttostr(i));
listbox1.Items.Add(inttostr(i));
end;
end;
closefile(F);
end;
```

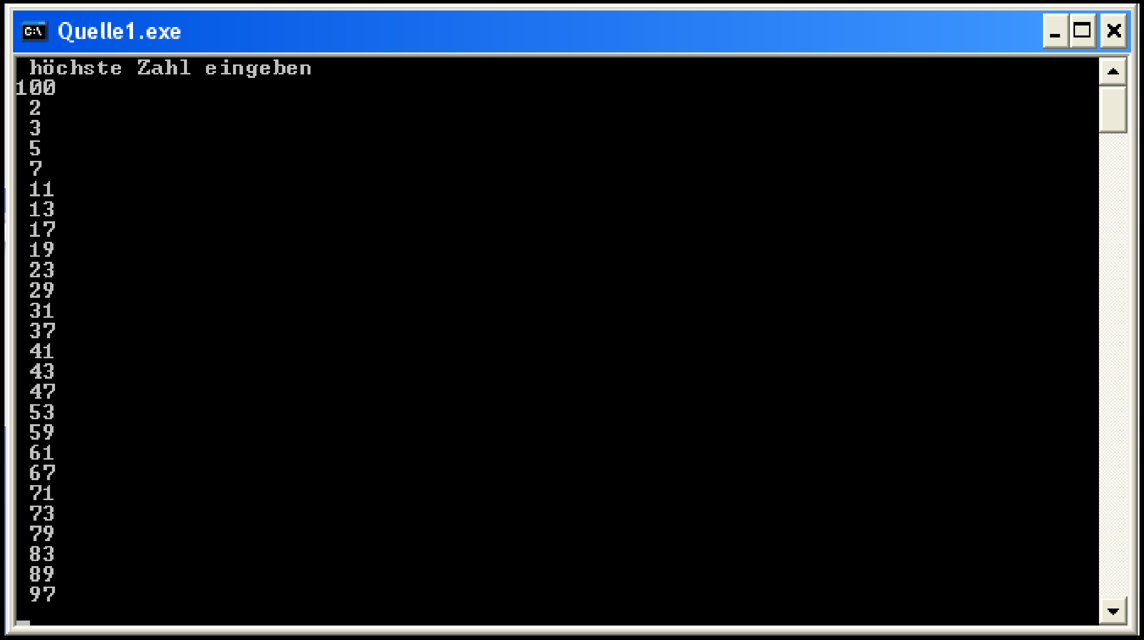
In Fortran

```
 integer a,i,e,max

    print*,'höchste Zahl eingeben'
    read*,max

    open(unit=11,file='ergebnis.txt',status='unknown')
    do 100 i=2,max
    a=0
    do 200 e=2,i-1
    if (MOD(i,e)==0) then
    a=a+1
    end if
200   continue
    if (a==0) then
    write(11,*) i
    print*, i
    end if
100   continue
    close(11)
    stop
    end
```





Delphi 6

Rechenzeit für
100000 Primzahlen:

2min 07s

Fortran 95

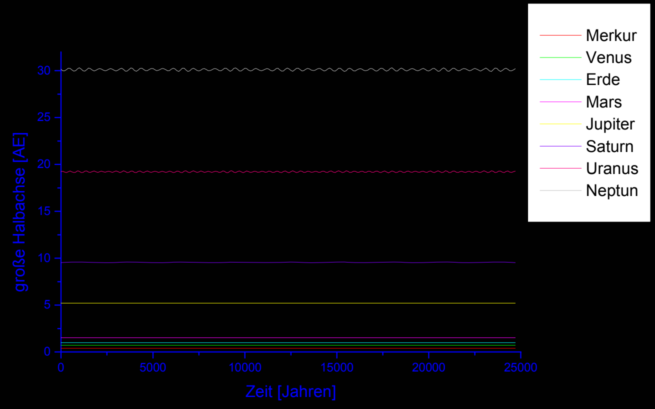
Rechenzeit für
100000 Primzahlen:

2min 09s

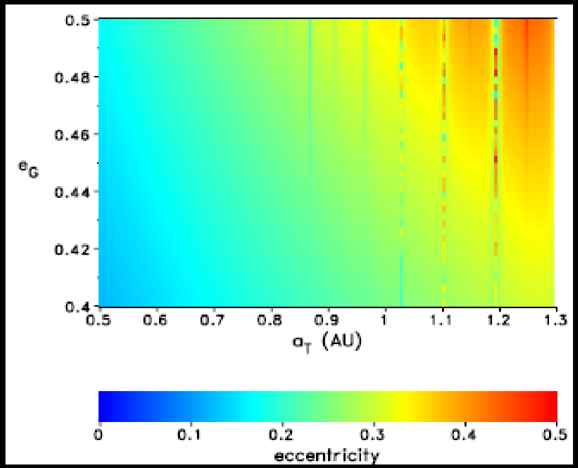
Trotz graphischer Benutzeroberfläche ist Delphi etwa gleich schnell wie Fortran

Verwendung des Programms

- Numerische Langzeitintegration von Planeten- und Asteroidenbahnen



- Stabilitätsanalyse von Sternensystemen

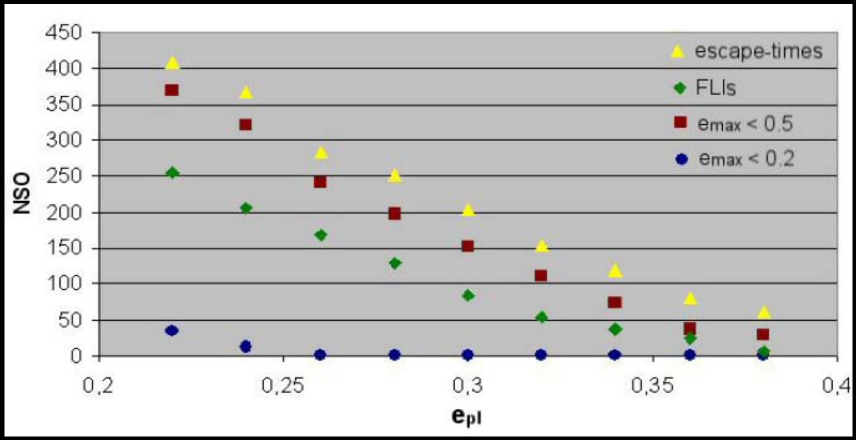


Änderungen

- Programm interaktiver und flexibler gestalten
- Quellcode etwas optimieren und entrümpeln
- Auswertung ins Programm integrieren
- Graphische Benutzeroberfläche

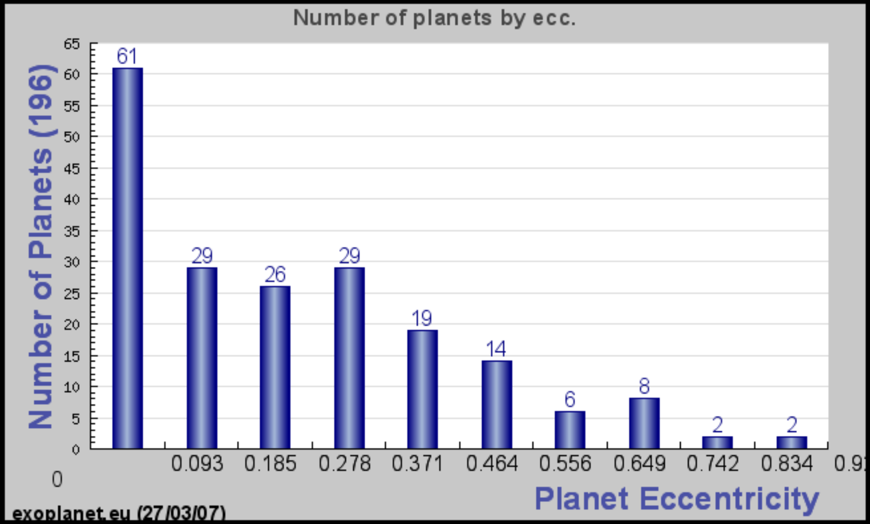
Instabilitätskriterium ändern

derzeit liegt das Instabilitätskriterium bei einer Exzentrizität von 0,5



erscheint ausreichend zu sein

doch es sind nun auch schon Exo-Planeten mir höheren Exzentrizitäten bekannt



Close-Encounters und Einfänge

- Wann ist das Gravitationsfeld eines Planeten stärker als das des Sterns?

Radius der Hill-Sphäre: $r \approx a * (1-e) \sqrt[3]{\frac{m}{3 * M}}$

Unterhalb dieser Entfernung vom Planeten kann es zu Einfängen kommen.

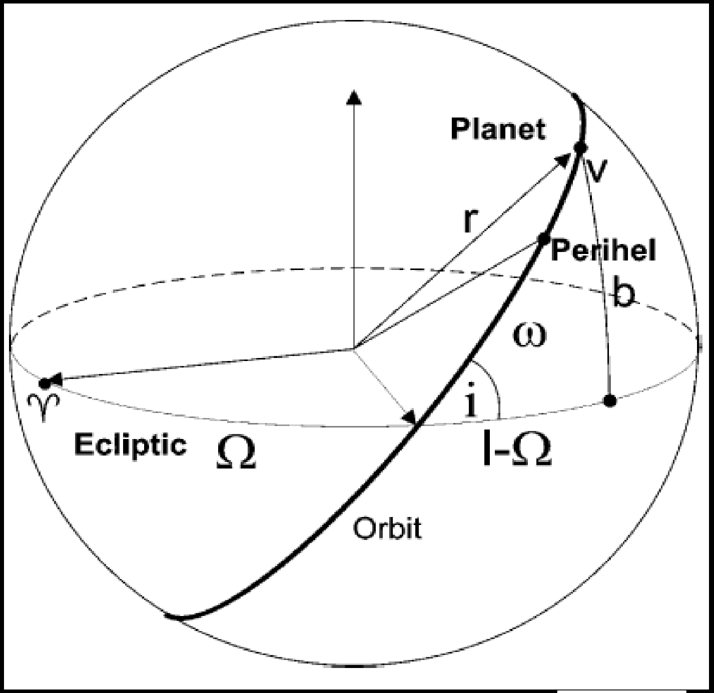
Störung vom Planeten sehr stark ➔ Schrittweite im Lie-Integrator sehr klein ➔ verlangsamt Berechnung ➔ gestörtes Objekt aus Berechnung nehmen

Objekt wird nicht gelöscht ➔ dafür weit weg vom System

Bereich scannen

Schaffung der Möglichkeit im Programm automatisch einen Bereich in den Bahnelementen in vorgegebenen Intervallen abzuscannen.

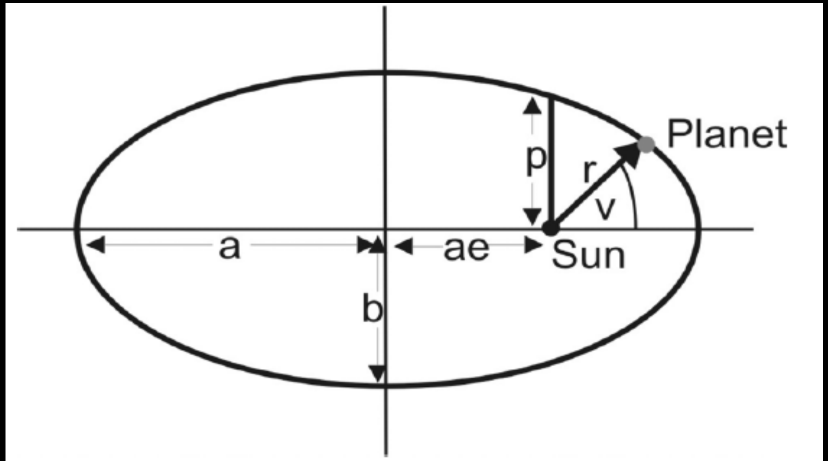
Finden von möglichen stabilen Bereiche für masselose oder auch massebehaftete Objekte.



$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

Bahnelemente:

a: große Halbachse
e: Exzentrizität
i: Inklination
Ω: Länge des aufsteigenden Knotens
ω: Argument des Perihels
M: mittlere Anomalie

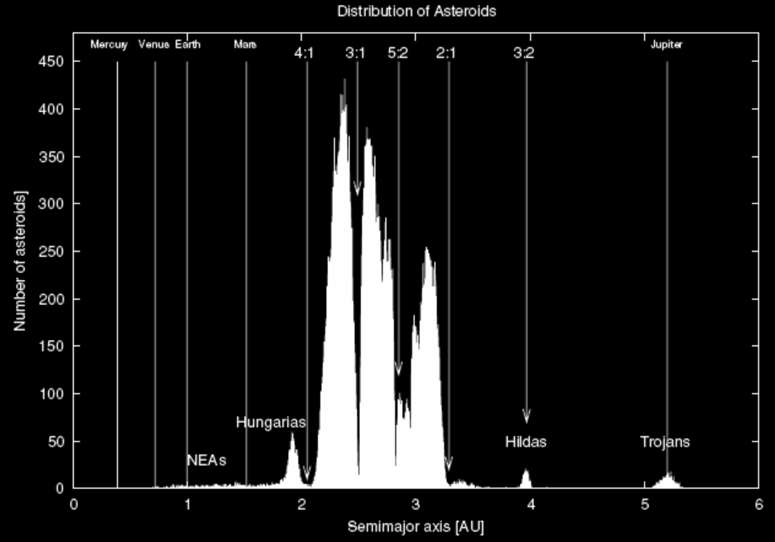


$$E - e * \sin(E) = M(t_0) + n(t - t_0)$$

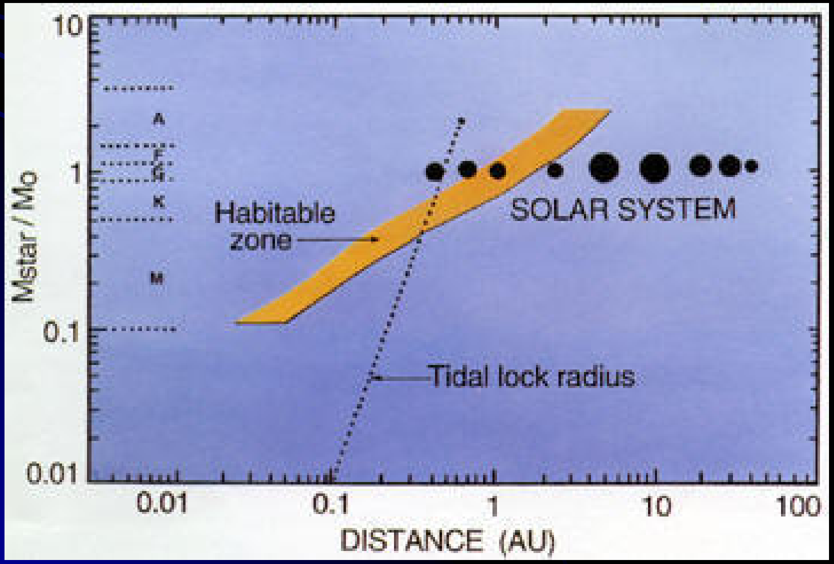
$$\tan(\frac{v}{2}) = \sqrt{\frac{1+e}{1-e}} * \tan(\frac{E}{2})$$

v: wahre Anomalie

- Stabile und Instabile Bereiche in Asteroidengürteln untersuchen



- Stabile Terrestrische Planeten in der Habitablen Zone um andere Sterne finden



Das Programm soll den zu untersuchenden Bereich selbstständig ermitteln können.

$$d[AU] \simeq \sqrt{L_* / L_\odot}$$

graphische Auswertung

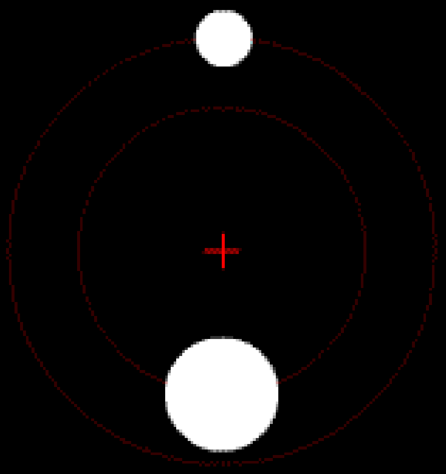
- Diagramme mit dem Verlauf von Bahnelementen über die Zeit
- färbige Stabilitätsmaps
- 3D-Darstellung der Planetenbewegung
- wahlweise: Darstellung auch schon während der Berechnung

Programmentwicklung

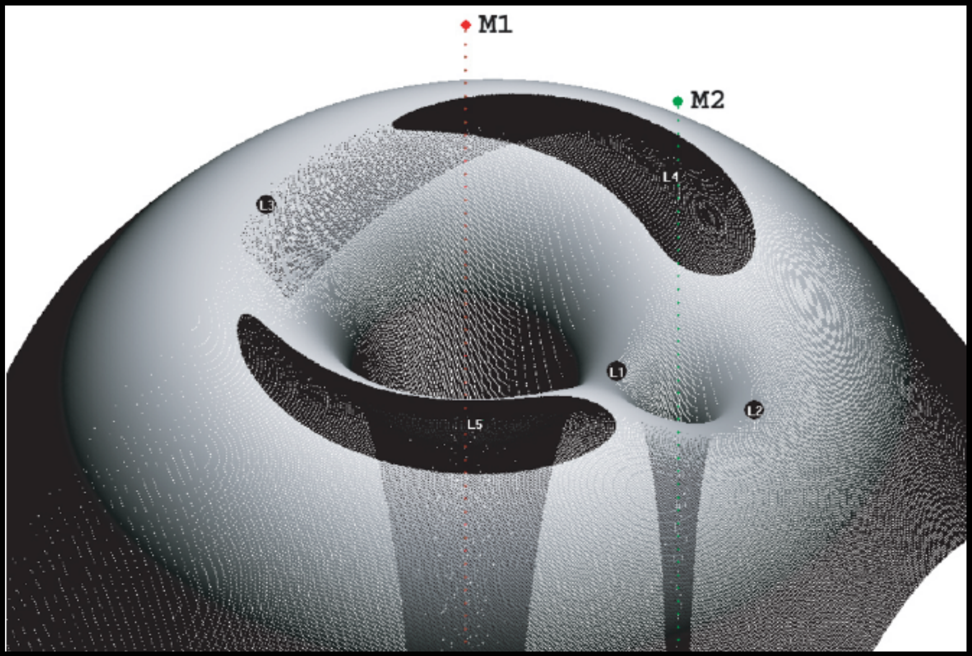
- Grundsätzlicher Aufbau
- Übersetzen von Fortran in Delphi
- Bugfix
- Hinzufügen von Extrafeatures
- Programmtest
- Optimierung von Details

Programmtests

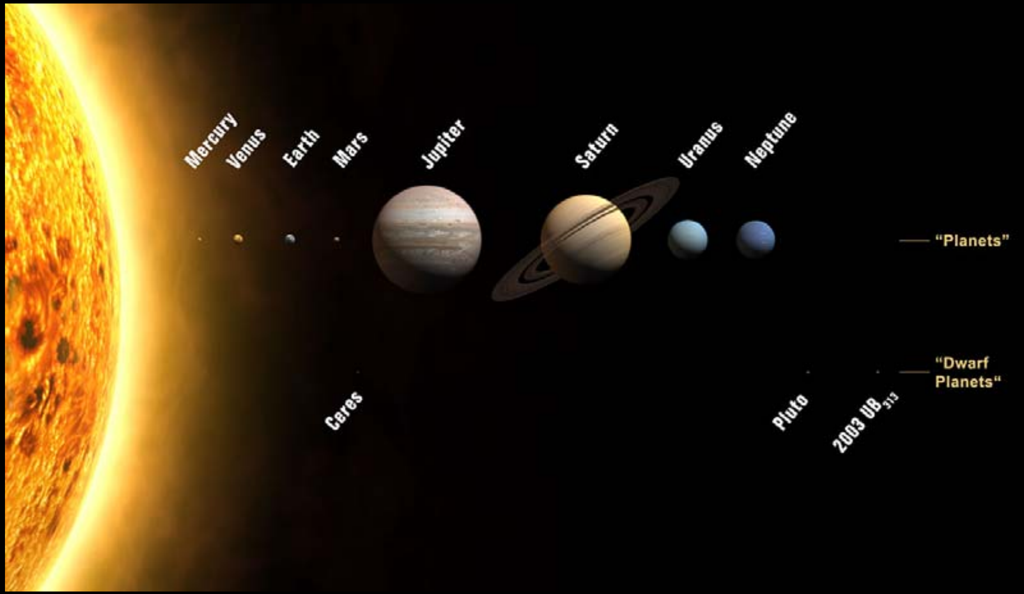
- Simulation des 2-Körperproblems



- Simulation des eingeschränkten 3-Körperproblems

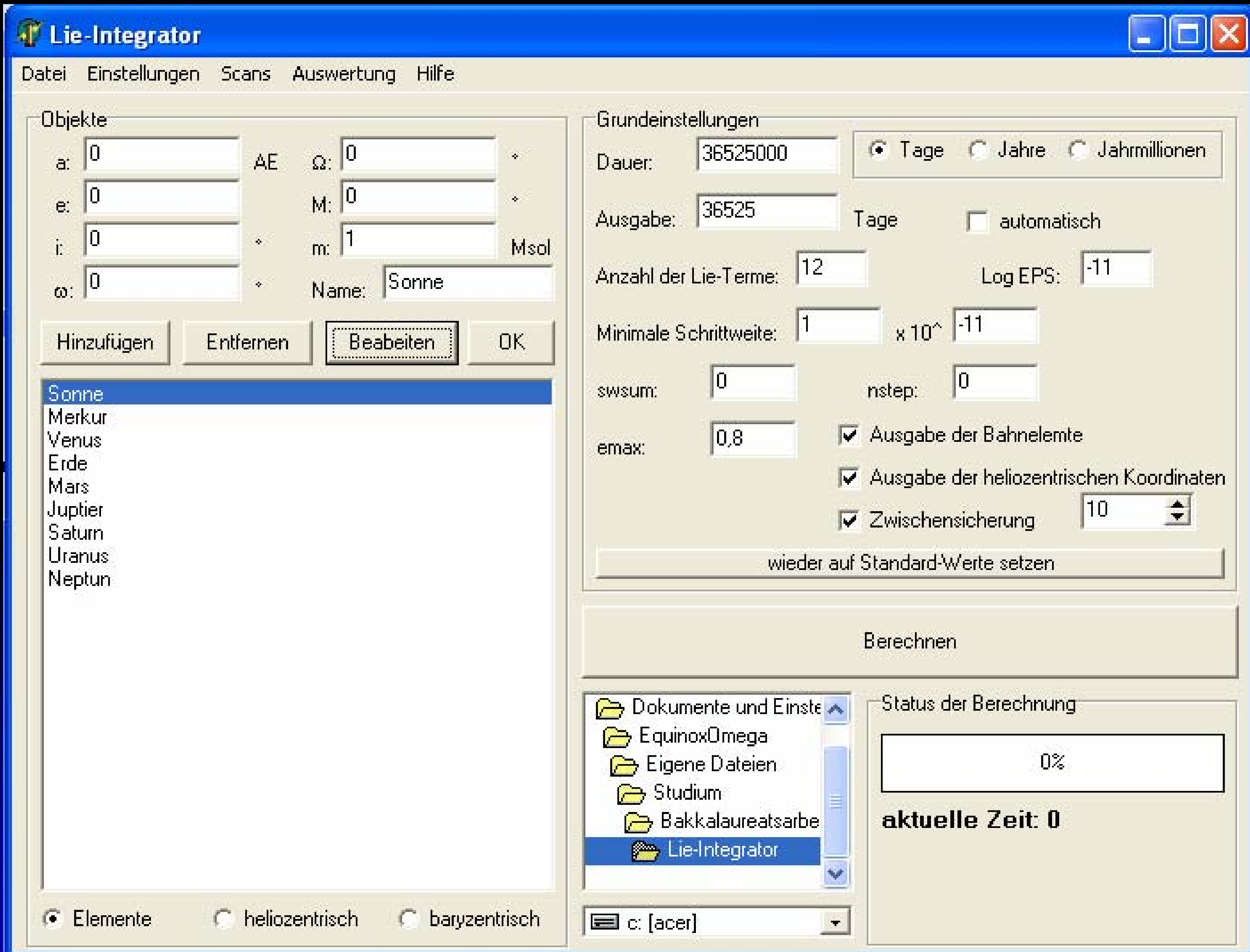


- Simulation unseres Sonnensystems



- Vergleich der Rechengeschwindigkeit zwischen altem und neuem Programm

- Ausführliche Tests der neuen Programmelemente


Lie-Integrator
Datei
Einstellungen
Scans
Auswertung
Hilfe
Objekte
a: 0 AE
e: 0
i: 0 °
ω: 0 °
Ω: 0 °
M: 0 °
m: 1 Msol
Name: Sonne
Hinzufügen
Entfernen
Beabeiten
OK
Sonne
Merkur
Venus
Erde
Mars
Juptier
Saturn
Uranus
Neptun
Elemente
heliozentrisch
baryzentrisch
Grundeinstellungen
Dauer: 36525000
Tage
Jahre
Jahrmillionen
Ausgabe: 36525 Tage
automatisch
Anzahl der Lie-Terme: 12
Log EPS: -11
Minimale Schrittweite: 1 x 10^ -11
swsum: 0
nstep: 0
emax: 0,8
Ausgabe der Bahnelemte
Ausgabe der heliozentrischen Koordinaten
Zwischensicherung 10
wieder auf Standard-Werte setzen
Berechnen
Dokumente und Einste
EquinoxOmega
Eigene Dateien
Studium
Bakkalaureatsarbe
Lie-Integrator
c: [acer]
Status der Berechnung
0%
aktuelle Zeit: 0