



Universität Potsdam

Udo Schwarz

## Einführung in die nichtlineare Dynamik

NLD Preprints ; 8



Januar 1994

Preprint NLD-008

**Interdisziplinäres Zentrum  
für  
Nichtlineare Dynamik**

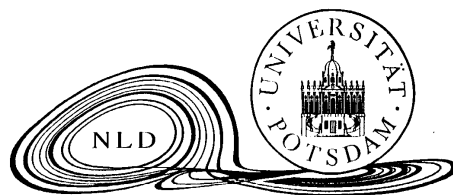
**Einführung  
in die Nichtlineare Dynamik**

Udo Schwarz

unter Mitwirkung von

Thomas Förster  
Andreas Jacob  
André Jannasch  
David Kappel  
Jörgen Kosche  
Andreas Schütze  
Sandro Wricke

[Skriptum]



**Universität Potsdam**

Potsdam & Kleinmachnow, im Januar 1994

Bisher erschienene NLD Preprints<sup>1</sup>:

1. F. Feudel and N. Seehafer: On the Bifurcation Phenomena in Truncations of the 2D Navier-Stokes Equations, March 1994
2. A.S. Pikovsky and U. Feudel: Characterizing Strange Nonchaotic Attractors, April 1994
3. J. Kurths, A. Pikovsky and C. Scheffczyk: Roughening Interfaces in Deterministic Dynamics, July 1994
4. A. Witt, J. Kurths, F. Krause and K. Fischer: On the Validity of a Model for the Reversals of the Earth's Magnetic Field, August 1994
5. J. Kurths, A. Voß, A. Witt, P. Saporin, H.J. Kleiner and N. Wessel: Quantitative Analysis of Heart Rate Variability, August 1994
6. Helmholtz – Kolloquium, Institut für Mathematik, 8. September 1994, Vortragsauszüge, September 1994
7. C.-V. Meister: Stochastic Forces on Electrons in the Solar Flare Plasma, November 1994
8. U. Schwarz, T. Förster, A. Jacob, A. Jannasch, D. Kappel, J. Kosche, A. Schütze und S. Wricke: Einführung in die Nichtlineare Dynamik. Skriptum, March 1994
9. F. Feudel, N. Seehafer and O. Schmidtman: Bifurcation Phenomena of the Magnetofluid Equations, January 1995
10. C.-V. Meister: Stochastic Forces in Space Plasmas with Ion-Acoustic and Lower-Hybrid-Drift Turbulence, January 1995
11. V.A. Liperovsky, C.-V. Meister, K. Schlegel and C. Haldoupis: Currents and Turbulence in and near Mid-Latitude Sporadic E-Layers Caused by Strong Acoustic Impulses, February 1995
12. M. Rosenblum and J. Kurths: A Model of Neural Control of Heart Rate, March 1995
13. K.-U. Thiessenhusen, L.W. Esposito, J. Kurths and F. Spahn: Detection of Hidden Resonances in Saturn's B-Ring, March 1995
14. F. Spahn, J.-M. Hertzsch and N.V. Brilliantov: The Role of Particle Collisions for the Dynamics in Planetary Rings, March 1995
15. A.S. Pikovsky, M.A. Zaks, U. Feudel and J. Kurths: Singular Continuous Spectra in Dissipative Dynamics, March 1995
16. A. Neiman, U. Feudel and J. Kurths: The Cumulant Approach for Investigating the Noise Influence on Mode-Locking Bifurcations, March 1995
17. O. Schmidtman: Modelling of the Interaction of Lower and Higher Modes in Two-Dimensional MHD-Equations, March 1995
18. F. Feudel, N. Seehafer and O. Schmidtman: Fluid Helicity and Dynamo Bifurcations, April 1995
19. P. Morin: Visualization in the Geosciences. Course Notes, July 1995
20. V.A. Liperovsky and C.-V. Meister: Sporadic E-Layers as Current Generator. Two-Dimensional Model, August 1995

<sup>1</sup>Manuskripte für weitere Preprints und Preprint-Anforderungen nimmt Udo Schwarz entgegen: Zentrum für Nichtlineare Dynamik, Universität Potsdam, Am Neuen Palais, Postfach 601553, D-14415 Potsdam, Tel. (49-331)-977-1658, Fax (49-331)-977-1142, E-Mail: USchwarz@agnld.Uni-Potsdam.de, WWW-Page: <http://www.agnld.Uni-Potsdam.de/>

21. V.A. Liperovsky, C.-V. Meister, K.V. Popov, S.A. Senchenkov, M.A. Oleynik and E.V. Liperovskaya: Quasi-Three Dimensional Model of Current Generation in the Ionosphere Caused by Neutral Wind Action on E<sub>s</sub>-Clouds, August 1995
22. V. Dicken and P. Maaß: Wavelet-Galerkin-methods for ill-posed problems, August 1995
23. F. Feudel and N. Seehafer: Bifurcations and Pattern Formation in a 2D Navier-Stokes Fluid, August 1995
24. I.I. Blekhman, P.S. Landa and M.G. Rosenblum: Synchronization and Chaotization in Interacting Dynamical Systems, August 1995
25. N. Seehafer: Nature of the  $\alpha$  Effect in Magnetohydrodynamics, October 1995
26. C.-V. Meister and I. Kubyshkin: Recalculation of the Diffusion Tensor for Plasmas with Ion-Acoustic Turbulence, November 1995
27. W. Jansen: CANDYS/QA: Algorithms, Programs, and User's Manual, December 1995
28. H. Voss, J. Kurths and U. Schwarz: Reconstruction of Grand Minima of Solar Activity from  $\Delta^{14}\text{C}$  Data — Linear and Nonlinear Signal Analysis, January 1996
29. R. Braun and F. Feudel: Supertransient Chaos in the Two-Dimensional Complex Ginzburg-Landau Equation, March 1996
30. P. Maaß and A. Rieder: Wavelet-Accelerated Tikhonov-Phillips Regularization with Applications, June 1996
31. F. Feudel, N. Seehafer, B. Galanti and S. Rüdiger: Symmetry Breaking Bifurcations for the Magneto-hydrodynamic Equations with Helical Forcing, June 1996
32. N. Seehafer, E. Zienicke and F. Feudel: Absence of Magnetohydrodynamic Activity in the Voltage-Driven Sheet Pinch, June 1996
33. W. Jansen: A Note on the Determination of the Type of Communication Areas, July 1996
34. Ch. Böckmann: Study of Error Sensitivities of Different Inverse Techniques for Determination of Aerosol Size Distribution, July 1996

# Inhaltsverzeichnis

<b>1. Einführung</b>	<b>1</b>
<b>2. Motivation für die nichtlineare Dynamik</b>	<b>3</b>
2.1 Beobachtung . . . . .	3
2.1.1 Irreguläres zeitliches Verhalten . . . . .	4
2.1.2 Formenvielfalt . . . . .	10
2.1.3 Zelluläre Automaten & Perkolaton . . . . .	10
2.1.4 Mustererkennung . . . . .	10
2.1.5 Symbolische Dynamik . . . . .	10
2.2 Literatur . . . . .	10
2.3 Paradigmatische Gleichungen . . . . .	10
2.3.1 Mathematisch gut untersuchte nichtlineare Gleichungen . . . . .	11
2.3.2 Wichtige nichtlineare Gleichungen der Physik . . . . .	11
<b>3. Logistische Abbildung (Parabel-Abbildung)</b>	<b>12</b>
3.1 Abbildungsgleichung (P.F. Verhulst 1845) . . . . .	13
3.2 Iterierte: $f^N(x)$ . . . . .	13
3.3 Fixpunkte und Stabilität . . . . .	19
3.4 Feigenbaum - Konstanten . . . . .	20
3.4.1 $\alpha = 2.502, 907, 875, 0\dots$ . . . . .	20
3.4.2 $\delta = 4.669, 201, 990, 97\dots$ . . . . .	21
3.5 Lyapunov-Exponent $\lambda$ . . . . .	21
3.5.1 $r \in [0, 1]$ . . . . .	21
3.5.2 $r \in (1, 3]$ . . . . .	22
3.5.3 $r \in (3, 3.5699\dots)$ . . . . .	22
3.5.4 $r \in (3.5699\dots, 4]$ . . . . .	22
3.6 Literatur . . . . .	22
<b>4. Lorenz - Gleichungen</b>	<b>25</b>
4.1 Rayleigh - Bénard - Konvektion . . . . .	25
4.1.1 Navier - Stokes - Gleichung . . . . .	25
4.1.2 Kontinuitätsgleichung . . . . .	25
4.1.3 Fouriersche Wärmeleitungsgleichung (1822) . . . . .	26
4.1.4 Saltzman - Gleichungen (1962) . . . . .	26
4.1.5 Lorenz - Gleichungen . . . . .	26
4.2 Langzeitvariabilität der Sonnenaktivität . . . . .	30
4.2.1 Beobachtungen . . . . .	30
4.2.2 Polarlichter . . . . .	31
4.2.3 Sonnenflecken . . . . .	31
4.2.4 Erdmagnetische Störungen . . . . .	33

4.2.5	$\Delta^{14}\text{C}$ Häufigkeit . . . . .	33
4.2.6	Große Minima . . . . .	35
4.2.7	Dynamo - Modell für die Sonnenaktivität . . . . .	35
4.2.8	Malinetzky-Ruzmaikin Modell . . . . .	37
4.3	Literatur . . . . .	38
<b>5.</b>	<b>Fraktale &amp; Selbstähnlichkeit</b>	<b>39</b>
5.1	Formenvielfalt in der Natur . . . . .	39
5.2	Mathematische Monster . . . . .	39
5.2.1	Kapazität . . . . .	39
5.2.2	Kochsche Schneeflocke . . . . .	39
5.2.3	Cantor-Staub . . . . .	40
5.2.4	Logistische Abbildung . . . . .	40
5.3	Dynamische Systeme . . . . .	40
5.4	Landschaftsgeneratoren . . . . .	40
5.5	Literatur . . . . .	40
<b>6.</b>	<b>Die Brownsche Bewegung</b>	<b>42</b>
6.1	Entdeckung . . . . .	42
6.2	Beginn der stochastischen Modellierung von Naturerscheinungen . . . . .	43
6.3	Experimentelle Bestätigung der Theorie von Einstein – Smoluchovski . . . . .	44
6.4	Kurze Geschichte der Atomtheorie . . . . .	44
6.4.1	Antiker Atomismus . . . . .	44
6.4.2	Erste Abschätzung der Loschmidtschen Zahl . . . . .	45
6.4.3	Die Chemie entdeckt das Atom . . . . .	45
6.4.4	Kinetische Gastheorie 1856 . . . . .	45
6.5	Herleitungen . . . . .	46
6.5.1	Plausible Herleitung der Einsteinschen Gleichung . . . . .	46
6.5.2	Einsteins Herleitung . . . . .	46
6.5.3	Langevins Herleitung (1908) . . . . .	47
6.6	Simulation der Brownschen Bewegung . . . . .	48
6.7	Bestimmung der Loschmidtschen Zahl . . . . .	48
6.8	Literatur . . . . .	48
<b>7.</b>	<b>Stöße &amp; Billards</b>	<b>51</b>
7.1	Stoßgesetze für Punktmassen . . . . .	51
7.1.1	Gerader Stoß für $v_2 = 0$ . . . . .	51
7.1.2	Kugelstoß . . . . .	52
7.1.3	Streu-Phänomene . . . . .	52
7.2	Elastische Festkörper . . . . .	52
7.2.1	Kugelstoßkette – Heinrich Hertz . . . . .	52
7.3	Unelastische Festkörper . . . . .	53
7.4	Billards . . . . .	53
7.5	Literatur . . . . .	53
<b>8.</b>	<b>Körper mit gravitativer Wechselwirkung</b>	<b>54</b>
8.1	Zwei - Körper - Problem . . . . .	54
8.2	Drei - Körper - Problem . . . . .	54
8.3	Beschränktes Drei - Körper - Problem . . . . .	54
8.4	Teilchen Simulation . . . . .	54
8.5	Literatur . . . . .	55



<b>9. GLOSSAR</b>	<b>56</b>
<b>10. Turbo-Pascal-Texte</b>	<b>61</b>
<b>11. IDL-Texte</b>	
<b>12. REDUCE-Texte</b>	
<b>Danksagung</b>	

# Abbildungsverzeichnis

2.1	Lichtkurve der Zwergnova SS Cygni . . . . .	3
2.2	Radiostrahlungsfluß der Sonne während eines Ausbruchs . . . . .	4
2.3	Umpolungen des Erdmagnetfeldes . . . . .	5
2.4	Zeitdauern für eine feste Erdmagnetfeldrichtung . . . . .	5
2.5	Plasmadichte in 2000 km Höhe . . . . .	6
2.6	DAX-Werte für 1990 . . . . .	6
2.7	Realisierung von weißem Rauschen (gleichverteilt) . . . . .	7
2.8	Realisierung von weißem Rauschen (normalverteilt) . . . . .	8
2.9	Realisierung einer 2D-Zufallsbewegung (gleichverteilt) . . . . .	8
2.10	Realisierung einer 2D-Zufallsbewegung (normalverteilt) . . . . .	9
2.11	Realisierung eines autoregressiven Prozesses . . . . .	9
3.1	Logistische Parabeln $f(x, r)$ . . . . .	12
3.2	Grafische Iteration für $r = 3$ . . . . .	14
3.3	Grafische Iteration für $r = 3.12$ . . . . .	14
3.4	Grafische Iteration für $r = 3.2$ . . . . .	15
3.5	Grafische Iteration für $r = 3.5699$ . . . . .	15
3.6	Grafische Iteration für $r = 3.67$ . . . . .	16
3.7	Grafische Iteration für $r = 4$ . . . . .	16
3.8	$f^2(x, r)$ . . . . .	17
3.9	$f^3(x, r)$ . . . . .	17
3.10	$f^4(x, r)$ . . . . .	18
3.11	Feigenbaum-Diagramm . . . . .	20
3.12	Lyapunov-Spektrum der logistischen Abbildung . . . . .	21
4.1	x-Komponente des Lorenz-Attraktors . . . . .	26
4.2	z-Komponente des Lorenz-Attraktors . . . . .	27
4.3	x-y-Projektion des Lorenz-Attraktors . . . . .	27
4.4	3D-Darstellung des Lorenz-Attraktors . . . . .	28
4.5	3D-Darstellung des Lorenz-Attraktors . . . . .	28
4.6	Sonnenfleckenbeobachtung mit bloßem Auge . . . . .	32
4.7	Sonnenflecken-Relativzahlen . . . . .	32
4.8	Minutenwerte des AE-Index der Erde . . . . .	34
4.9	$\Delta^{14}\text{C}$ Häufigkeit der letzten 9000 Jahre . . . . .	34
4.10	Beobachtungen zur solaren Aktivität innerhalb der letzten 1000 Jahre . . . . .	36
6.1	Realisierung einer 1-dimensionalen Brownschen Bewegung . . . . .	42
6.2	Realisierung einer 2-dimensionalen Brownschen Bewegung . . . . .	43

# 1. Einführung

Das vorliegende Skriptum<sup>1</sup> entstand während meiner einjährigen (1992/93) Zusammenarbeit mit Schülern des Gymnasiums in Kleinmachnow. Bei diesem Kurs konnte ich davon ausgehen, daß die Schüler virtuos mit ihren PCs umgehen können, bereits fraktale Mengen auf den Bildschirmen ihrer Computer sichtbar gemacht haben und das eine oder andere populäre Buch zur nichtlinearen Dynamik gelesen haben. Die sehr günstige Situation des Gymnasiums, mit einem Wohnheim ausgestattet zu sein, erlaubte die Arbeit in den Abendstunden zwischen 19 und 22 Uhr.

Das Ziel der wöchentlich stattgefundenen Arbeitsgemeinschaft war es, den Schülern eine Einführung in einige grundlegende Begriffe der nichtlinearen Dynamik zu geben, deren Beziehung zu physikalischen Fragestellungen zu verdeutlichen und auf die damit verknüpfte Vielzahl offener Fragen aufmerksam zu machen. Gleichzeitig kam es mir darauf an zu zeigen, daß die Ansätze der nichtlinearen Dynamik ein Aspekt der Naturbeschreibung neben vielen anderen darstellen. Aus der Spiegel-Serie<sup>2</sup> über *Ausbreitung und Mißbrauch der "Chaostheorie"* geht unter anderem hervor, daß eine eingehende Betrachtung der Grundideen und Aussagen der nichtlinearen Dynamik von großer Wichtigkeit ist.

Der z.T. sehr unterschiedliche Umfang der Ausarbeitung der einzelnen Kapitel ist mehr oder weniger zufällig<sup>3</sup>. Die vorliegende Arbeit ist noch nicht abgeschlossen. Die Erfahrungen, die sich aus der laufenden Arbeitsgemeinschaft ergeben, werden noch eingearbeitet.

Als einführende Schriften, die die verschiedensten Aspekte der nichtlinearen Dynamik, ausgehend von naturwissenschaftlichen Fragestellungen, erläutern, sind zu empfehlen:

- Gleick, J. 1990 Chaos - die Ordnung des Universums. Vorstoß in Grenzbereiche der modernen Physik, Knauer, München
- Stewart, I. 1993 Spielt Gott Roulette? Insel Verlag, Frankfurt am Main - Leipzig
- Prigogine, I. & I. Stengers 1981 Dialog mit der Natur. Neue Wege naturwissenschaftlichen Denkens, Piper, München - Wien
- Eigen, M. & R. Winkler 1975 Das Spiel. Naturgesetze steuern den Zufall, Piper, München - Wien
- Haken, H. 1981 Erfolgsgeheimnisse der Natur. Synergetik: Die Lehre vom Zusammenwirken, Deutsche Verlagsanstalt, Stuttgart

---

<sup>1</sup>Das PostScript-File (3,550,125 Bytes) kann über die WWW-Page <http://www.agnld.uni-potsdam.de/~shw/shw> erreicht werden.

<sup>2</sup>Peter Brügge, Kult um das Chaos. Aberglaube oder Welterklärung? DER SPIEGEL 39/1993, 156-164; 40/1993, 232-241 & 41/1993, 7, 240-252. Repliken: Richter, P.H., H. Dullin & H.-O. Peitgen, 1994 Der SPIEGEL, das Chaos – und die Wahrheit, Phys. Bl. 50, 355-358, Adams, E. 1994 Antwort, Phys. Bl. 50, 359, Geisel, T. 1994 Leserbrief zum SPIEGEL-Titel "Chaostheorie", Phys. Bl. 50, 359, Bernd Schöne, Der Sturm im Spiegel, Chaotische Nachrichten Nr. 29, Mai 1994, S. 5

<sup>3</sup>Als ich z.B. den Abschnitt über die Brownsche Bewegung ausarbeitete, war ich wegen einer Grippe zu Hause – der Abschnitt bekam daher einen größeren Umfang.

## 1. Einführung

- Lauwerier, H. 1989 Fraktale verstehen und selbst programmieren, Wittig-Fachbuchverlag, Hückelhoven
- Peitgen, H.-O., H. Jürgens, D. Saupe, E. Maletsky, T. Perciante & L. Yunker et al. 1992 Chaos. Iteration, Sensitivität, Mandelbrot-Menge. Ein Arbeitsbuch, Springer-Verlag & Ernst Klett Schulbuchverlag
- Heinrichs, Georg 1993<sup>2</sup> Chaos: Einführung in eine neue physikalische Theorie, Aulis-Verlag Deubner & Co KG, Köln
- Schuster, H.G. 1988<sup>2</sup> Deterministic Chaos. An Introduction, Physik-Verlag, Weinheim
- Gaponov-Grekhov, A.V. & M.I. Rabinovich 1992 Nonlinearities in Action: Oscillation, Chaos, Order, Fractals, Springer-Verlag, Berlin
- Argyris, J., G. Faust & M. Haase (1994) Die Erforschung des Chaos, Vieweg, Braunschweig
- D.K. Arrowsmith & C.M. Place: Dynamische Systeme, Spektrum-Verlag, Weinheim 1994

Zu danken habe ich insbesondere Jürgen Kurths, dem Leiter der Max-Planck-Arbeitsgruppe *Nicht-lineare Dynamik* an der Universität Potsdam, für die Anregung zu dieser Arbeit und seine Unterstützung, und Stefan Schmidt, jetzt TU Berlin, für viele Anregungen und seine Hilfe bei der Anfertigung einiger Abbildungen für die vorliegende Arbeit.

## 2. Motivation für die nichtlineare Dynamik

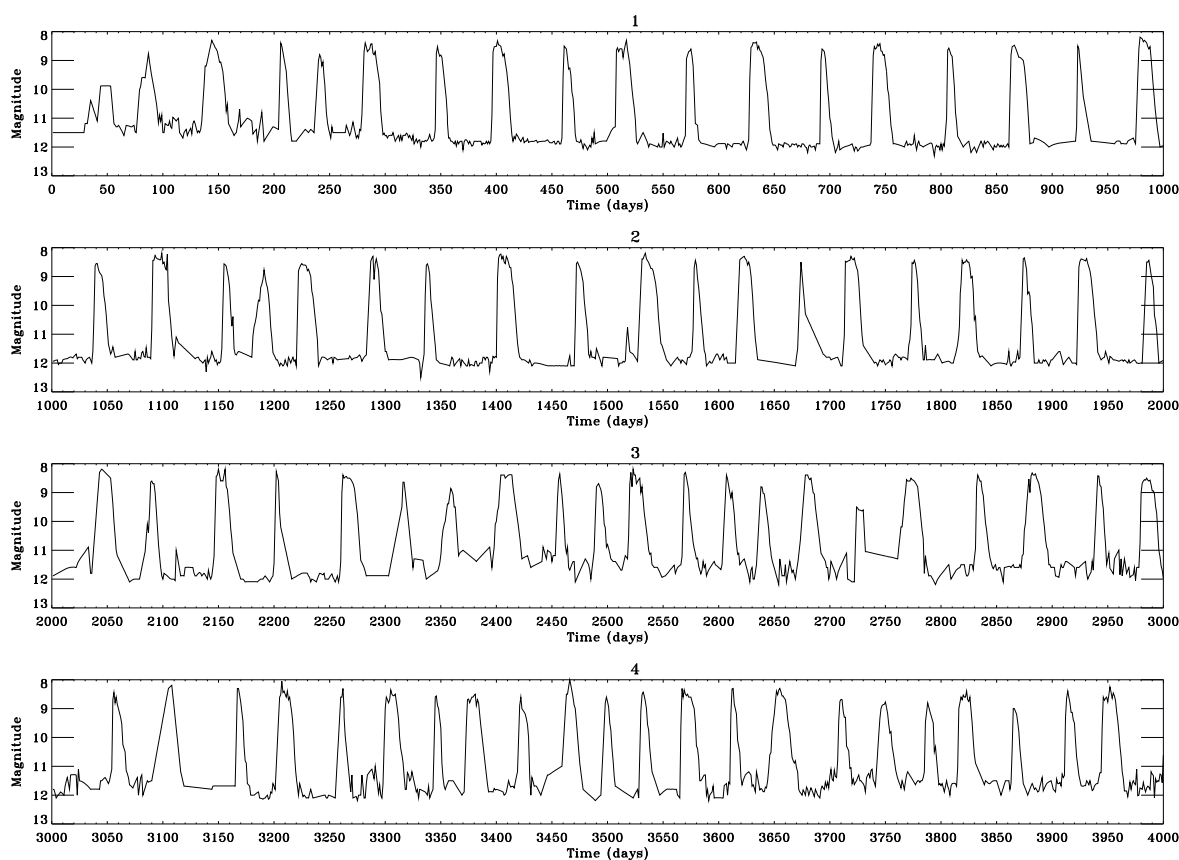
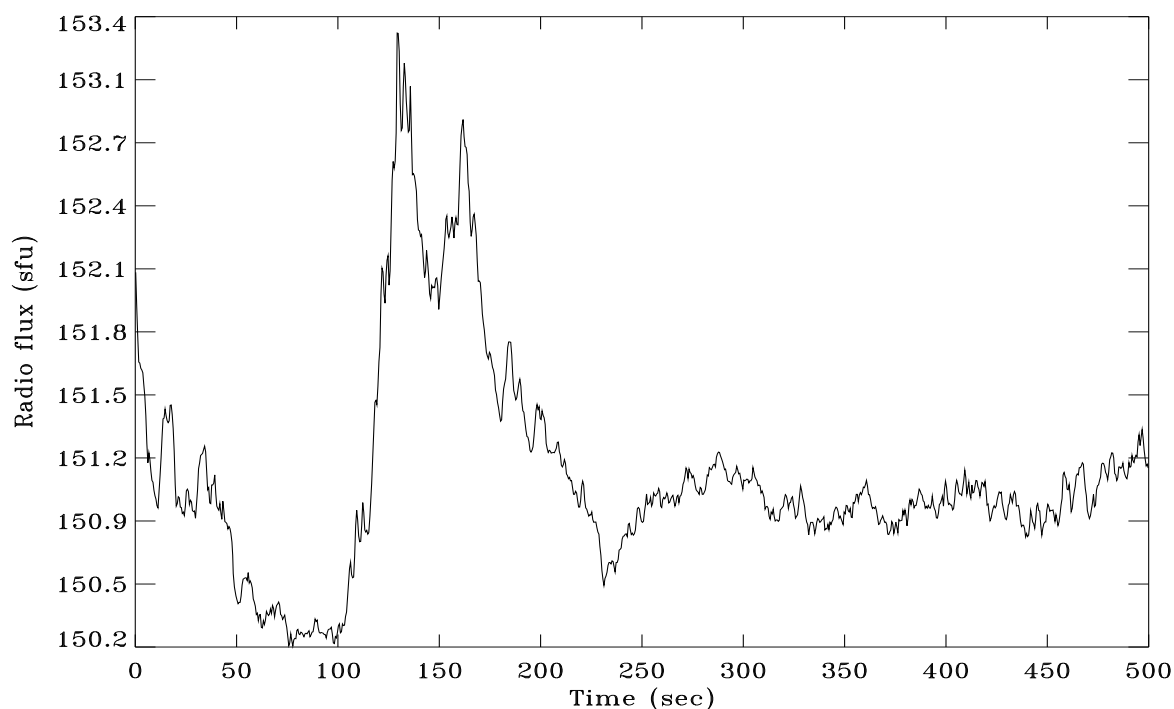


Abbildung 2.1: Die Tagesmittelwerte der scheinbaren Helligkeit der Zwergnova SS Cyg für den Zeitraum von 1896 bis 1907 (AAVSO). Deutlich sichtbar heben sich die Ausbruchshelligkeiten von der Grundhelligkeit bei 11.5 *mag* ab. Der Helligkeitszuwachs entspricht einer etwa 40-fachen Zunahme der Leuchtkraft des Sterns. Die Ausbrüche finden im Mittel alle 50 Tage statt und dauern zwischen 7 und 14 Tage.

### 2.1 Beobachtung

Die im folgenden eingestreuten Abbildungen sollen die Vielfältigkeit des Zeitverhaltens von verschiedensten Größen, meist physikalischer Systeme, illustrieren. Durch diese Galerie von Zeitreihen soll nicht der Eindruck erweckt werden, daß allein die nichtlineare Dynamik als erklärender Ansatz in

## 2. Motivation für die nichtlineare Dynamik



**Abbildung 2.2:** Radiostrahlungsfluß der Sonne bei 37 GHz während eines Bursts am 23. März 1991 gegen 11:58 UT. Die Zeitauflösung ist 0.05 s. Die Messung erfolgte mit einem Radioteleskop in Metsähovi nahe Helsinki (S. Urpo).

Frage kommt.

### 2.1.1 Irreguläres zeitliches Verhalten

Man beobachtet für viele Merkmale der Realität aperiodische und irreguläre Bewegungen. Zeitliche Entwicklung von Zuständen mit kompliziertem Verhalten: Zeitreihen der Astrophysik (solare und stellare Aktivität, Veränderliche), Geophysik (Gezeiten, Variation des Erdmagnetfelds), Meteorologie (Wetter), Ökologie (Populationsdichte), Medizin (EEG, EKG), Soziologie, Wirtschaft (DAX).

**Quantitative Beschreibung des Regularitätsgrades:**

- Spektralanalyse, Wavelets
- Lorenz map, Return map
- Phasenraumrekonstruktion (Dimensionen, Vorhersage, Lyapunov Exponent)
- Symbolische Dynamik (Entropie, Komplexität, Transinformation)

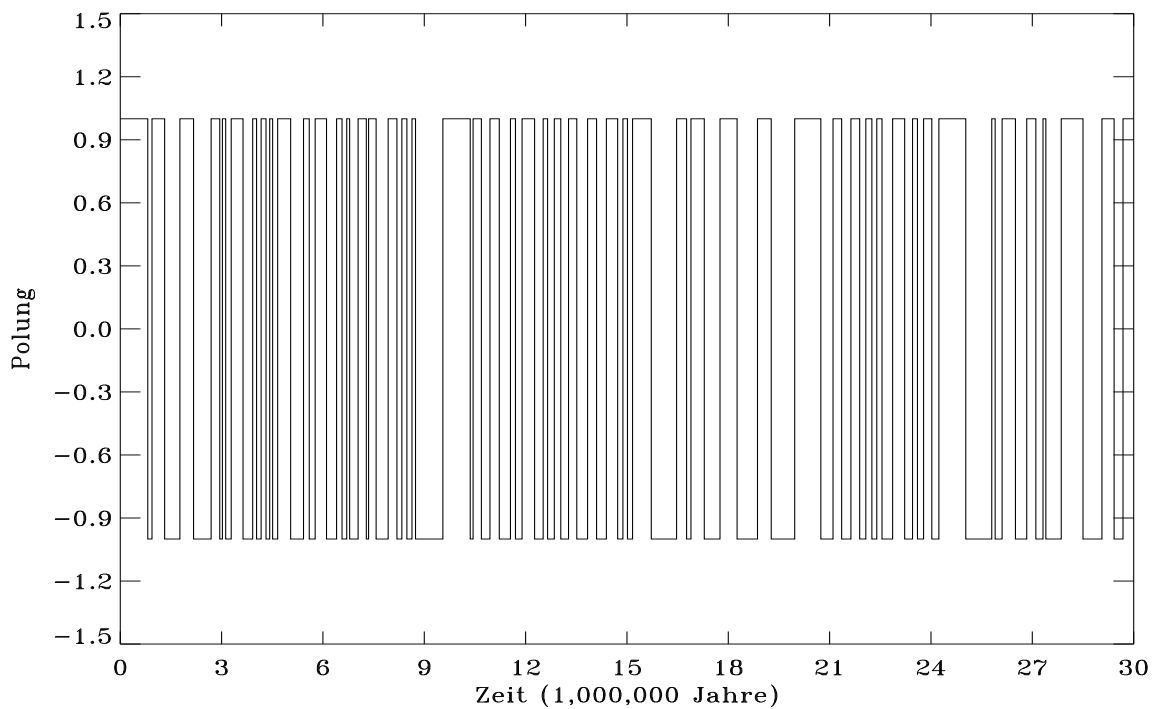
**Modell für Prozeß- und Meßrauschen**

Rauschfreier Prozeß:

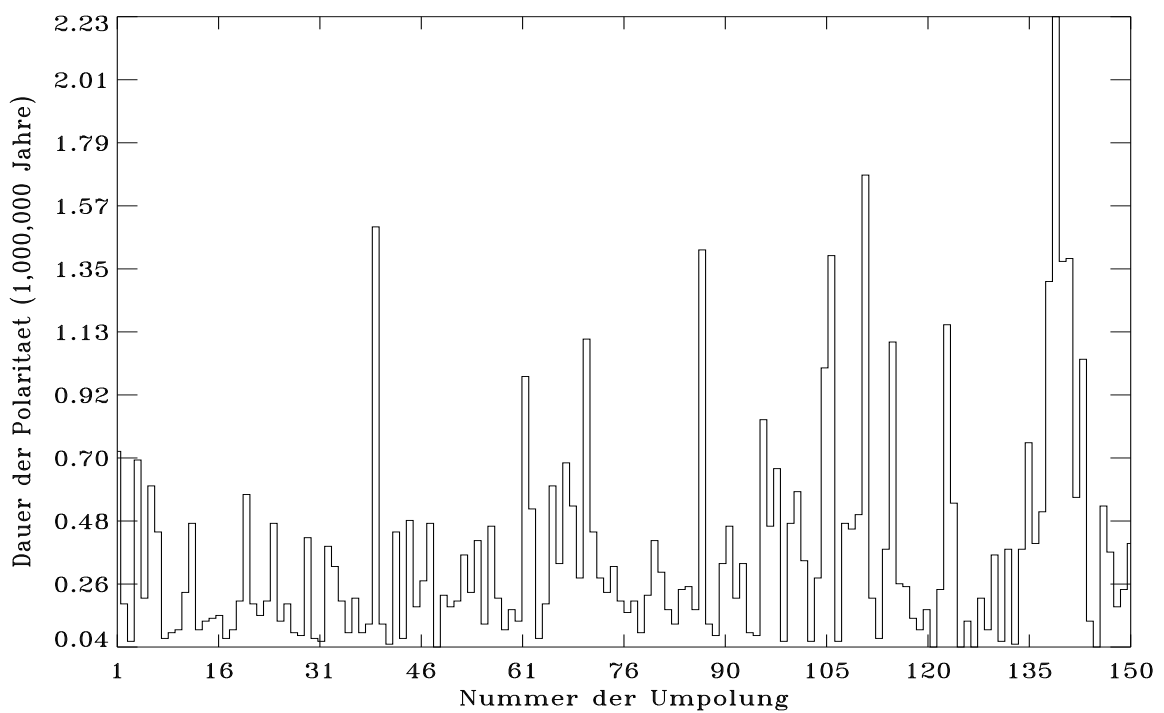
$$x_{n+1} = rx_n(1 - x_n) \quad (2.1)$$

Meßrauschen:

$$y_{n+1} = x_{n+1} + \eta_{n+1} \quad (2.2)$$

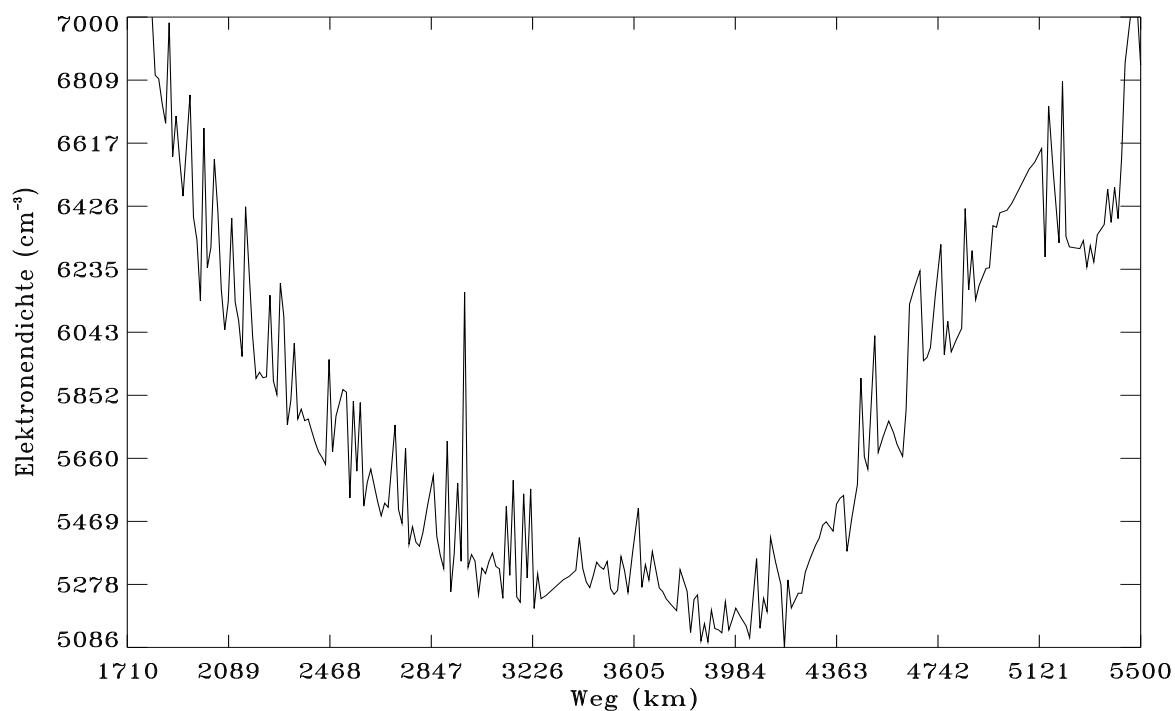


**Abbildung 2.3:** Zeitliche Entwicklung der Richtung des globalen magnetischen Dipols der Erde für die letzten 30 Megajahre. Das obere Niveau kennzeichnet die momentane Ausrichtung des Dipols.

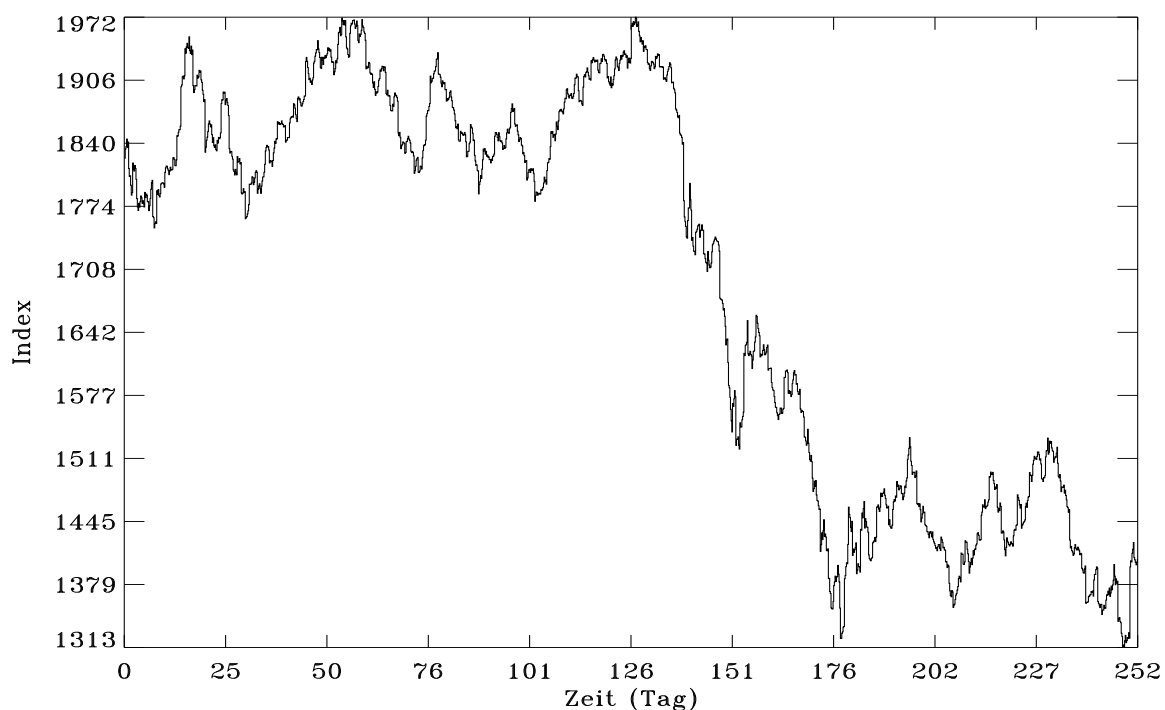


**Abbildung 2.4:** Zeitliche Entwicklung der Dauer für Ausrichtung des globalen magnetischen Dipols der Erde für die letzten 30 Megajahre.

## 2. Motivation für die nichtlineare Dynamik



**Abbildung 2.5: Plasmadichte längs der Flugbahn von IK 24 am 13. Juni 1990 in etwa 2000 km Höhe in höheren Breiten (Orbit 3112).**



**Abbildung 2.6: DAX-Werte (Deutscher Aktien Index) für mehr als acht Monate des Jahres 1990. Täglich wird dieser Index um 11:00, 11:30, 12:00, 12:30, 13:00 und 13:30 MEZ herausgegeben.**



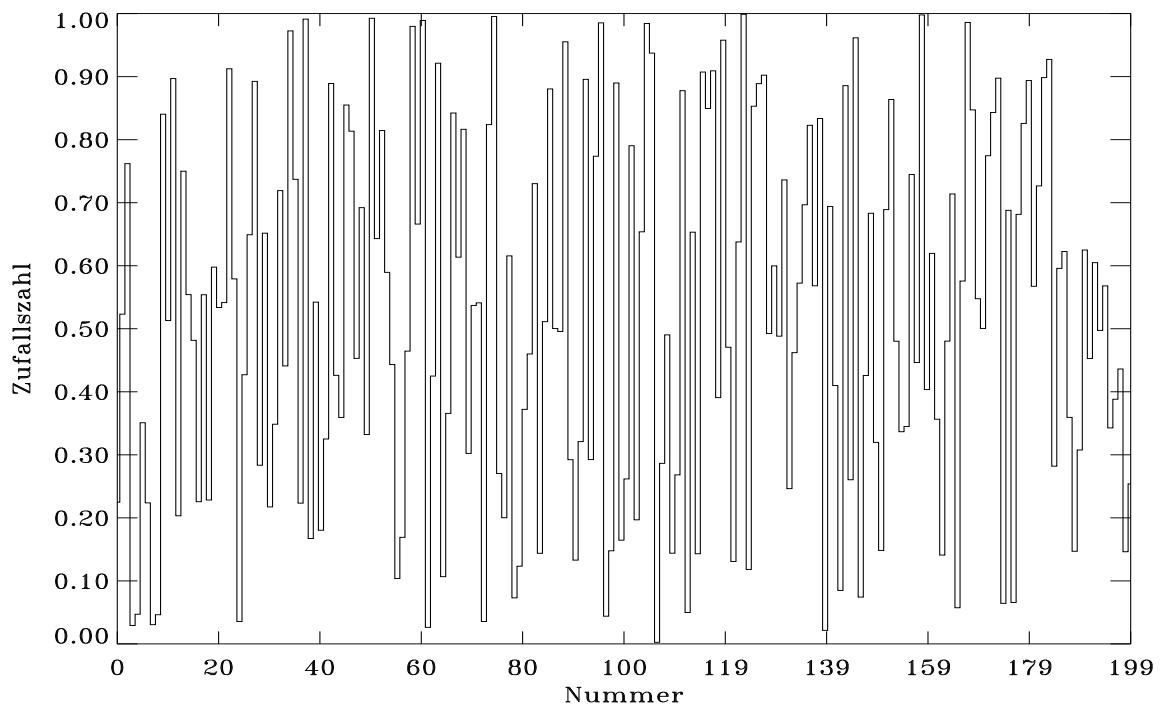
Äußere Störung der dynamischen Variablen:

$$x_{n+1} = rx_n(1 - x_n) + \eta_n \quad (2.3)$$

Äußere Störung eines Systemparameters:

$$x_{n+1} = (r + \eta_n)x_n(1 - x_n) \quad (2.4)$$

Ursache für das zeitlich irreguläre Verhalten in den Meßresultaten:



**Abbildung 2.7: Weißes Rauschen: Gleichmäßig verteilte Zufallszahlen aus dem Intervall [0,1].**

- Unkontrollierte Einflüsse (äußere Störungen)
  - Weißes Rauschen
  - Farbiges Rauschen (autoregressives Modell)
- Nichtlineare Terme in den Bewegungsgleichungen
  - Niedrig-dimensionales Chaos
  - Hoch-dimensionales Chaos (?)
- Heisenbergsche Unschärferelation

2. Motivation für die nichtlineare Dynamik

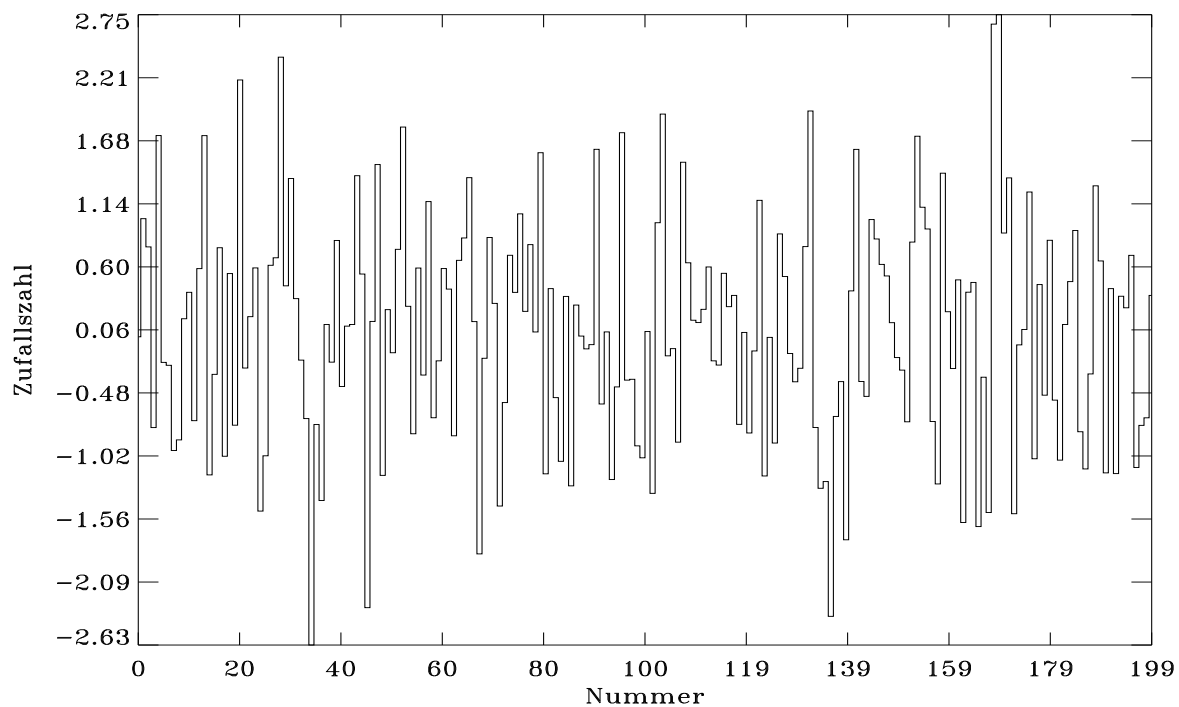


Abbildung 2.8: Weißes Rauschen: Gauß-verteilte Zufallszahlen mit der Varianz 1.

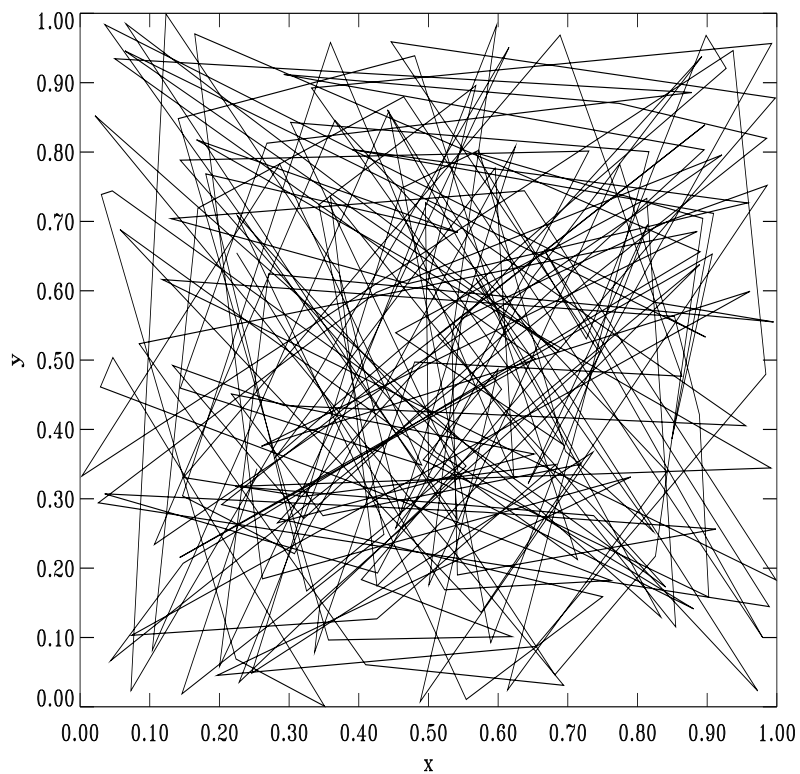


Abbildung 2.9: Zweidimensionale Zufallsbewegung bei gleichmäßig verteilten Zufallszahlen.

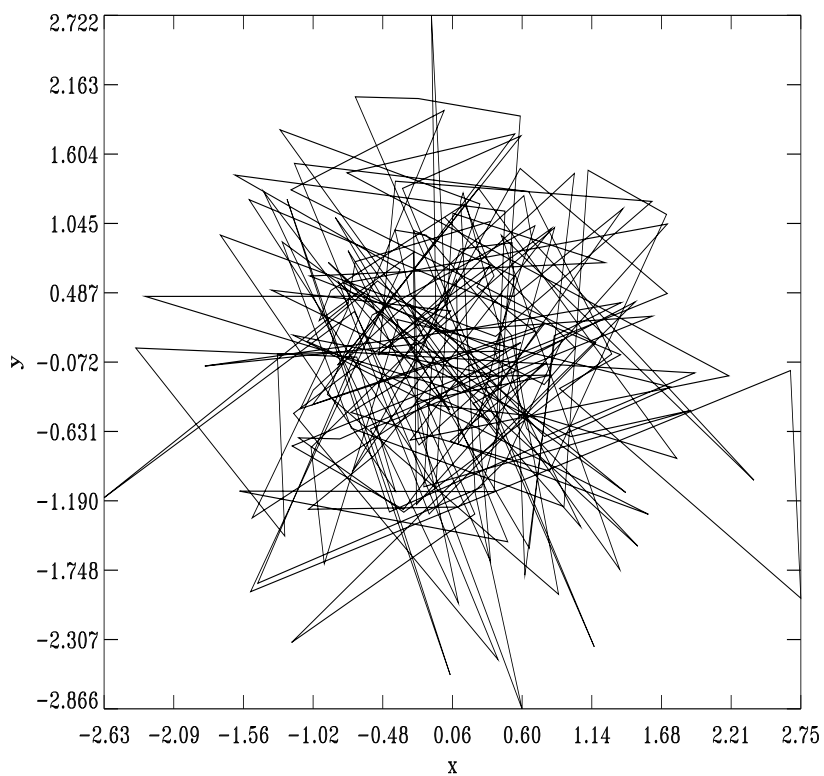


Abbildung 2.10: Zweidimensionale Zufallsbewegung bei Gauß-verteilter Zufallszahlen.

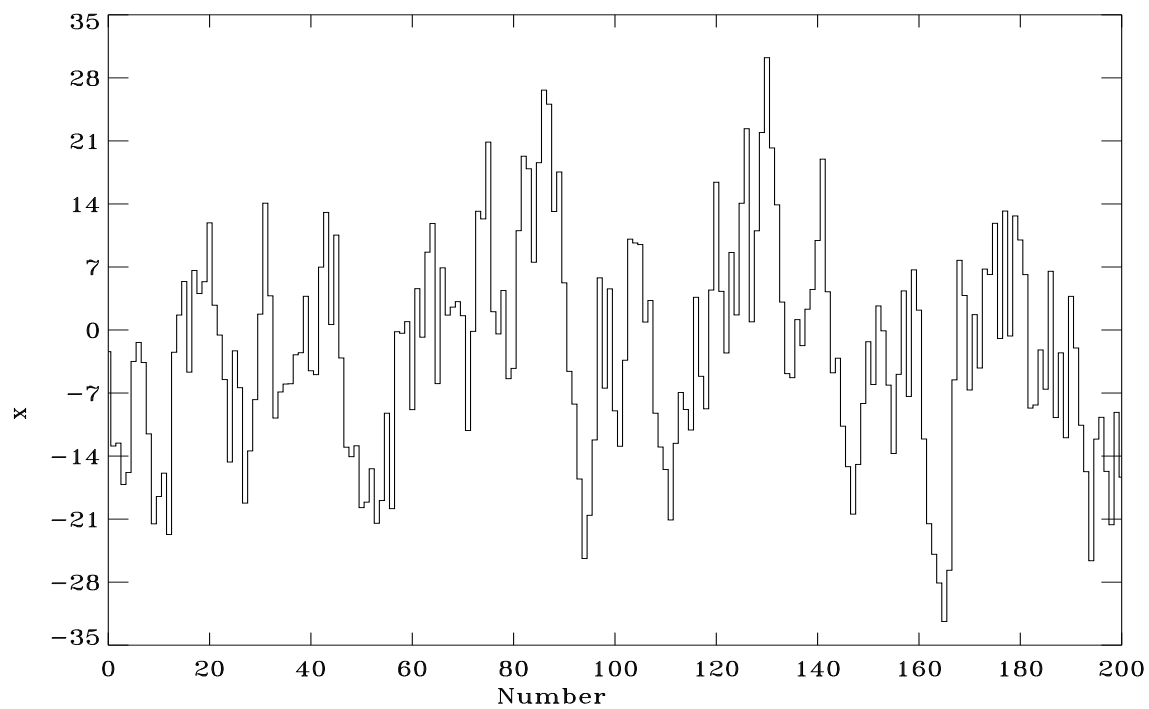


Abbildung 2.11: Realisierung eines autoregressiven Prozesses 1.-ter Ordnung mit  $a_1 = 0.7$  und  $\sigma = 10$ .

## 2. Motivation für die nichtlineare Dynamik

### 2.1.2 Formenvielfalt

Fraktale (Küstenlinien, Bruchflächen, Bahn eines Brownschen Teilchens, Galaxienverteilung)

- Koch-Kurve, fraktale Dimension
- Selbstähnlichkeit in der Natur
- Lindenmayer - Algorithmus
- Landschaftsgeneratoren
- Julia - Mengen & Apfelmännchen

### 2.1.3 Zelluläre Automaten & Perkolation

Hier bilden sich aus eng begrenzten Nachbarschaftsbeziehungen zwischen Maschen eines Netzes diffraktale Gebilde, Turbulenzen, Landschaften, wandernde Individuen. Die Entstehung des Komplexen aus dem Einfachen lädt zu allerlei Spekulationen ein, nicht zuletzt zum spielerischen Herumprobieren.

R. Mahnke, J. Schmelzer, G. Röpke, Nichtlineare Phänomene und Selbstorganisation, B.G. Teubner, Stuttgart 1992, p.191

Stauffer, Computers in Physics, 1991

V. Demmel, CA & Astrophysik

S. Wolfram, Theory and Applications of Cellular Automata, World Scientific, Singapore, 1986

Cellular Automata, Theory and Experiment, edited by H. Gutowitz, MIT Press, 1991

### 2.1.4 Mustererkennung

Neuronale Netze (OCR, **O**ptical **C**haracter **R**ecognition)

### 2.1.5 Symbolische Dynamik

Maße (Informations-Entropie), die z.T. Schätzungen von (bedingten) Wahrscheinlichkeiten erfordern.

## 2.2 Literatur

Hopcroft, J.E & J.D. Ullman (1994) Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, Addison-Wesley, Bonn

## 2.3 Paradigmatische Gleichungen

Linearität (Superpositionsprinzip)

$$f(\alpha_1 x_1 + \alpha_2 x_2) = \alpha_1 f(x_1) + \alpha_2 f(x_2) \quad (2.5)$$

Danach sind nichtlineare Funktionsgleichungen:

$$y = mx + n \quad (2.6)$$

$$y = x^2 \quad (2.7)$$

Ein bekanntes Beispiel für eine nichtlineare Additionsvorschrift ist das Einsteinsche Additionstheorem der Geschwindigkeiten für  $v_i \rightarrow c$

$$v = \frac{v_1 + v_2}{1 + \frac{v_1 \cdot v_2}{c^2}} \quad (2.8)$$

### 2.3.1 Mathematisch gut untersuchte nichtlineare Gleichungen

- Logistische Abbildung (Populationsdynamik)
- Lorenz - Gleichungssystem (Konvektion)

### 2.3.2 Wichtige nichtlineare Gleichungen der Physik

- Navier - Stokes - Gleichung (Turbulenz)
- Dynamo - Gleichungen (Magnetfelder von Planeten, Sternen und Galaxien)

### 3. Logistische Abbildung (Parabel-Abbildung)

Dieses Kapitel soll deutlich werden lassen, daß die Berücksichtigung nichtlinearer Terme in den Bewegungsgleichungen eine drastische Änderung für deren Lösungsverhalten bedeutet.

Mathematisches Pendel

$$\ddot{x} + \frac{g}{m} \sin x = 0 \tag{3.1}$$

$$\ddot{x} + \omega^2 x = 0 \tag{3.2}$$

Parametrisch-angeregtes Pendel (Pompe)

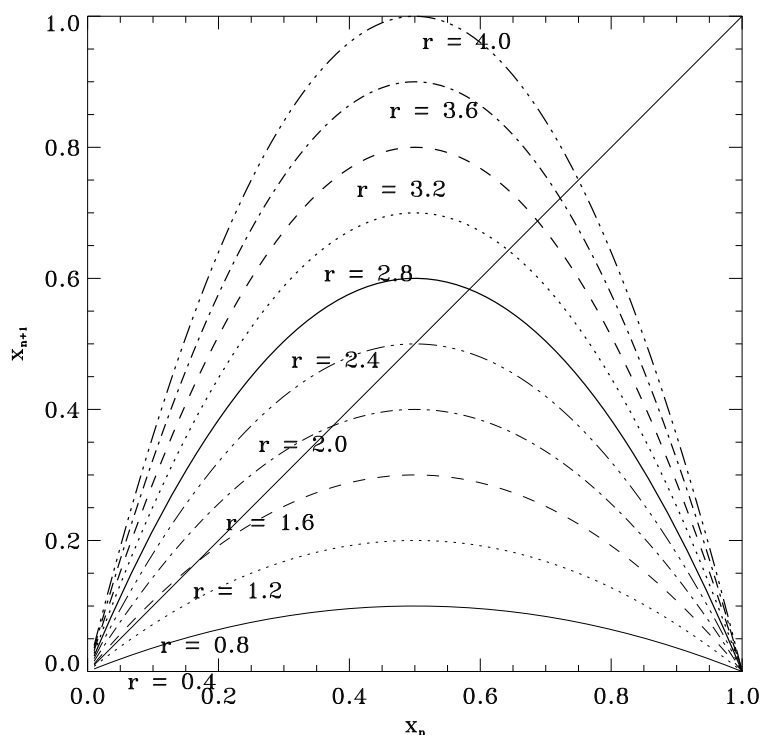


Abbildung 3.1: Logistische Parabeln  $f(x, r)$  für 10 verschiedene Werte des Kontrollparameters  $r$ .

### 3.1 Abbildungsgleichung (P.F. Verhulst 1845)

Iterationsgleichung:

$$x_{n+1} = f(x_n), \quad f(x_n) = r x_n (1 - x_n), \quad x_n \in [0, 1] \quad (3.3)$$

$r$  = Kontrollparameter mit  $r \in [0, 4]$

Ableitung von Gl. 3.3:

$$f'(x_n) = r(1 - 2x_n), \quad x_n \in [0, 1] \quad (3.4)$$

Merkmale physikalischer Systeme, die mit der Nichtlinearität von Gleichungen modelliert werden können:

- Qualitative Veränderung des asymptotischen Verhaltens eines physikalischen Systems in Abhängigkeit von den Systemparametern (quantitative Abhängigkeiten sind allgemein akzeptiert: Harmonischer Oszillator)
- Empfindlichkeit gegenüber Anfangsbedingungen
- Weg ins Chaos via Periodenverdopplung

Visualisierung des Lösungsverhaltens der Gl. 3.3:

- Zeitreihe  $x_n(r, n > 300)$  gemäß Gl. 3.3 in Abhängigkeit vom Kontrollparameter  $r$
- Graphische Iteration (Abb. 3.1 bis 3.7)
- Feigenbaum-Diagramm (Bifurkationsdiagramm: Abb. 3.11)
- Histogramm für  $x_n$  und die lokalen Expansionsraten (Anstiege)  $f'(x_n)$
- Farbplot zur Abhängigkeit von den Anfangsbedingungen:  $x_{10}$  über  $x_0$  und  $r$

### 3.2 Iterierte: $f^N(x)$

$$f^1(x) = f(x) = rx(-x + 1) \quad (3.5)$$

$$f^2(x) = f(f(x)) = r^2x(-rx^3 + 2rx^2 - rx - x + 1) \quad (3.6)$$

$$f^3(x) = f(f(f(x))) = r^3x(-r^4x^7 + 4r^4x^6 - 6r^4x^5 + 4r^4x^4 - r^4x^3 - 2r^3x^5 + 6r^3x^4 - 6r^3x^3 + 2r^3x^2 - r^2x^3 + 2r^2x^2 - r^2x - rx^3 + 2rx^2 - rx - x + 1) \quad (3.7)$$

$$f^4(x) = f(f(f(f(x)))) = r^4x(-r^{11}x^{15} + 8r^{11}x^{14} - 28r^{11}x^{13} + 56r^{11}x^{12} - 70r^{11}x^{11} + 56r^{11}x^{10} - 28r^{11}x^9 + 8r^{11}x^8 - r^{11}x^7 - 4r^{10}x^{13} + 28r^{10}x^{12} - 84r^{10}x^{11} + 140r^{10}x^{10} - 140r^{10}x^9 + 84r^{10}x^8 - 28r^{10}x^7 + 4r^{10}x^6 - 6r^9x^{11} + 36r^9x^{10} - 90r^9x^9 + 120r^9x^8 - 90r^9x^7 + 36r^9x^6 - 6r^9x^5 - 2r^8x^{11} + 12r^8x^{10} - 34r^8x^9 + 60r^8x^8 - 70r^8x^7 + 52r^8x^6 - 22r^8x^5 + 4r^8x^4 - 6r^7x^9 + 30r^7x^8 - 61r^7x^7 + 64r^7x^6 - 36r^7x^5 + 10r^7x^4 - r^7x^3 - 6r^6x^7 + 24r^6x^6 - 36r^6x^5 + 24r^6x^4 - 6r^6x^3 - r^5x^7 + 4r^5x^6 - 8r^5x^5 + 10r^5x^4 - 7r^5x^3 + 2r^5x^2 - r^4x^7 + 4r^4x^6 - 8r^4x^5 + 10r^4x^4 - 7r^4x^3 + 2r^4x^2 - 2r^3x^5 + 6r^3x^4 - 7r^3x^3 + 4r^3x^2 - r^3x - r^2x^3 + 2r^2x^2 - r^2x - rx^3 + 2rx^2 - rx - x + 1) \quad (3.8)$$

3. Logistische Abbildung (Parabel-Abbildung)

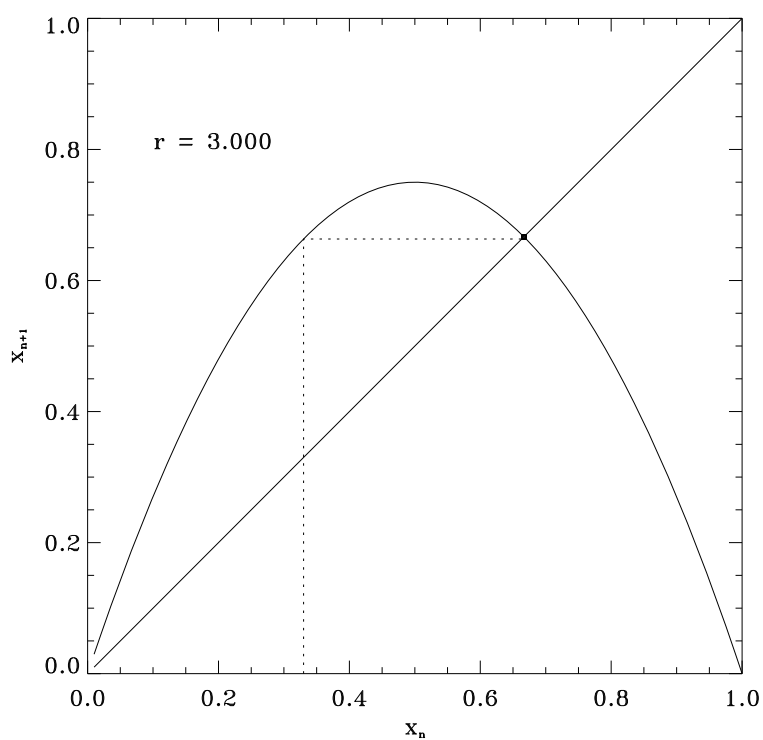


Abbildung 3.2: Grafische Iteration für die logistische Abbildung mit  $r = 3$ .

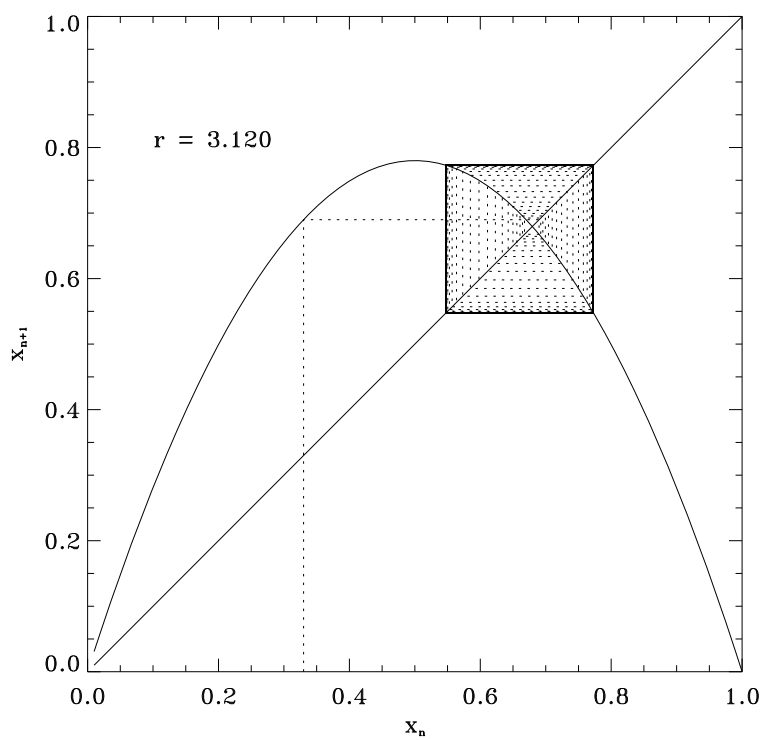


Abbildung 3.3: Grafische Iteration für die logistische Abbildung mit  $r = 3.12$ .



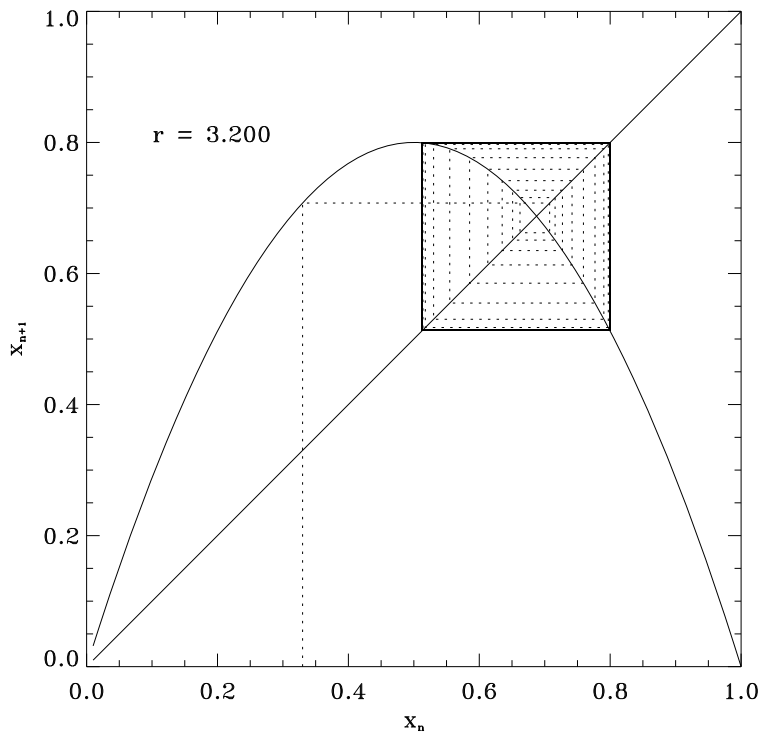


Abbildung 3.4: Grafische Iteration für die logistische Abbildung mit  $r = 3.2$ .

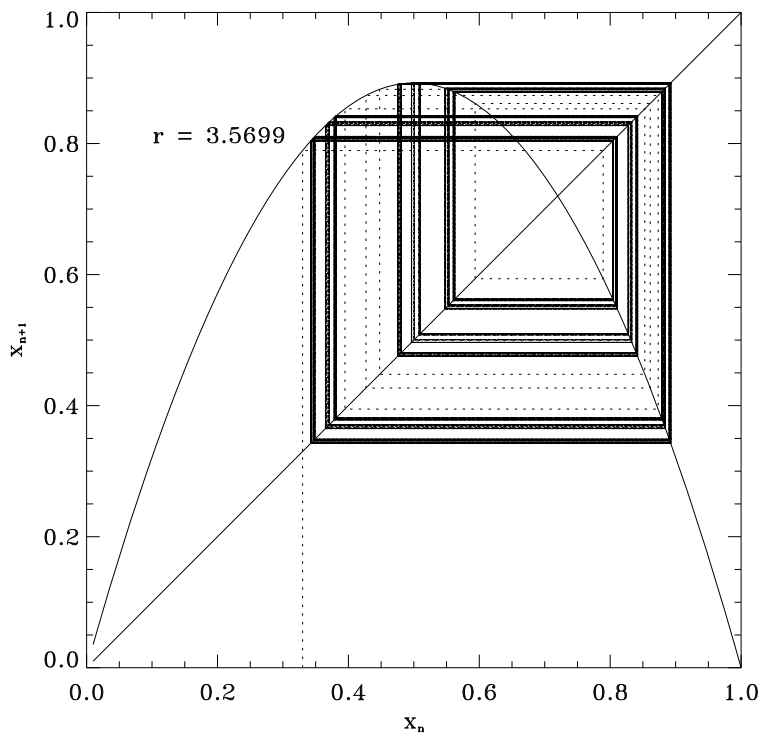


Abbildung 3.5: Grafische Iteration für die logistische Abbildung mit  $r = 3.5699$ .

3. Logistische Abbildung (Parabel-Abbildung)

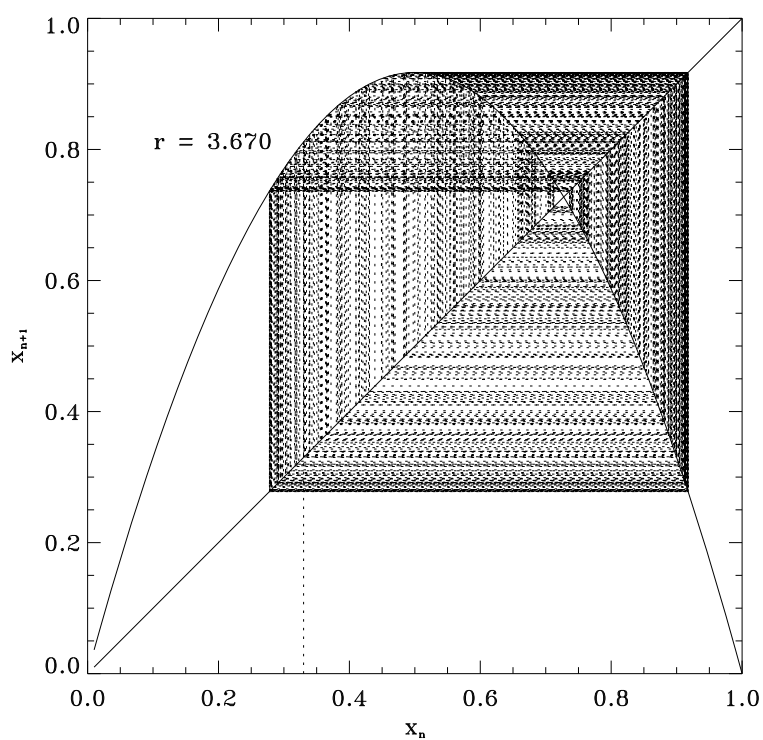


Abbildung 3.6: Grafische Iteration für die logistische Abbildung mit  $r = 3.67$ .

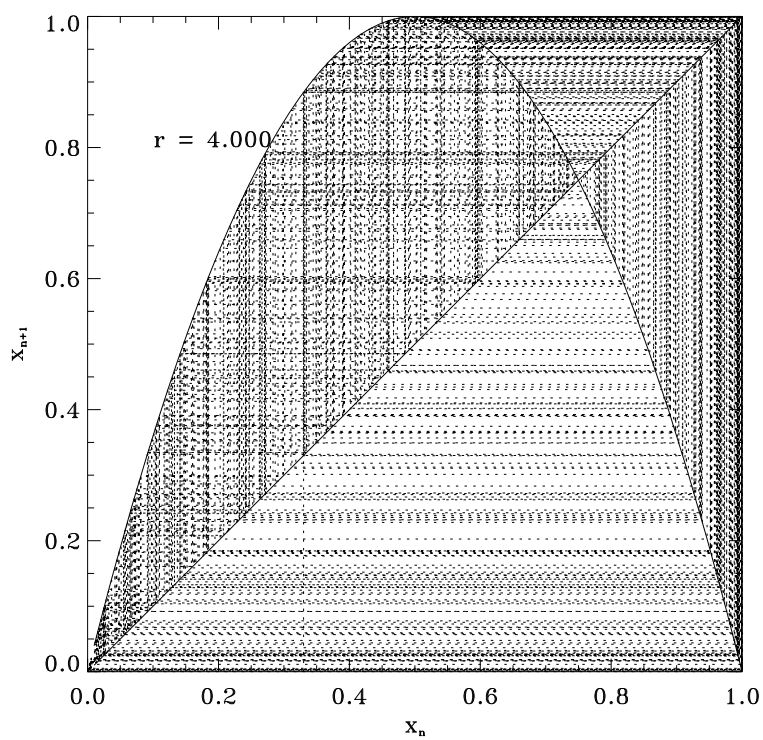


Abbildung 3.7: Grafische Iteration für die logistische Abbildung mit  $r = 4$ .

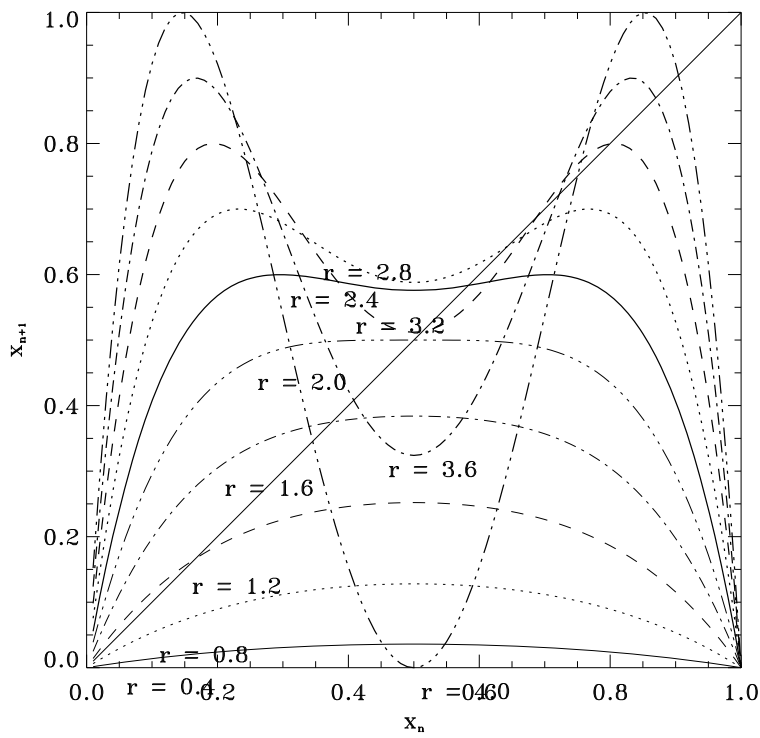


Abbildung 3.8: Die Abbildung  $f^2(x)$  für 10 verschiedene Werte des Kontrollparameters  $r$ .

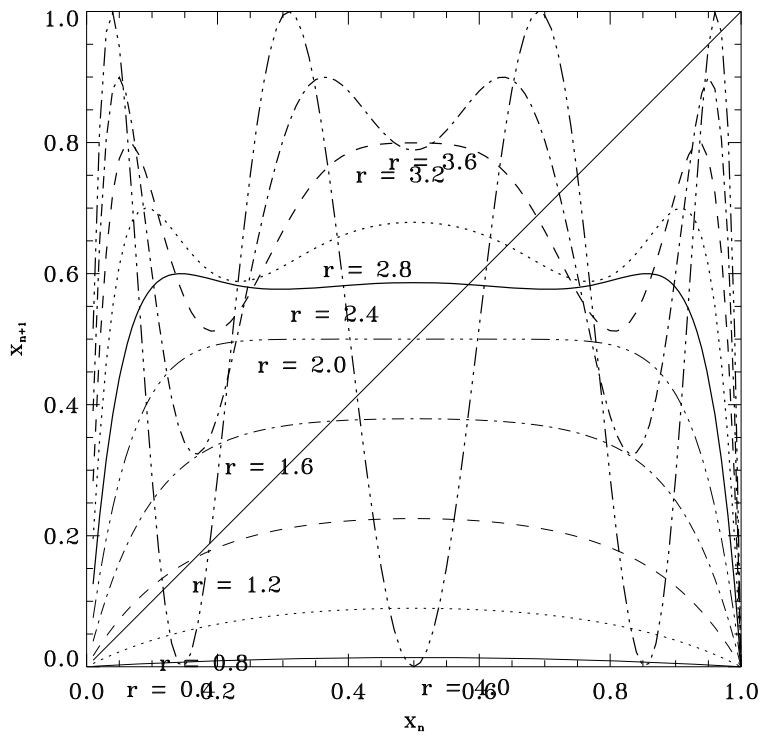


Abbildung 3.9: Die Abbildung  $f^3(x)$  für 10 verschiedene Werte des Kontrollparameters  $r$ .

### 3. Logistische Abbildung (Parabel-Abbildung)

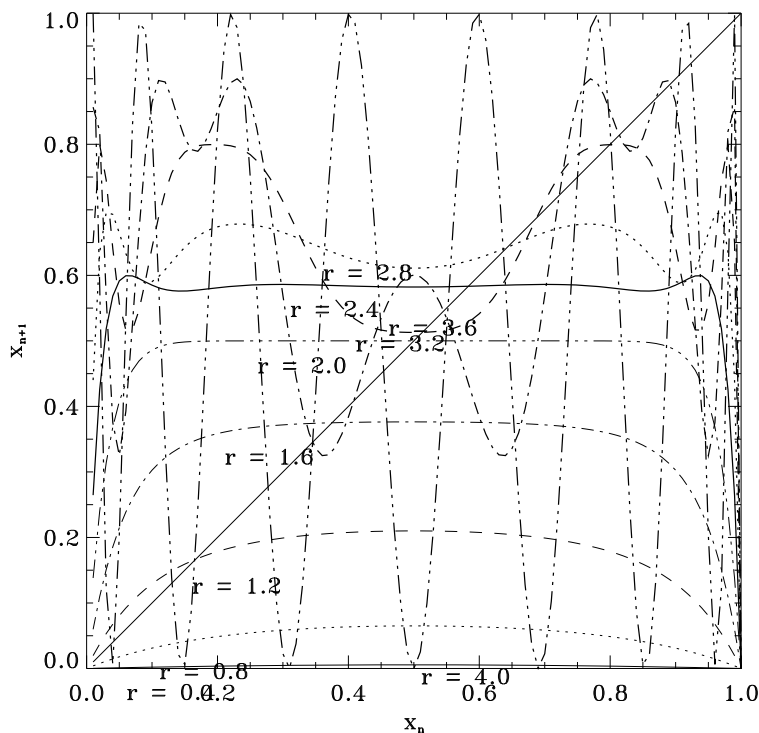


Abbildung 3.10: Die Abbildung  $f^4(x)$  für 10 verschiedene Werte des Kontrollparameters  $r$ .

$$\begin{aligned}
 f^5(x) = f(f(f(f(f(x)))))) = & r^5 x (-r^{26} x^{31} + 16r^{26} x^{30} - 120r^{26} x^{29} + 560r^{26} x^{28} - 1820r^{26} x^{27} \\
 & + 4368r^{26} x^{26} - 8008 \\
 & r^{26} x^{25} + 11440r^{26} x^{24} - 12870r^{26} x^{23} + 11440r^{26} x^{22} - 8008r^{26} x^{21} + 4368r^{26} x^{20} \\
 & - 1820r^{26} x^{19} + 560r^{26} x^{18} - 120r^{26} x^{17} + 16r^{26} x^{16} - r^{26} x^{15} - 8r^{25} x^{29} + 120r^{25} x^{28} - \\
 & 840r^{25} x^{27} + 3640r^{25} x^{26} - 10920r^{25} x^{25} + 24024r^{25} x^{24} - 40040r^{25} x^{23} + \\
 & 51480r^{25} x^{22} - 51480r^{25} x^{21} + 40040r^{25} x^{20} - 24024r^{25} x^{19} + 10920r^{25} x^{18} - 3640r^{25} \\
 & x^{17} + 840r^{25} x^{16} - 120r^{25} x^{15} + 8r^{25} x^{14} - 28r^{24} x^{27} + 392r^{24} x^{26} - 2548r^{24} x^{25} \\
 & + 10192r^{24} x^{24} - 28028r^{24} x^{23} + 56056r^{24} x^{22} - 84084r^{24} x^{21} + 96096r^{24} x^{20} - 84084r^{24} x^{19} + \\
 & 56056r^{24} x^{18} - 28028r^{24} x^{17} + 10192r^{24} x^{16} - 2548r^{24} x^{15} + 392r^{24} x^{14} - 28r^{24} x^{13} - \\
 & 4r^{23} x^{27} + 56r^{23} x^{26} - 420r^{23} x^{25} + 2184r^{23} x^{24} - 8372r^{23} x^{23} + 24024r^{23} x^{22} - \\
 & 52052r^{23} x^{21} + 85800r^{23} x^{20} - 108108r^{23} x^{19} + 104104r^{23} x^{18} - 76076r^{23} x^{17} + \\
 & 41496r^{23} x^{16} - 16380r^{23} x^{15} + 4424r^{23} x^{14} - 732r^{23} x^{13} + 56r^{23} x^{12} - 28r^{22} x^{25} + 364r^{22} x^{24} - \\
 & 2254r^{22} x^{23} + 8848r^{22} x^{22} - 24640r^{22} x^{21} + 51436r^{22} x^{20} - 82698r^{22} x^{19} + 103488r^{22} x^{18} - \\
 & 100716r^{22} x^{17} + 75460r^{22} x^{16} - 42658r^{22} x^{15} + 17584r^{22} x^{14} - \\
 & 4984r^{22} x^{13} + 868r^{22} x^{12} - 70r^{22} x^{11} - 84r^{21} x^{23} + 1008r^{21} x^{22} - 5600r^{21} x^{21} \\
 & + 19096r^{21} x^{20} - 44660r^{21} x^{19} + 75768r^{21} x^{18} - 96096r^{21} x^{17} + 92400r^{21} x^{16} - 67452r^{21} x^{15} + \\
 & 36960r^{21} x^{14} - 14784r^{21} x^{13} + 4088r^{21} x^{12} - 700r^{21} x^{11} + 56r^{21} x^{10} - 6r^{20} \\
 & x^{23} + 72r^{20} x^{22} - 536r^{20} x^{21} + 2860r^{20} x^{20} - 10698r^{20} x^{19} + 28132r^{20} x^{18} - 53004r^{20} x^{17} + \\
 & 72792r^{20} x^{16} - 73530r^{20} x^{15} + 54576r^{20} x^{14} - 29376r^{20} x^{13} + 11132r^{20} x^{12} - \\
 & 2806r^{20} x^{11} + 420r^{20} x^{10} - 28r^{20} x^9 - 2r^{19} x^{23} + 24r^{19} x^{22} - 168r^{19} x^{21} + 836r^{19} x^{20} - 3110r^{19} x^{19} +
 \end{aligned}$$

$$\begin{aligned}
 & 8924r^{19}x^{18} - 20036r^{19}x^{17} + 35088r^{19}x^{16} - 47310r^{19}x^{15} + \\
 & 48272r^{19}x^{14} - 36480r^{19}x^{13} + 19812r^{19}x^{12} - 7370r^{19}x^{11} + 1724r^{19}x^{10} - 212r^{19}x^9 + 8r^{19}x^8 - \\
 & 12r^{18}x^{21} + 132r^{18}x^{20} - 750r^{18}x^{19} + 2880r^{18}x^{18} - 8094r^{18}x^{17} + \\
 & 17100r^{18}x^{16} - 27469r^{18}x^{15} + 33704r^{18}x^{14} - 31492r^{18}x^{13} + 22100r^{18}x^{12} - 11308r^{18} \\
 & x^{11} + 3992r^{18}x^{10} - 874r^{18}x^9 + 92r^{18}x^8 - r^{18}x^7 - 34r^{17}x^{19} + 340r^{17}x^{18} - \\
 & 1650r^{17}x^{17} + 5160r^{17}x^{16} - 11488r^{17}x^{15} + 18872r^{17}x^{14} - 23044r^{17}x^{13} + 20768r^{17}x^{12} - \\
 & 13570r^{17}x^{11} + 6228r^{17}x^{10} - 1898r^{17}x^9 + 344r^{17}x^8 - 28r^{17}x^7 - 6r^{16}x^{19} + \\
 & 60r^{16}x^{18} - 330r^{16}x^{17} + 1260r^{16}x^{16} - 3510r^{16}x^{15} + 7272r^{16}x^{14} - 11344r^{16}x^{13} \\
 & + 13348r^{16}x^{12} - 11694r^{16}x^{11} + 7400r^{16}x^{10} - 3206r^{16}x^9 + 864r^{16}x^8 - 118r^{16}x^7 + \\
 & 4r^{16}x^6 - 30r^{15}x^{17} + 270r^{15}x^{16} - 1150r^{15}x^{15} + 3080r^{15}x^{14} - 5776r^{15}x^{13} + 7952 \\
 & r^{15}x^{12} - 8176r^{15}x^{11} + 6260r^{15}x^{10} - 3490r^{15}x^9 + 1346r^{15}x^8 - 322r^{15}x^7 + 36 \\
 & r^{15}x^6 - 61r^{14}x^{15} + 488r^{14}x^{14} - 1760r^{14}x^{13} + 3780r^{14}x^{12} - 5368r^{14}x^{11} + 5272r^{14}x^{10} - \\
 & 3618r^{14}x^9 + 1700r^{14}x^8 - 515r^{14}x^7 + 88r^{14}x^6 - 6r^{14}x^5 - 6r^{13}x^{15} + \\
 & 48r^{13}x^{14} - 232r^{13}x^{13} + 784r^{13}x^{12} - 1786r^{13}x^{11} + 2708r^{13}x^{10} - 2738r^{13}x^9 + \\
 & 1832r^{13}x^8 - 784r^{13}x^7 + 196r^{13}x^6 - 22r^{13}x^5 - r^{12}x^{15} + 8r^{12}x^{14} - 52r^{12}x^{13} \\
 & + 224r^{12}x^{12} - 610r^{12}x^{11} + 1112r^{12}x^{10} - 1412r^{12}x^9 + 1252r^{12}x^8 - 749r^{12}x^7 + \\
 & 280r^{12}x^6 - 56r^{12}x^5 + 4r^{12}x^4 - r^{11}x^{15} + 8r^{11}x^{14} - 32r^{11}x^{13} + 84r^{11}x^{12} - \\
 & 190r^{11}x^{11} + 412r^{11}x^{10} - 718r^{11}x^9 + 862r^{11}x^8 - 669r^{11}x^7 + 320r^{11}x^6 - 86r^{11}x^5 + \\
 & 10r^{11}x^4 - 4r^{10}x^{13} + 28r^{10}x^{12} - 92r^{10}x^{11} + 188r^{10}x^{10} - 284r^{10}x^9 + \\
 & 364r^{10}x^8 - 389r^{10}x^7 + 296r^{10}x^6 - 134r^{10}x^5 + 28r^{10}x^4 - r^{10}x^3 - 8r^9x^{11} + \\
 & 48r^9x^{10} - 130r^9x^9 + 210r^9x^8 - 226r^9x^7 + 172r^9x^6 - 94r^9x^5 + 34r^9x^4 - 6 \\
 & r^9x^3 - 2r^8x^{11} + 12r^8x^{10} - 40r^8x^9 + 90r^8x^8 - 137r^8x^7 + 140r^8x^6 - \\
 & 94r^8x^5 + 38r^8x^4 - 7r^8x^3 - 6r^7x^9 + 30r^7x^8 - 67r^7x^7 + 88r^7x^6 - 74r^7x^5 + \\
 & 40r^7x^4 - 13r^7x^3 + 2r^7x^2 - 7r^6x^7 + 28r^6x^6 - 44r^6x^5 + 34r^6x^4 - \\
 & 13r^6x^3 + 2r^6x^2 - r^5x^7 + 4r^5x^6 - 10r^5x^5 + 16r^5x^4 - 13r^5x^3 + 4r^5x^2 - \\
 & r^4x^7 + 4r^4x^6 - 8r^4x^5 + 10r^4x^4 - 8r^4x^3 + 4r^4x^2 - r^4x - 2r^3x^5 + 6r^3x^4 \\
 & - 7r^3x^3 + 4r^3x^2 - r^3x - r^2x^3 + 2r^2x^2 - r^2x - rx^3 + 2rx^2 - rx - x + 1 \tag{3.9}
 \end{aligned}$$

### 3.3 Fixpunkte und Stabilität

Ein Punkt  $x^0$  mit  $x^0 = f(x^0)$  heißt **Fixpunkt** der Abbildung

$$x_{n+1} = f(x_n). \tag{3.10}$$

Eine Trajektorie  $\{x_n\}_{k=0}^{N-1}$  mit  $f^N(x_0) = x_0$  und  $f^i(x_0) \neq x_0$  für  $i=1, \dots, N-1$  heißt **periodischer Orbit der Periode  $N$** .

$$f'(x_n) \begin{cases} > 1 & : \text{ instabiler Fixpunkt} \\ < 1 & : \text{ stabiler Fixpunkt} \end{cases}$$

$x_{(1)}^0 = 0$  ist stets ein Fixpunkt.

$$x_{(2)}^0 = 1 - \frac{1}{r} \tag{3.11}$$

ist für  $r \geq 1$  ein Fixpunkt. Wegen  $f'(x_{(1)}^0) = r$  ist  $x_{(1)}^0$  asymptotisch stabil für  $r \in [0, 1]$ . Für  $r > 1$  ist  $x_{(1)}^0$  instabil.

### 3. Logistische Abbildung (Parabel-Abbildung)

$x_{(2)}^0$  ist wegen

$$f'(x_{(2)}^0) = 2 - r \quad (3.12)$$

für  $r \in (1, 3)$  stabil, anderenfalls instabil.

- Periodisches Regime  $[r_1 = 3, r_\infty = 3.569, 945, 6 \dots]$  Für große  $n$  hüpfen die  $x_n$  mit wachsendem  $r$  zuerst zwischen zwei Werten hin und her, dann werden mit weitersteigendem  $n$  wiederholt dieselben Zahlensequenzen, der Länge 4, 8,  $\dots 2^n$  durchlaufen. Bei  $r_\infty$  ist dann schließlich das Chaos ausgebrochen:  $x_n$  hüpfert im gesamten Intervall statistisch herum.
- Chaotisches Regime  $[r_\infty, 4]$ , aperiodische Bahnen, aber für die meisten Anfangswerte periodische Orbits mit riesiger Periode. Es gibt jedoch Bereiche, die vom Chaos gemieden werden. Im chaotischen Bereich hängt  $x_n$  für festes  $n$  zwar überaus empfindlich von  $x_0$  ab. Die Bereiche des Intervalls  $(0,1)$ , die die Folge bevorzugt besucht, und die Wahrscheinlichkeit, mit der  $x_n$  in einem gegebenen Intervall angetroffen wird, sind aber von  $x_0$  unabhängig und können berechnet werden (Histogramme für verschiedene  $r$ ).

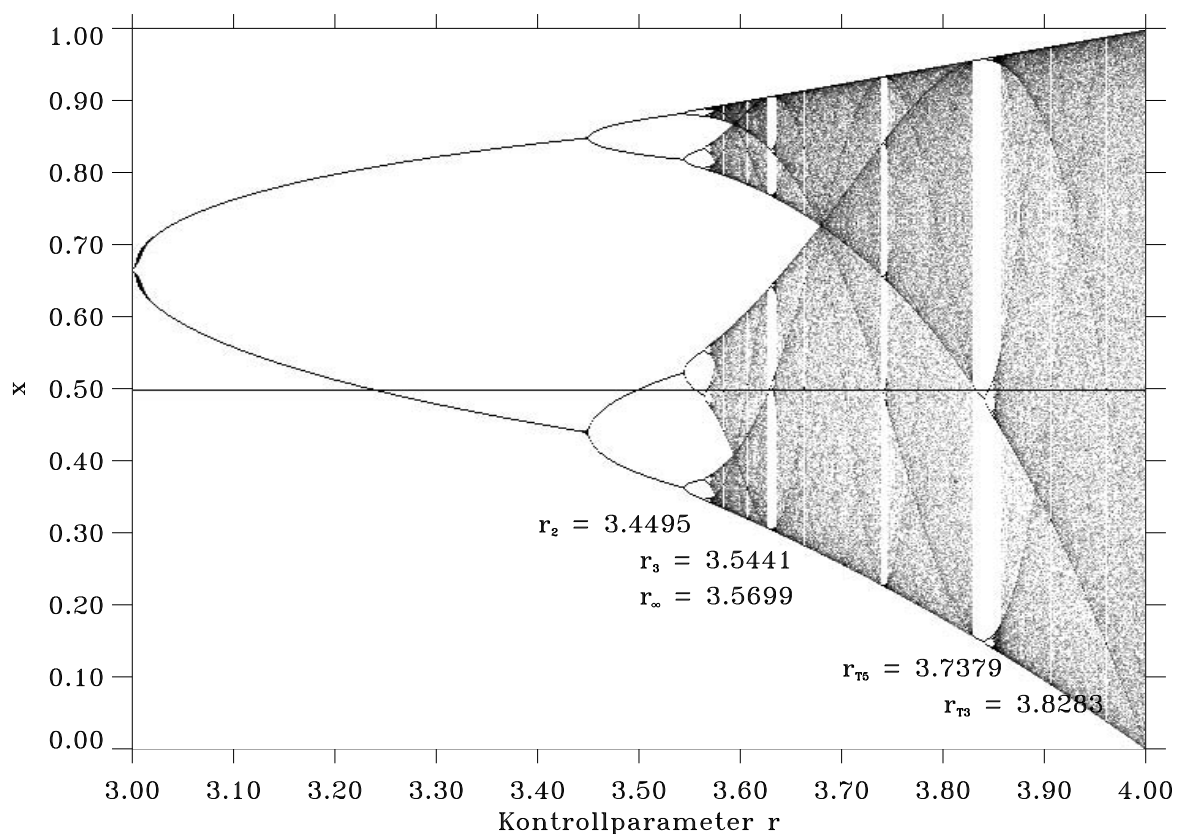


Abbildung 3.11: Das Feigenbaum-Diagramm für das Kontrollparameterintervall  $r \in [3, 4]$ .

## 3.4 Feigenbaum - Konstanten

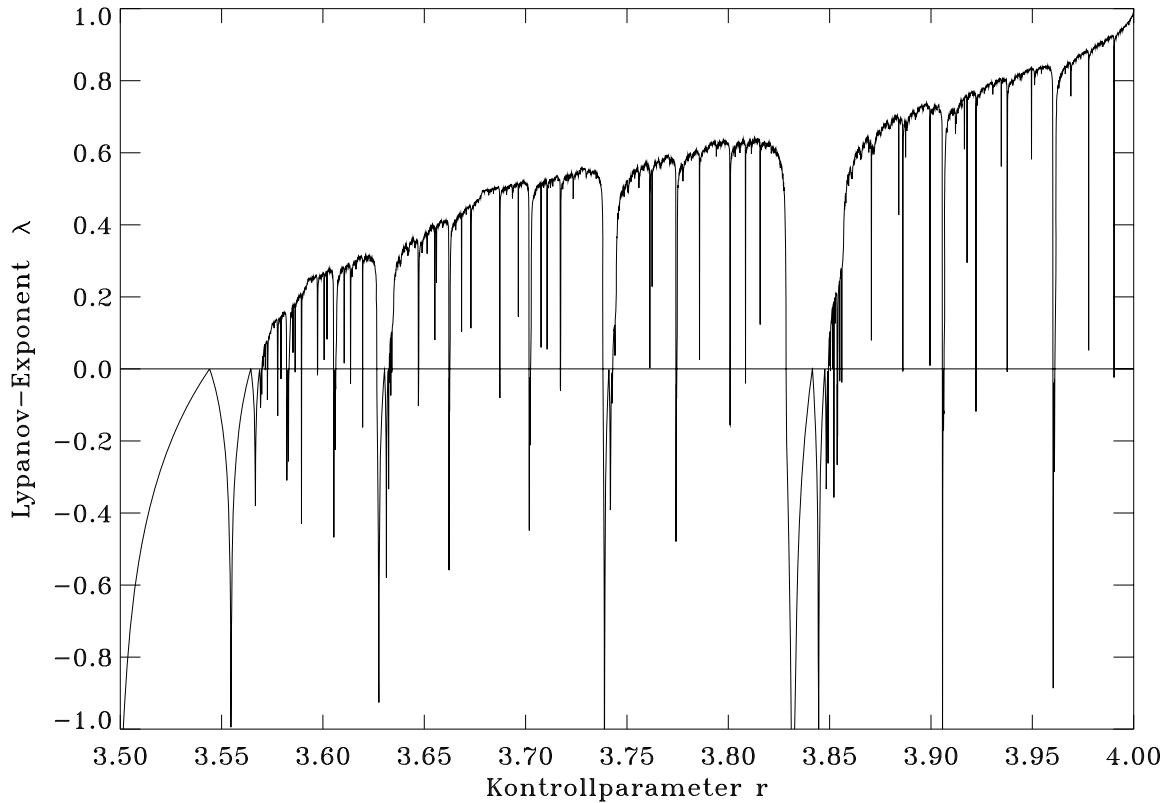
### 3.4.1 $\alpha = 2.502, 907, 875, 0 \dots$

$$\alpha = -\frac{d_n}{d_{n+1}}, \quad n \gg 1. \quad (3.13)$$

Die neue Punktgeneration sollte etwa  $\alpha$  - mal dichter liegen als die alte.

**3.4.2**  $\delta = 4.669, 201, 990, 97 \dots$

$$\delta = \lim_{n \rightarrow \infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} \quad (3.14)$$



**Abbildung 3.12:** Lyapunov-Exponenten  $\lambda(r)$  für  $r \in [3.5, 4]$ . Die Schrittweite des Kontrollparameters  $r$  ist  $10^{-4}$ .

### 3.5 Lyapunov-Exponent $\lambda$

$$\lambda(r) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'[f^i(0, r)]| \quad (3.15)$$

$$\lambda \begin{cases} > 0 & : \text{chaotische Region} \\ = 0 & : \text{Grenzzyklus} \\ < 0 & : \text{stabiler Fixpunkt} \end{cases}$$

**3.5.1**  $r \in [0, 1]$

$x_{(1)}^0 = 0$  stabil, die  $x_n$  streben gegen Null.

$$\lambda(r) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln(r) \quad (3.16)$$

$$\lambda(r) = \ln(r) \leq 0 \quad (3.17)$$

### 3. Logistische Abbildung (Parabel-Abbildung)

#### 3.5.2 $r \in (1, 3]$

Die  $x_n$  streben gegen einen festen Fixpunkt:

$$x_{(2)}^0 = 1 - \frac{1}{r} \quad (3.18)$$

$$f'_n = r \left(1 - 2 + \frac{2}{r}\right) = 2 - r \quad (3.19)$$

$$\lambda(r) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |2 - r| \quad (3.20)$$

$$\lambda(r) = \ln |2 - r| \quad (3.21)$$

#### 3.5.3 $r \in (3, 3.5699\dots)$

$$3 < r \leq 1 + \sqrt{6}$$

#### 3.5.4 $r \in (3.5699\dots, 4]$

## 3.6 Literatur

May, Robert M. 1976 Simple mathematical models with very complicated dynamics, *Nature* **261**, 459-467

Brun, Ernst 1986 Ordnungs-Hierarchien, Orell Füssli Graphische Betriebe AG, Zürich

Seifritz, Walter 1989 Wachstum, Rückkopplung und Chaos. Eine Einführung in die Welt der Nicht-linearität und des Chaos, Carl Hanser Verlag, München-Wien

Bahr, Ute & Gerhard Diener, Chaotisches Verhalten vollständig deterministischer Systeme, *Wissenschaft & Fortschritt* **39**, 209-212

Skriptensammlung zum Chaos-Seminar im Wintersemester 1990/1991 an der Technischen Universität München, Chaos-Gruppe e.V.

Jaeckel, Klaus & Joachim Schellnhuber 1986 Exemplarische Darstellung des aktuellen Paradigmenwechsels innerhalb der klassischen Mechanik, *PdN-Physik* **35**, 41-44

Thomas, Bernhard 1986 Systemtheoretische Konzepte für fächerübergreifenden Unterricht, *MNU* **39**, 464-477

Silverberg, Lars, Karl Luchner & Roman Worg 1986 Nichtlineare gekoppelte mechanische Systeme. Simulation, Experiment, stabiles und chaotisches Verhalten, *PhuD* **1**, 23-38

Luchner, Karl & Roman Worg 1987 Harmonische und chaotische Schwingungen, *MNU* **40**, 337-343

Kirchgraber, Urs 1987 Das Vorhersehbare und das Unvorhersehbare, *Naturwissenschaften* **74**, 105-110

Backhaus, U. & H.J. Schlichting 1987 Ein Karussell mit chaotischen Möglichkeiten, *PdN-Physik*



**36**, 14-22

Metzler, Wolfgang 1987 Chaos und Fraktale bei zwei gekoppelten Modelloszillationen, PdN-Physik **36**, 23-29

Glatzer, G. & R. Seydel 1986 Schwingung eines Duffing Oszillators (Unterrichtsfilm C 1532), PdN-Physik **35**, 42-45

Holden, Arun 1985 Chaos is no longer a dirty word, New Scientist 25 April, 12-15

Crutchfield, James P, J. Doyn Farmer, Norman H. Packard & Robert S. Shaw 1986 Chaos, Scientific American **254** (Heft 12), 46-57

Jung, Andreas & Klaus G. Schröder 1994 Bizarre Bilder des Herrn Poincaré, DOS Heft 4, 240-243

Lauterborn, Werner & Werner Meyer-Ilse 1986 Chaos, Physik in unserer Zeit **17**, 177-187

Feigenbaum, Mitchell J. 1980 Universal Behavior in Nonlinear Systems, Los Alamos Science, Summer 1980, 4-27

Ruelle, David 1992 Zufall und Chaos, Springer-Verlag, Berlin

Briggs, John & F. David Peat 1990 Die Entdeckung des Chaos. Eine Reise durch die Chaostheorie, DTV

Wesson, R. 1991 Die unbekannte Ordnung. Chaos, Zufall und Auslese in der Natur

Scholl, R. & O. Pfeiffer 1991 Natur als fraktale Grafik. Stochastische Fraktale und L-Systeme programmiert in Turbopascal, Markt und Technik

Faszination Fraktale (incl. Fractint, Einführung, Geschichtliches, viele Bilder) SysThema

GEO-WISSEN - CHAOS UND KREATIVITÄT, Verlag Gruner und Jahr 1990

Kuhn, Wilfried 1989 Physik, Bd. II, 1. Teil: Klasse 11, Westermann, Braunschweig, ISBN: 3-14-152131-x

Kaye, Brian H. 1994 Katastrophen im Labor, Physik in unserer Zeit, **25**, 29-35

Haken, Hermann & Arne Wunderlich 1991 Die Strukturierung der Materie. Synergetik in der un- belebten Welt, Vieweg, Braunschweig

Baker, G. L. & J. P. Gollub 1990 Chaotic Dynamic - An Introduction, Cambridge UP

Falconer, Kenneth T. 1993 Fraktale Geometrie, Spektrum, Heidelberg, ISBN: 3-86025-075-2

Frøyland Jan 1992 Introduction to Chaos and Coherence, Institute of Physics Publishing, Bristol

Backhaus, U. & H.-Joachim Schlichting 1992 Was hat Schönheit mit Chaos zu tun? Vergleich der

### *3. Logistische Abbildung (Parabel-Abbildung)*

Mandelbrot-Iteration mit einem chaotischen Drehpendel, Preprint

Loistl, Otto & Iro Betz 1993<sup>2</sup> Chaostheorie. Zur Theorie nichtlinearer dynamischer Systeme, Oldenbourg, ISBN: 3-486-22756-4

Peak, D. & M. Frame 1994 Chaos under Control, Freeman, New York

Zeitler, H. & W. Neidhardt 1994 Fraktale und Chaos, Wissenschaftliche Buchgesellschaft, Darmstadt

# 4. Lorenz - Gleichungen

Obgleich die Lorenz-Gleichungen ein sehr oft zitiertes Beispiel für ein chaotisches dynamisches System darstellen, sind die Eigenschaften des Lorenz-Attraktors gänzlich andere als diejenige, die das Ausgangsgleichungssystem hervorbringt. Die Ursache liegt in der unzulässig verkürzten Galerkin-Approximation bei der Herleitung der Lorenz-Gleichungen aus den ursprünglich partiellen Differentialgleichungen. Dennoch haben die Lorenz-Gleichungen nicht nur zur Veranschaulichung, sondern auch zum Verständnis des Phänomens der nichtlinearen Dynamik beigetragen. Die Suche nach Reduktionen von hoch- oder unendlich-dimensionalen Systemen, die den gesamten Attraktor und damit das Langzeitverhalten widerspiegeln, ist Gegenstand der mathematischen Forschung.

## 4.1 Rayleigh - Bénard - Konvektion

Bénard (1900) Experiment

Rayleigh (1916) Theorie

- von unten erhitzte Flüssigkeiten (Konvektionszellen)
- Atmosphären der Planeten, Materiebewegung in den Konvektionszonen von Sternen und Akkretionsscheiben

### 4.1.1 Navier - Stokes - Gleichung

Navier (1785-1836)

Stokes (1819-1903)

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v} \cdot \vec{\nabla})\vec{v} = F - \vec{\nabla} p + \eta \Delta \vec{v} \quad (4.1)$$

Skalenfreie Form:

$$\frac{\partial \vec{v}'}{\partial t'} + (\vec{v}' \cdot \vec{\nabla}')\vec{v}' = F' - \vec{\nabla}' p' + \frac{1}{R} \Delta' \vec{v}' \quad (4.2)$$

Reynold'sche Zahl (1842-1912):

$$R = \frac{\rho L V}{\eta} \quad (4.3)$$

Boussinesq Näherung: Nur die Dichte ist von der Temperatur abhängig:

$$\rho = \rho_0(1 - \alpha \Delta T) \quad (4.4)$$

### 4.1.2 Kontinuitätsgleichung

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (4.5)$$

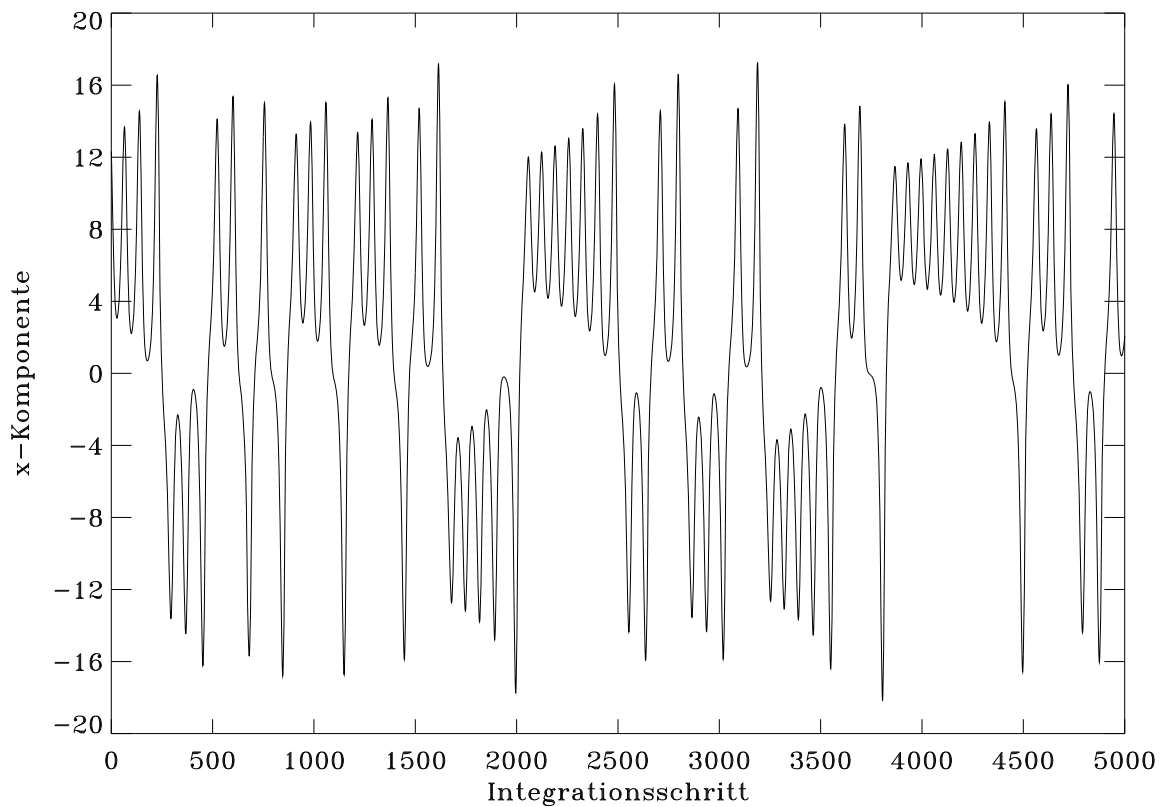
#### 4. Lorenz - Gleichungen

##### 4.1.3 Fouriersche Wärmeleitungsgleichung (1822)

$$\frac{\partial T}{\partial t} = \chi \nabla^2 T \quad (4.6)$$

##### 4.1.4 Saltzman - Gleichungen (1962)

##### 4.1.5 Lorenz - Gleichungen



**Abbildung 4.1: Die x-Komponente des Lorenz-Attraktors.**

Für die Saltzman-Gleichungen des Rayleigh-Bénard-Experiments wird in niedrigster Ordnung ein Fourier-Modenansatz gewählt  $\rightarrow$  Lorenz-Gleichungen (nichtlineare Terme!):

$$\dot{x} = \sigma(y - x), \quad (4.7)$$

$$\dot{y} = x(r - z) - y \quad \text{und} \quad (4.8)$$

$$\dot{z} = xy - bz, \quad (4.9)$$

mit

$$\sigma = 10, \quad r = 28, \quad b = \frac{8}{3} \quad (4.10)$$

$x$  = dimensionslose Geschwindigkeitsamplitude, Rate des konvektiven Overturning  
 $y, z$  = dimensionslose Amplituden der horizontalen und vertikalen Temperaturvariation  
 $\sigma$  ist proportional zur Prandtl-Zahl  $Pr$

$$Pr = \frac{\nu}{\chi} \quad (4.11)$$

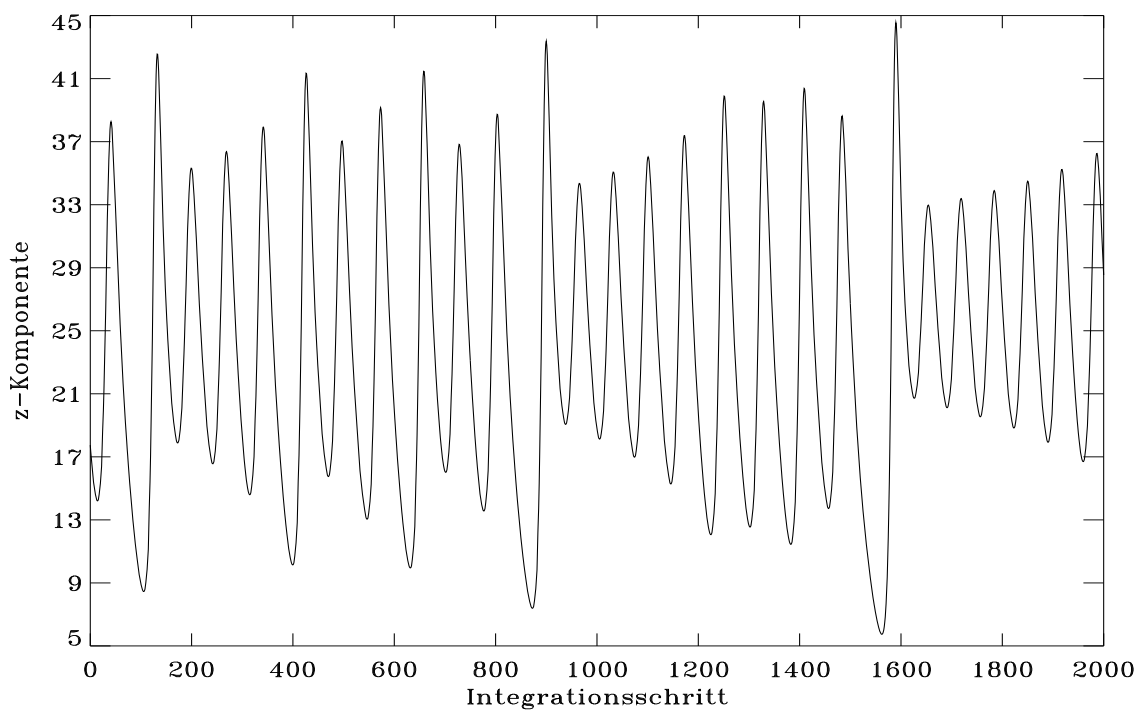


Abbildung 4.2: Die z-Komponente des Lorenz-Attraktors.

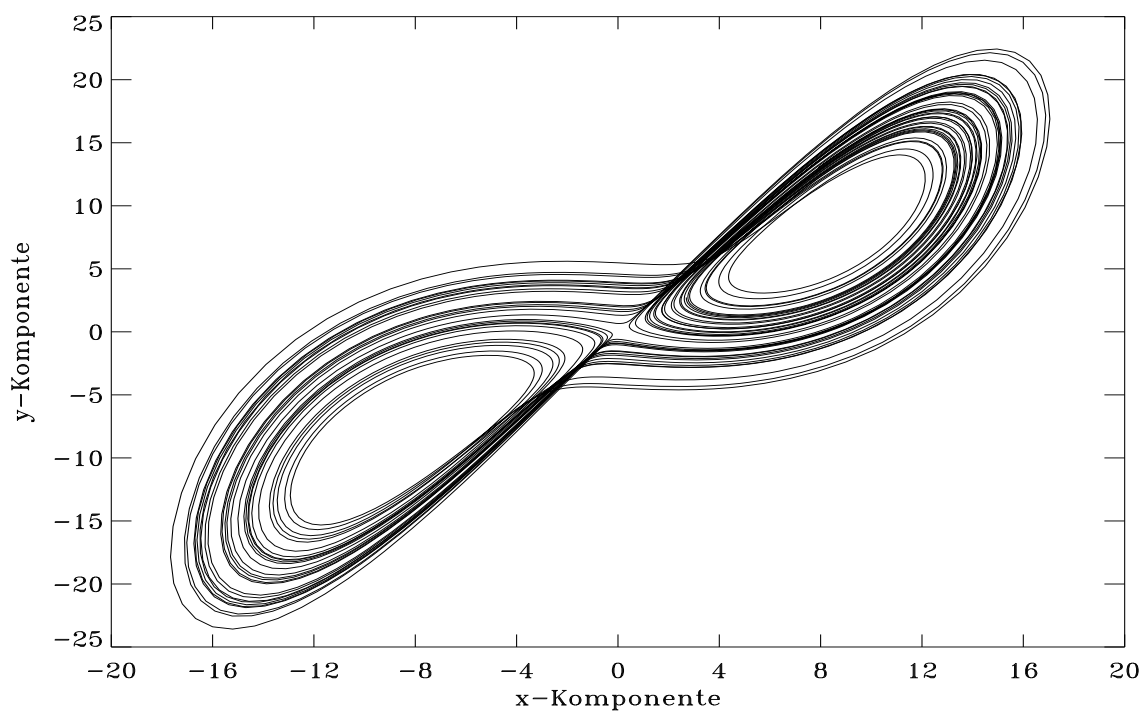
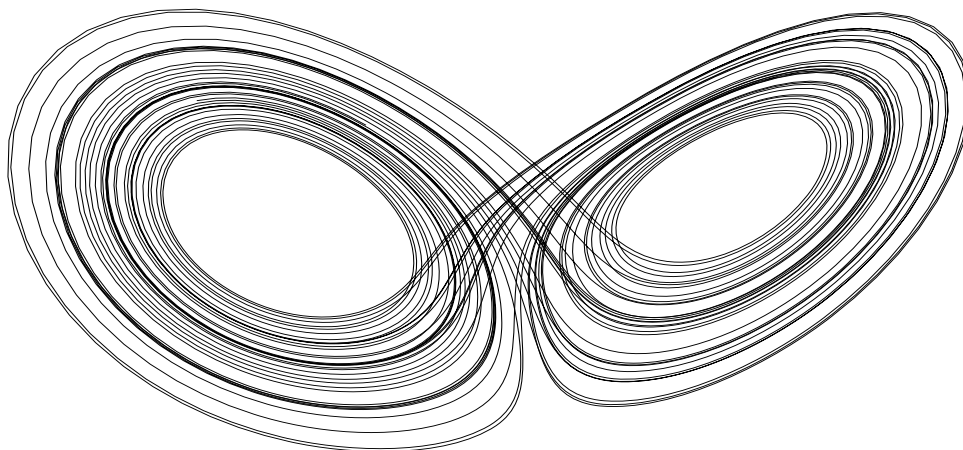
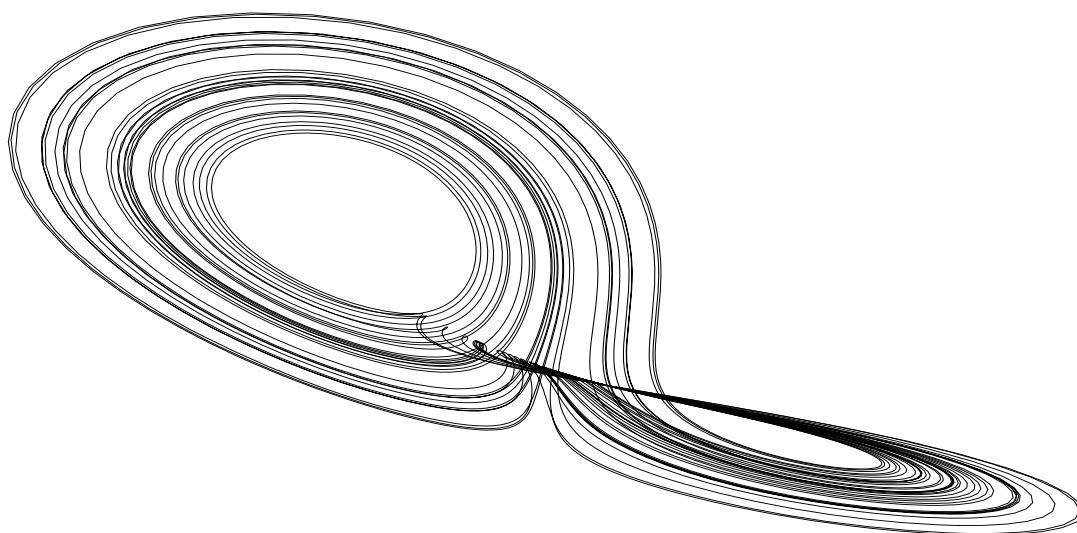


Abbildung 4.3: Die x-y-Projektion des Lorenz-Attraktors.

4. Lorenz - Gleichungen



**Abbildung 4.4:** Eine 3D-Darstellung des Lorenz-Attraktors.



**Abbildung 4.5:** Eine 3D-Darstellung des Lorenz-Attraktors.

$\nu$  = kinematische Zähigkeit

$\chi$  = thermische Diffusivität

$r$  = normierte Rayleigh-Zahl, Kontrollparameter

$$r = \frac{\text{Ra}}{\text{Ra}_c} \propto \Delta T \quad (4.12)$$

$a$  = Wellenzahl der horizontalen Fundamentalmode

$$b = \frac{4}{1 + a^2} \quad (4.13)$$

Normierte Zeit:

$$\tau = \frac{\pi^2}{h^2} (1 + a^2) \chi t \quad (4.14)$$

- 1D-Projektionen des Lorenz-Attraktor
- 1D-Projektionen von reduzierte Gleichungen für Sonnendynamo

Die Bifurkationsstruktur der ursprünglichen Gleichungen und der stark reduzierter Gleichungen kann grundlegend verschieden sein. Man beachte, daß z.B. die 2D Rayleigh-Benard-Konvektion kein Chaos zeigt, also nicht durch die Lorenz-Gleichungen beschrieben wird.

## Euler – Verfahren

Anwendung auf die Gleichungen für einen gedämpften harmonischen Oszillator:

$$\dot{x} = -cx - \omega^2 y, \quad (4.15)$$

$$\dot{y} = x \quad (4.16)$$

Abb. 15. Gedämpfte harmonische Schwingung  $x(t)$

Abb. 16. Phasenraumdarstellung für einen gedämpften harmonischen Oszillator

Die Gleichungen für eine Integration des obigen Differentialgleichungssystems mittels Euler-Verfahren lauten:

$$x(t + \Delta t) = x(t) - (cx(t) + \omega^2 y(t)) \Delta t \quad (4.17)$$

$$y(t + \Delta t) = y(t) + x(t) \Delta t \quad (4.18)$$

Als Parameter und Anfangsbedingungen eignen sich z.B. folgende Werte:

$x = 5$ ,  $y = 0$ ,  $\Delta t = 0.01$ , Reibungskonstante  $= c = 0.01$ , Winkelgeschwindigkeit  $= \omega = 1$ .

Die Anwendung desselben Lösungsalgorithmus auf die Lorenz - Gleichungen führt auf die folgenden Gleichungen:

$$x(t + \Delta t) = x(t) + \sigma(y(t) - x(t)) \Delta t, \quad (4.19)$$

$$y(t + \Delta t) = y(t) + (x(t)(r - z(t)) - y(t)) \Delta t, \quad (4.20)$$

$$z(t + \Delta t) = z(t) + (x(t)y(t) - bz(t)) \Delta t, \quad (4.21)$$

wobei wir die Standardparameter verwenden:

$$\sigma = 10, r = 28, b = \frac{8}{3}. \quad (4.22)$$

Benachbarte Trajektorien:

$$\vec{d}(\vec{r}, \vec{r}') \quad (4.23)$$

Beobachtung von Lorenz (1963):

#### 4. Lorenz - Gleichungen

- Unabhängig von den Anfangsbedingungen und unabhängig von der Integrationsroutine (Euler - Methode erster Ordnung reicht aus!) gibt es einen Attraktor mit einer charakteristischen Form
- Die Trajektorie ist weder periodisch noch stationär
- Die Details des Attraktors hängen sehr empfindlich von den Anfangsbedingungen und der Integrationsmethode ab. Keine Voraussagbarkeit (Loop - Sequenzen).

Numerische Experimente:

- $x(t), y(t), z(t)$
- Phasenraumplot
- Zwei parallele Orbits mit Verbindungslinie
- Simple Schätzung des Lyapunov-Exponenten

$$\lambda(r) = \frac{1}{n\Delta t} \sum_{i=1}^n \ln \left| \frac{\Delta x_i}{\Delta x_1} \right| \quad (4.24)$$

Lorenzsches Wasserrad

## 4.2 Langzeitvariabilität der Sonnenaktivität

### 4.2.1 Beobachtungen

Die Vorgänge innerhalb der Sonne, auf ihrer Oberfläche und in ihrer näheren Umgebung sind Beispiele der solaren Aktivität, die durch die Wechselwirkung der Bewegung geladener Teilchen (vorwiegend Wasserstoffionen und Elektronen) und von Magnetfeldern bestimmt werden. Bei aktiven Phänomenen, wie Sonnenflecken und Flares, kann die Veränderung des Magnetfeldes unmittelbar beobachtet werden. Sonnenflecken sind, im Visuellen, dunklere (kühlere 5000 K) Gebiete in der Photosphäre, die durch starke Magnetfelder ausgezeichnet sind (Zeeman-Effekt). Oberhalb der Fleckenregionen befinden sich Quellen für Radio- und Röntgenstrahlung. Die explosive Umwandlung von magnetischer Energie in Radio- und Röntgenstrahlung wird als Flare bezeichnet.

Daten	Seit
Tägliche Sonnenfleckenrelativzahlen (Wolf-Zahlen)	1750
Polarlichter ( $\Phi > 62^\circ$ )	1780
Magnetische Aktivität	1868
Aktivitätszentren der Chromosphäre	1911
Kritische Frequenz der Ionosphäre	1931
Kosmische Strahlung: Harte Komponente	1937
Kosmische Strahlung: Nukleare Komponente ( $^{14}\text{C}$ )	1957
10 cm Strahlung	1947
Chromosphärische Eruptionen und Radioausbrüche	1957
Magnetische Aktivität (Dst, AE)	1957
Absorption der galaktischen Radiostrahlung (Riometrie)	1958

**Tabelle 4.1: Beobachtungsreihen zur Sonnenaktivität**



### 4.2.2 Polarlichter

Seit dem Altertum werden Polarlichter, die ebenfalls eine Folge der Sonnenaktivität darstellen, beobachtet. Der Begriff *aurora borealis* ist erst 1619 erwähnt worden. Das Polarlichtoval, die bevorzugte Region dieser Erscheinung, findet man zwischen  $67^\circ$  und  $70^\circ$  nördlicher bzw. südlicher Breite.

Folgende Bestandteile der mittleren und unteren Ionosphäre der Erde werden dabei von intensiven energetischen Teilchenschauern (Elektronenprecipitation:  $E_{kin} \approx 100eV \dots 10keV$ ) bevorzugt angeregt.

Atomar Sauerstoff in ca. 110 km Höhe:

- $\lambda(2^1S_0(\text{metastabil}) \rightarrow 2^1D_2) = 557.7 \text{ nm}$  (gelblich grün, verbotene Linie, nur bei sehr geringem Druck beobachtbar),  $\tau_w = 0.7 \text{ s}$
- $\lambda(2^1D_2 \rightarrow 2^3P_2) = 630 \text{ nm}$  (rot),  $\tau_w = 100 \text{ s}$

Molekularer Stickstoff:  $N_2, N_2^+$  (blau, violett)

Atomarer Wasserstoff in ca. 250 km Höhe:

- $\lambda(H\alpha) = 656.3 \text{ nm}$  (rot)
- $\lambda(H\beta) = 486.1 \text{ nm}$  (blau)

Folgende Vorgänge spielen dabei eine wesentliche Rolle:

Ionisation



Umladung



Dissoziative Rekombination



### 4.2.3 Sonnenflecken

#### Beobachtungen mit bloßem Auge

Abb. 18 Sonnenfleckenbeobachtungen mit bloßem Auge gemäß historischer Quellen

354 v.Chr.

807

1607 Johannes Kepler, 28. Mai

#### Fernrohrbeobachtung ab 1608

1610 Galileo Galilei

Thomas Harriot, 8. Dezember

1611 Christoph Scheiner und Johann Baptist Cysat, 6. März

Johannes Fabricius, 9. März

4. Lorenz - Gleichungen

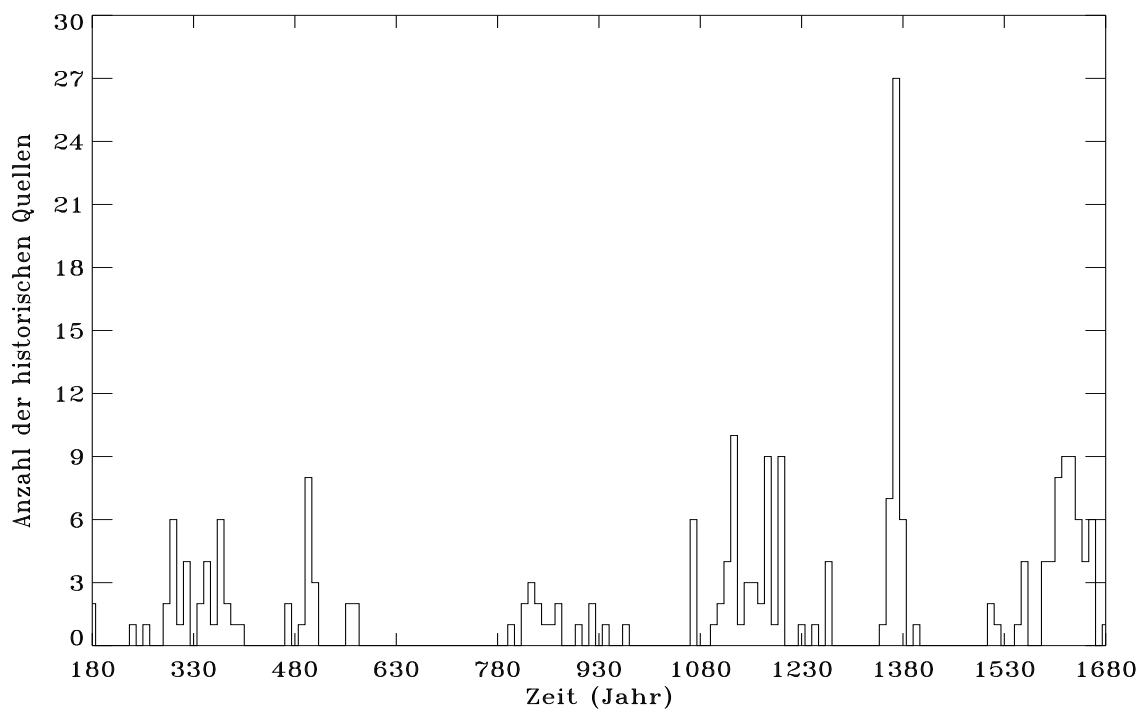


Abbildung 4.6: Die Jahresmittelwerte der Sonnenflecken-Relativzahlen.

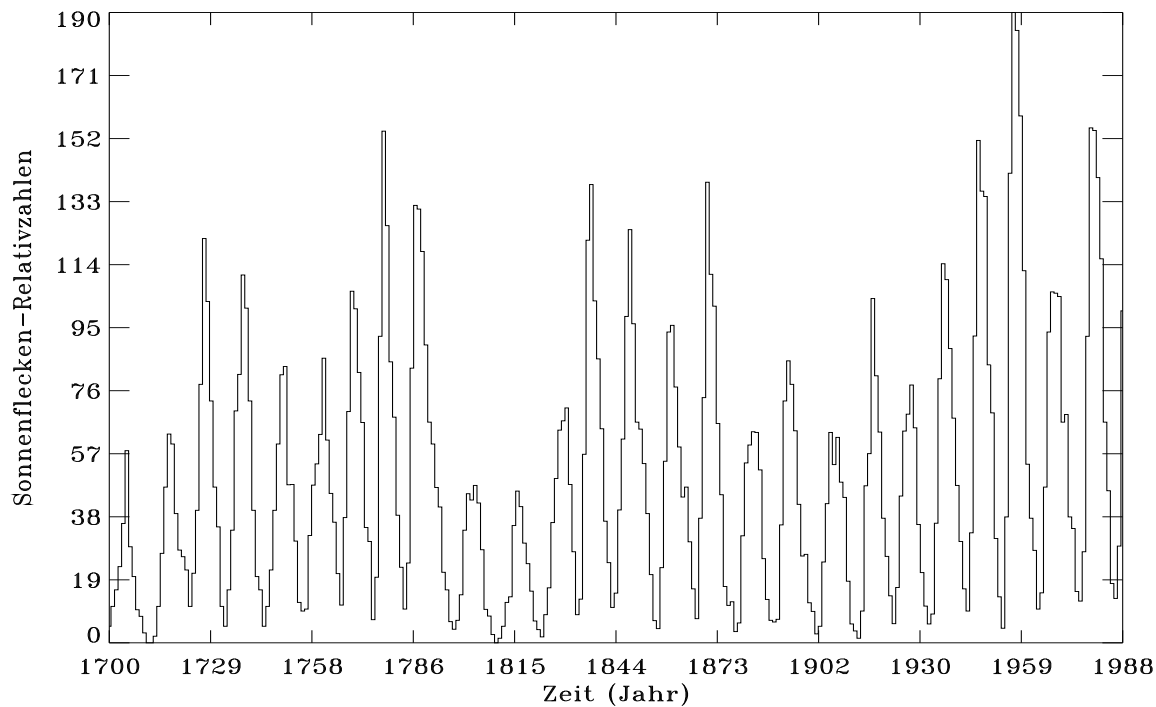


Abbildung 4.7: Die Jahresmittelwerte der Sonnenflecken-Relativzahlen (National Geophysical Data Center, Boulder).

**Systematische Sonnenbeobachtung**

ab 1749 Monatsmittel

ab 1849 Tagesmittel

Jagd nach "Vulkan"<sup>1</sup> ab 1826 durch Heinrich Samuel Schwabe (1789-1875)

Astronomische Nachrichten **21** (1844) Nr. 495, 233-236, Mitteilung vom 31. Dezember 1843: Periode beträgt 10 Jahre

Alexander von Humboldt (1850) Kosmos, 3. Band.

Täglich mindestens ein Fleck:

1828, 1829, 1836-1839, 1847-1849

100 fleckenfreie Tage im Jahr:

um 1833 und 1843.

Bereits 1843 entdeckte der Apotheker Heinrich Samuel Schwabe (1789-1875) aus Dessau, daß die Sonnenfleckenzahl eine Zyklusdauer von etwa 10 Jahren aufweist. Um die Sonnenaktivität auf eine einfache Weise zu beschreiben, führte Rudolf Wolf (1816-1893) aus Zürich im Jahre 1830 Relativzahlen ein, die durch Zählen von Einzelflecken und Fleckengruppen bestimmt wird. Diese Zahl wird als ein geeigneter Indikator für die Entwicklung des Magnetfeldes der Sonne angesehen.

Die Zyklusdauer der Sonne beträgt etwa 11 Jahre. Das Maximum der Sonnenfleckenzahlen von 1760 beschreibt den ersten Zyklus; 1986 endete der 21. Zyklus. Die Amplitude der Sonnenfleckenzahlen variiert aperiodisch und weist Intervalle extrem niedriger Sonnenaktivität auf.

Die Sonnenaktivität dokumentiert sich außerdem in der:

- <sup>10</sup>Be- und  $\delta^{18}\text{O}$ -Häufigkeit in Eiskernen und der Thermolumineszenz von Sedimenten
- 10 cm Strahlung
- Flare-Häufigkeit.

**4.2.4 Erdmagnetische Störungen**

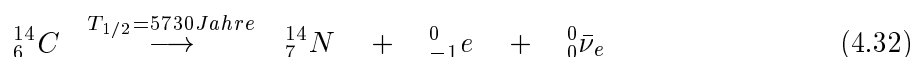
1770 ff. A.v. Humboldt, Ritter, Gauß

Ritter 1803: Periode 9 ... 10 Jahre

Polarlichter und magnetische Gewitter sind korreliert.

**4.2.5  $\Delta^{14}\text{C}$  Häufigkeit**

Ein weiterer Indikator für die Sonnenaktivität ist der durch Assimilation von  $\text{CO}_2$  in den fossilen Bäumen befindliche radioaktive Kohlenstoff, der auch zur Altersbestimmung herangezogen wird. Die heliomagnetische Modulation des Flusses der galaktischen kosmischen Strahlung in der Erdumgebung bedingt eine Variation des  $\Delta^{14}\text{C}$  – Gehaltes der Erdatmosphäre.



Durch die akribische Analyse der Jahresringe fossiler Bäume (C.P. Sonett, University of Arizona) steht uns ein  $\Delta^{14}\text{C}$  – Datensatz zur Verfügung, der etwa 9000 Jahre umfaßt. Typische Zeitskalen des  $\Delta^{14}\text{C}$ -Datensatzes sind:

<sup>1</sup>Man vermutete innerhalb der Merkur-Bahn einen weiteren Planeten

#### 4. Lorenz - Gleichungen

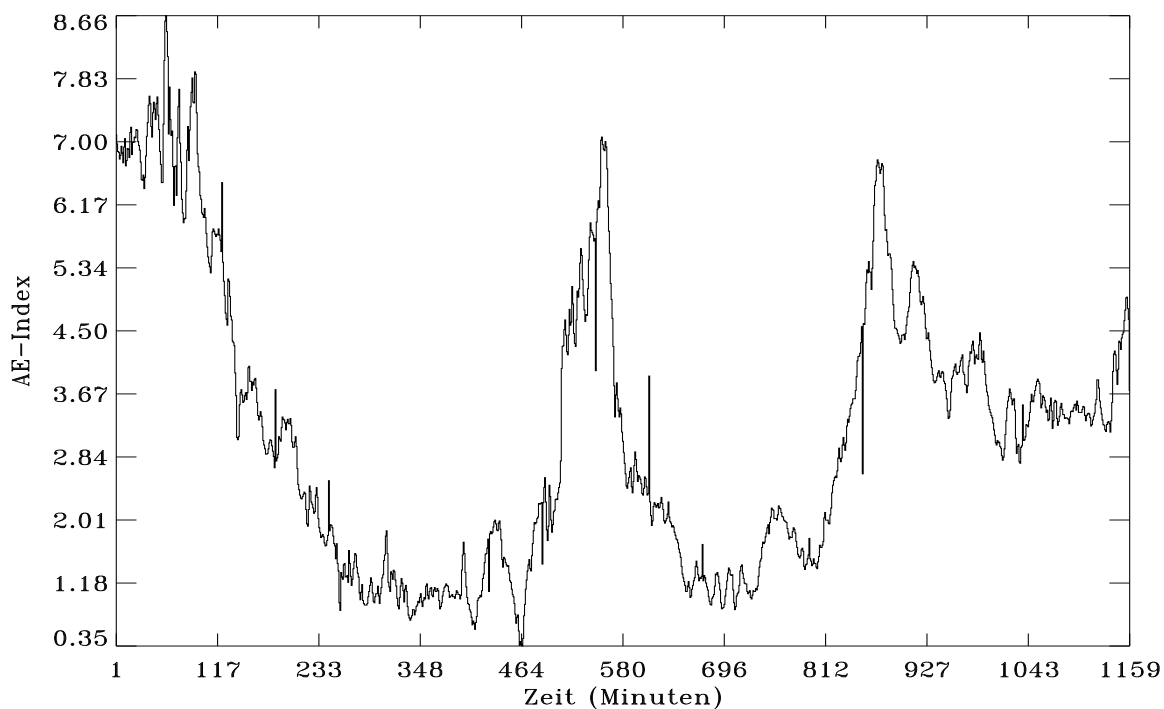


Abbildung 4.8: Minutenwerte des AE-Index am 1. Juli 1981 in Kyoto. Der AE-Index beschreibt die geomagnetischen Störungen, die durch den auroralen Elektrojet hervorgebracht werden. Der aurale Elektrojet reflektiert die Wechselwirkung des Sonnenwindes mit der Erdmagnetosphäre.

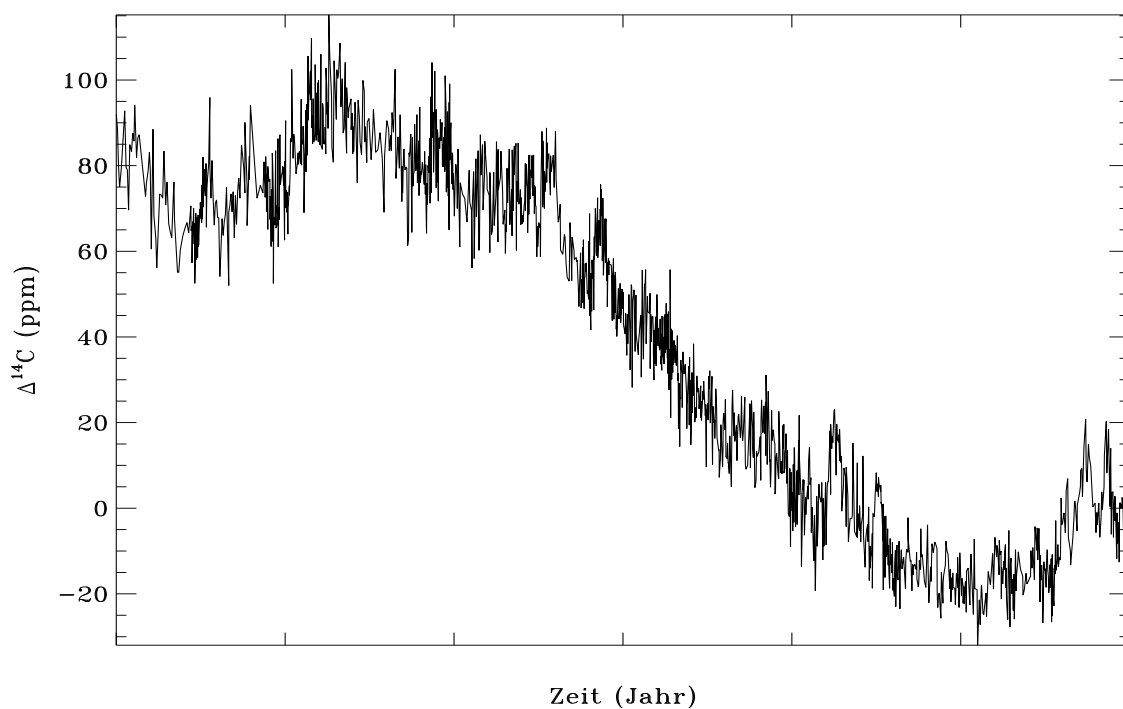


Abbildung 4.9: Die  $\Delta^{14}\text{C}$  Häufigkeit der letzten 9000 Jahre.

- 10000 Jahre, geomagnetische Modulation (Geodynamo)
- 2000 Jahre, ozeanische CO<sub>2</sub> - Zirkulation
- 210 Jahre solaren Ursprungs
- 80 - 90 Jahre Gleissberg - Zyklus (solar)
- 11 Jahre
- Response Zeit = 30 Jahre

Die Analyse dieses Datensatzes mit Methoden der nichtlinearen Dynamik sollte einen tieferen Einblick in die Langzeitvariation der Sonnenaktivität gestatten.

#### 4.2.6 Große Minima

Name des Minimums	Zeitraum
Oort	1010 bis 1050
Wolf	1282 bis 1342
Spörer	1416 bis 1534
Maunder	1654 bis 1714
Dalton	1795 bis 1830

**Tabelle 4.2: Große Minima der Sonnenaktivität**

Berücksichtigt man alle bekannten Phänomene der solaren Aktivität, so lassen sich für folgende Zeitintervalle Große Minima der Sonnenaktivität feststellen.

Hermann Fritz (1872): Zusammenhang von Polarlichtern und Sonnenflecken.

Positive Korrelation zeigt die Cholerasterblichkeit bis 1880.

1638-1715 kaum Sonnenflecken (Sonnenkönig)

#### 4.2.7 Dynamo - Modell für die Sonnenaktivität

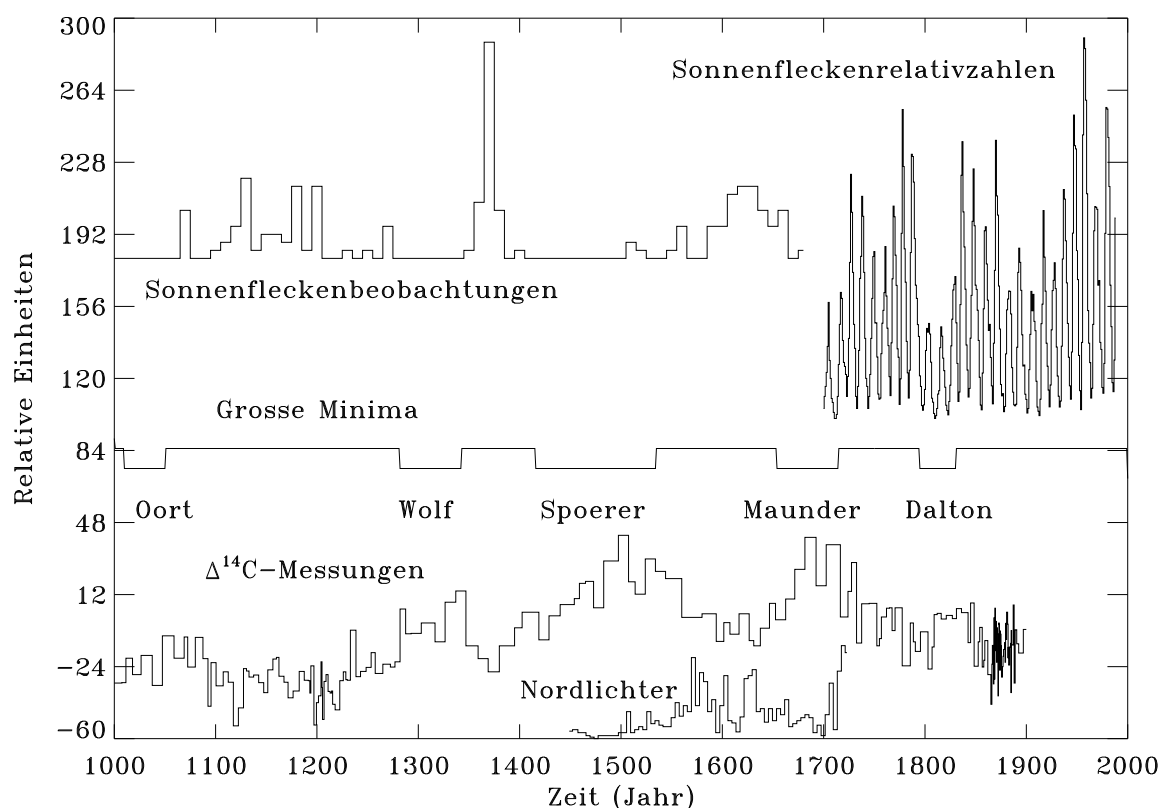
Die globale Sonnenaktivität kann mit der Dynamotheorie erklärt werden. Die Grundidee der Dynamotheorie ist, daß Magnetfeld der Sonne durch die differentielle Rotation und helikale Turbulenz verstärkt und aufrechterhalten wird. Die helikalen Bewegungen werden durch Wirkung der Coriolis-Kraft auf die ungleichförmigen konvektiven Bewegungen hervorgebracht. Daher sind die magneto-hydrodynamischen Gleichungen der theoretische Hintergrund für die theoretische Beschreibung der Vorgänge in der Konvektionszone. Die Änderung des Magnetfeldes  $\mathbf{B}$  wird durch die Induktionsgleichung beschrieben:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \eta \nabla^2 \mathbf{B} \quad (4.33)$$

( $\eta$  ist die magnetische Diffusivität). Die Änderung der Fluidgeschwindigkeit  $\mathbf{u}$  ist in erster Linie durch die Gravitationsfeldstärke  $\rho \mathbf{g}$ , den Druckgradienten  $\nabla p$ , die Lorentz-Kraft  $\mathbf{j} \times \mathbf{B}$  und die viskose Kraft  $\nabla \cdot \boldsymbol{\tau}$  bedingt. Die Grundgleichung, die diese Kräfte verknüpft, ist die Navier-Stokes-Gleichung

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = \rho \mathbf{g} - \nabla p + \mathbf{j} \times \mathbf{B} + \nabla \cdot \boldsymbol{\tau} \quad (4.34)$$

( $\mathbf{j}$  – elektrische Stromdichte,  $\rho$  – Fluidichte,  $\boldsymbol{\tau}$  – Viskositätstensor,  $\mathbf{g}$  – Gravitationsfeldstärke,  $p$  – Druck). Man kann nun zwei Wege verfolgen, um die Dynamik des Magnetfeldes und der Fluidbewegung untersuchen.



**Abbildung 4.10:** Sonnenflecken sind dem Menschen seit langem bekannte Naturerscheinungen. Doch erst seit unserem Jahrhundert ist bekannt, daß diese und andere Phänomene (z.B. Polarlichter in mittleren Breiten und die magnetische Aktivität der Erde) Ausdruck der Aktivität der scheinbar gleichmäßig leuchtenden Sonne sind.

Dargestellt sind Beobachtungen zur Sonnenaktivität seit dem Jahre 1000 n. Chr. Der Graph oben links ist den Angaben von Wittmann und Xu (A Catalogue of Sunspot Observations from 165 BC to AD 1684, Astron. Astrophys. Suppl. Ser. 70 (1987) 83-94) entnommen und spiegelt die Anzahl historischer Quellen aus dem asiatischen Raum wider, die von Sonnenfleckenbeobachtungen mit bloßem Auge berichten. Die Darstellung rechts daneben stellt die Jahresmittelwerte für die Sonnenfleckenrelativzahlen dar (C. Abston, L. Morris & J. Allen: CD-ROM NGDC01, National Geophysical Data Center, Boulder). Die Rechteckkurve darunter deutet die Zeiträume der großen Minima an (Kleine Eiszeit: 1550 bis 1850). Darunter sind die Resultate der  $\Delta^{14}\text{C}$ -Messungen (Laboratorien in La Jolla, Belfast und Stuttgart) dargestellt. Sie sind ein weiteres Maß für die Sonnenaktivität ist die Häufigkeit des durch Bäume aufgenommenen radioaktiven Kohlenstoffs  $^{14}\text{C}$ . Die  $^{14}\text{C}$ -Produktion hängt von der Sonnenaktivität ab. Durch die Radiokarbon-Analyse der Jahresringe fossiler Bäume ist die Abweichung von der mittleren  $^{14}\text{C}$ -Produktion  $\Delta^{14}\text{C}$  gemessen worden.

Auffallend ist die Antikorrelation zwischen der Anzahl der beobachteten Sonnenflecken und den Meßwerten für den  $^{14}\text{C}$ -Überschuß. Ganz unten ist für ein ausgewähltes Zeitintervall die Anzahl historischer Quellen zur Nordlichtbeobachtung dargestellt (W. Schröder).

- Zur Beschreibung der großskaligen Variationen der Sonnenaktivität werden die Gleichungen für das mittlere Magnetfeld untersucht.
- Die Gleichungen 4.33 und 4.34 werden gelöst und lokale Effekte in der Konvektionszone betrachtet.

Der mittlere Magnetfeldtransport wird durch Turbulenz-Transport-Koeffizienten, wie durch das  $\alpha$  ( $\alpha$ -Effekt) und Turbulenzdiffusivität bestimmt.

Folgende Fakten stützen die Rolle der Turbulenz:

1. Man beobachtet an der Sonnenoberfläche eine differentielle Rotation, d.h., daß die äquatornahen Bereiche der Sonnenoberfläche den polnahen vorausziehen.
2. Die Sonne (Stern der unteren Hauptreihe!) hat eine konvektive Hülle, die fast 30 % des Sonnenradius umfaßt (Modellrechnungen).

Die Coriolis-Kraft bringt im Mittel auf der Nord(Süd)-Halbkugel der Sonne eine links(rechts)drehende helikale Bewegung hervor. Dies wird durch den Zusatzterm in der Induktionsgleichung für die gemittelten Felder (4.33) berücksichtigt, der den  $\alpha$ -Effekt beschreibt. Die Folge des  $\alpha$ -Effekts ist ein modifiziertes Ohmsches Gesetz für die gemittelten Größen, bei dem zusätzlich zur elektromotorischen Kraft durch die großskalige Geschwindigkeit  $\mathbf{v} \times \mathbf{B}$  eine durch kleinskalige Turbulenz bedingte elektromotorische Kraft  $\alpha \mathbf{B}$  auftritt.

Die obigen Gleichungen haben periodische Lösungen, die regulären Dynamowellen entsprechen und eine Grundlage für die Erklärung des periodischen Sonnenzyklus liefern. Die komplizierte Amplituden- und Frequenzmodulation, die aus Abb. 4.7 hervorgeht, erfordert die Berücksichtigung stochastischer oder nichtlinearer Effekte.

#### 4.2.8 Malinetzky-Ruzmaikin Modell

Nichtlineares Modell des hydrodynamischen Dynamos mit wenigen Moden:

- 7 autonome ODE's mit den 7 Parametern ( $D, \sigma, \alpha, \nu_0, \nu, p, q$ ) für die ersten Moden:
  - $D$  Dynamozahl (differentielle Rotation  $\times$  mittlere Helizität / Quadrat der turbulenten Diffusivität)
  - $q$  Rückkopplung von magnetischer Helizität und Magnetfeldstärke
- Einfluß des magnetischen Feldes auf die mittlere Helizität des Magnetfeldes (Spiralität der Strömung)

$$\dot{a}_1 = -\sigma a_1 + (\alpha + c_0)b_1 + \frac{1}{2}(c_1 b_1 + c_2 b_2) \quad (4.35)$$

$$\dot{a}_2 = -\sigma a_2 + (\alpha + c_0)b_2 - \frac{1}{2}(c_1 b_1 - c_2 b_2) \quad (4.36)$$

$$\dot{b}_1 = -b_1 - D a_2 \quad (4.37)$$

$$\dot{b}_2 = -b_2 + D a_1 \quad (4.38)$$

$$\dot{c}_0 = -\nu_0 c_0 + p(a_1 b_1 + a_2 b_2) - q[(\alpha + c_0)(b_1^2 + b_2^2) + \frac{1}{2}c_1(b_1^2 - b_2^2) + c_2 b_1 b_2] \quad (4.39)$$

$$\dot{c}_1 = -\nu c_1 + p(a_1 b_1 - a_2 b_2) - q[(\alpha + c_0)(b_1^2 - b_2^2) + c_1(b_1^2 + b_2^2)] \quad (4.40)$$

$$\dot{c}_2 = -\nu c_2 + p(a_1 b_2 + a_2 b_1) - q[(\alpha + c_0)2b_1 b_2 + c_2(b_1^2 + b_2^2)] \quad (4.41)$$

### 4.3 Literatur

W. Ebeling, Chaos, Ordnung und Information, Urania - Verlag, Leipzig-Jena-Berlin, 1989, pp. 25-35.  
Verlag Harri Deutsch 1991

Vongxaya, B., Ordnung und Chaos bei nichtlinearen Schwingungen, Verlag Harri Deutsch, Thun 1995

Lorenz, Edward N., Deterministic Nonperiodic Flow, J.Atmos.Sciences **20** (1963) 130-141

Cvitanović, Predrag (ed.), Universality in Chaos, Adam Hilger Ltd., Bristol 1984, 367-378

Bai-Lin, Hao (ed.), Chaos, World Scientific, Singapore 1984, 282-293

Schuster, Heinz Georg, Deterministic Chaos: an Introduction, VCH, Weinheim 1988<sup>2</sup>

Argyris, J., G. Faust & M. Haase (1994) Die Erforschung des Chaos, Vieweg, Braunschweig

C. Sparrow, The Lorenz Equation: Bifurcations, Chaos, and Strange Attractors, Springer, Berlin, 1982

P.V. Foukal, Die veränderliche Sonne, Spektrum der Wissenschaft, April 1990, 66-75

G. Cini-Castagnoli (ed.): Solar - Terrestrial Relationships and the Earth Environment in the Last Millennia. Proceedings of the International School of Physics Enrico Fermi; Course XCV, Varenna, Italy, 1985. North - Holland Physics Publ., 1988

Kippenhahn, Rudolf (1990) Der Stern, von dem wir leben. Den Geheimnissen der Sonne auf der Spur, Deutsche Verlags-Anstalt, Stuttgart

Schröder, Wilfried (1984) Das Phänomen des Polarlichts. Geschichtsschreibung, Forschungsergebnisse und Probleme, Wissenschaftliche Buchgemeinschaft, Darmstadt, S.38-46

Bührke (1994) Thomas Das Maunder-Minimum. Alles sonnenklar, oder? Sterne und Weltraum Januar-Heft, 5-6



# 5. Fraktale & Selbstähnlichkeit

## 5.1 Formenvielfalt in der Natur

Statistische Selbstähnlichkeit

- Fraktales Wachstum: Diffusion Limited Aggregation (Witten-Sander Modell). Straßennetz, Blutbahnen, Nervenbahnen, Schneeflocken, Kristallisation in übersättigten Lösungen
- Brownsche Bewegung (J. Perrin, Reif, Gardiner)

(Perrin 1913 Selbstähnlichkeit, Mandelbrot 1977 Fraktal)

## 5.2 Mathematische Monster

Iterierte geometrisch-kombinatorische Prozesse (Skalenverhalten, aber verschieden von den Fraktalen der Natur (nicht mehr als 3 Größenordnungen!) und den Lösungen dynamischer Gleichungen)

### 5.2.1 Kapazität

Hausdorff 1919

Kolmogorov 1958 Kapazität einer geometrischen Figur

$n$  = Iterationsschritte

$l$  = Länge eines Basiselements

$N$  = Kleinste Anzahl von geometrischen Basiselementen zur vollständigen Überdeckung mit Basiselementen der Länge  $l$

$$N(l) \sim l^D \quad (5.1)$$

$$D = \lim_{l \rightarrow 0} \frac{\ln N(l)}{\ln(\frac{1}{l})} \quad (5.2)$$

### 5.2.2 Kochsche Schneeflocke

$$l(n) = \left(\frac{1}{3}\right)^n \quad (5.3)$$

$$N(n) = 4^n \quad (5.4)$$

$$D = \lim_{n \rightarrow \infty} \frac{n \ln 4}{n \ln 3} \quad (5.5)$$

$D(\text{Kochkurve}) = \ln 4 / \ln 3 \approx 1.2619 \dots$

### 5.2.3 Cantor-Staub

Georg Cantor 1845-1918

Standard-Cantor-Menge, Cantor-Staub, Wischmenge (offene Intervalle werden weggelassen → überabzählbare Menge vom Maß Null)

$$l(n) = \left(\frac{1}{3}\right)^n \quad (5.6)$$

$$N(n) = 2^n \quad (5.7)$$

$$D = \lim_{n \rightarrow \infty} \frac{n \ln 2}{n \ln 3} \quad (5.8)$$

$$D(\text{Cantor-Staub}) = \ln 2 / \ln 3 \approx 0.6309 \dots$$

### 5.2.4 Logistische Abbildung

$$D(\text{Logistische Abbildung für } r = r_\infty) \approx 0.54 \dots$$

## 5.3 Dynamische Systeme

Metrische Struktur von Phasenräumen dynamischer Systeme

- Mond-Exzentrizität (Saturn)
- $D(\text{Punktattraktor}) = 0$
- $D(\text{Grenzyklus}) = 1$
- $D(\text{2er Torus}) = 2$
- $D(\text{Lorenz-Attraktor: } r = 28, \sigma = 10, b = 8/3) \approx 2.06 \dots$

## 5.4 Landschaftsgeneratoren

## 5.5 Literatur

Classics on Fractals 1993 Gerald A. Edgar (ed.) Studies in Nonlinearity, Addison-Wesley Publ. Company, Reading, MA (Originalarbeiten von Weierstraß, Cantor, Koch, Carathéodory, Hausdorff, ... , Lévy, ... , Mandelbrot)

Perrin, Jean, Die Atome, Dresden 1913

Reif, F., Statistische Theorie der Wärme, De Gruyter, 1987

Lauwerier, Hans, Fraktale verstehen und selbst programmieren, R.Wittig Fachbuchverlag, Hückelhoven (BASIC-Disk) (der Fraktal)

Mandelbrot, Benoît, Die fraktale Geometrie der Natur, Akademie-Verlag, Berlin 1987, W.H. Freeman & Co. 1977 (das Fraktal)

Fricke, Jochen 1986 Fraktale Systeme, Physik in unserer Zeit **17**, 151-156

West, Bruce J. & Ary L. Goldberger 1987 Physiology in Fractal Dimensions, American Scientist

75, 354-365

Sander, Leonhard M. 1987 Fractal Growth, Scientific American **255**, Nr. 1, 94-100

Sander, Leonhard M. 1986 Fractal Growth Processes, Nature **322**, 789-793

Falconer, K., Fractal geometry, Wiley, 1985

Haberland, Kolja 1993 Das Drachenfraktal, Preprint, TU Berlin

## 6. Die Brownsche Bewegung

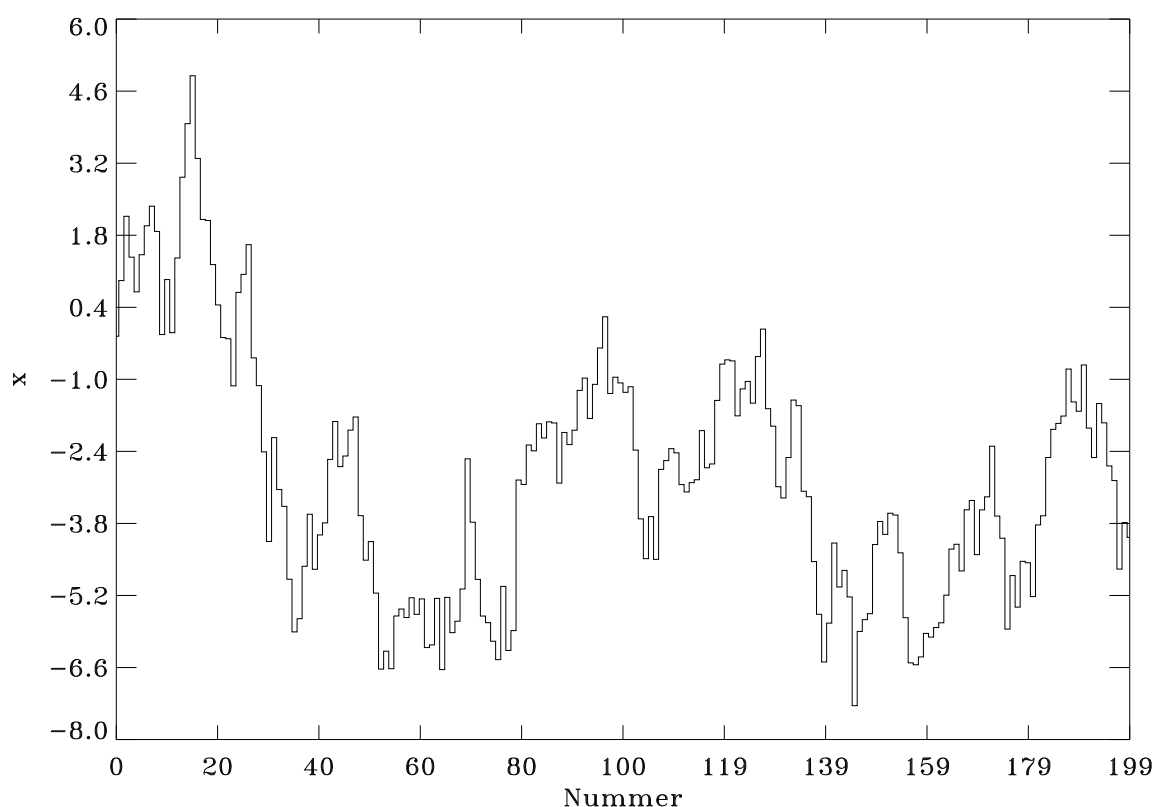
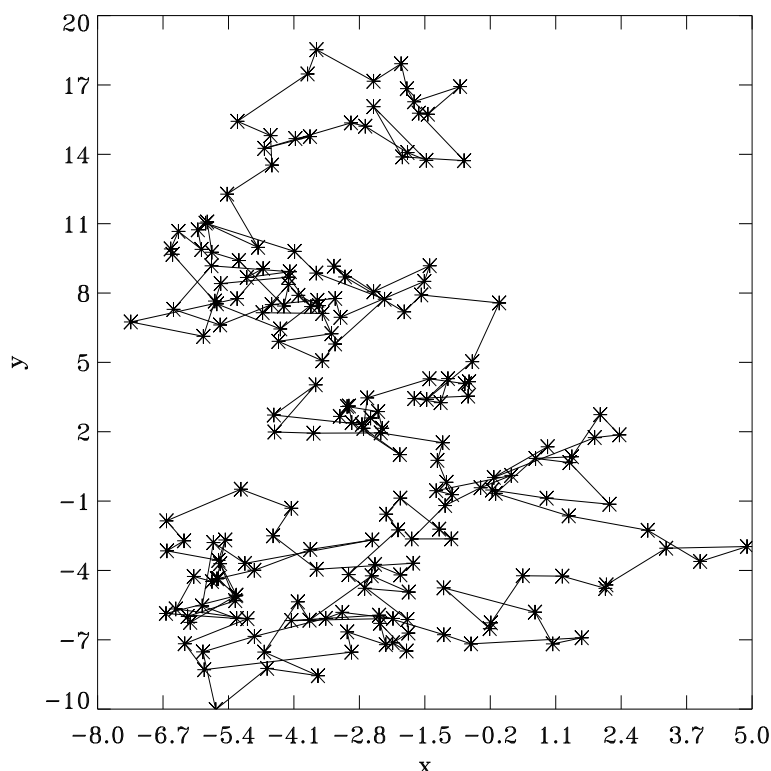


Abbildung 6.1: Realisierung der Brownschen Bewegung in einer Koordinate.

### 6.1 Entdeckung

1650 wurde das Phänomen der Brownschen Bewegung das erste mal durch Leeuwenhoek (1632-1723) erwähnt. Robert Brown, ein schottischer Regimentsarzt und Botaniker (1773-1858), untersuchte 1827 den Befruchtungsvorgang von Pflanzen und verfolgte den Lauf der kleinsten Pollenkörner, um die Stelle ihrer Vereinigung mit dem Ei zu entdecken. "Als ich die Gestalt dieser in Wasser getauchten Partikeln untersuchte, bemerkte ich, daß viele von ihnen sichtlich in Bewegung waren. Ihre Bewegung bestand nicht bloß aus einer Ortsveränderung in der Flüssigkeit, wie es sich durch die Veränderungen in ihren gegenseitigen Lagen ergab; sondern auch nicht selten aus einer Veränderung in der Gestalt der Teilchen selbst."

Blütenstaubkörner oder andere kleine ( $0.1 \leq a \leq 4\mu m$ ) in Wasser suspendierte Teilchen führen eine nicht endende und sehr irreguläre zitternde Bewegung aus (Beobachtungsintervall 15 ... 5 sec). Der



**Abbildung 6.2: Realisierung der Brownschen Bewegung eines Teilchens in der Ebene.**

sichtbarer Tanz von Rauchteilchen ist ein weiteres Beispiel für die Brownsche Bewegung.  
 Alte Brown-These: Die Lebenskraft (*vis vitalis*) von den Molekülen einer Pflanze wird lange Zeit über den Tod der Pflanze hinaus bewahrt.  
 Aber: Untersuchung von in Quarz eingeschlossenem Wasser mit Sporen und Pollen zeigt, daß die Zitterbewegung nicht die Lebenskraft darstellt. Die Ursache scheint nicht biologischer Natur zu sein.

## 6.2 Beginn der stochastischen Modellierung von Naturerscheinungen

Albert Einstein (1879-1955) über seine Entdeckung aus dem Jahre 1905:

“ Mein Hauptziel war es, Tatsachen zu finden, welche die Existenz von Atomen von bestimmter endlicher Größe möglichst sicherstellten. Dabei entdeckte ich, daß es nach der atomistischen Theorie eine der Beobachtung zugängliche Bewegung suspendierter mikroskopischer Teilchen geben müsse, ohne zu wissen, daß die Beobachtungen über die ‘Brownsche Bewegung’ schon lange bekannt waren.” (Einstein 1951)

Über die von der molekular-kinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, *Annalen der Physik* 17, 549 (1905), Eingangsdatum 11. Mai 1905.

Marian von Smoluchowski (1872-1917), *Annalen der Physik* 21, 756 (1906)

Idee: Die Atombewegung muß sich in der Teilchenbewegung zeigen. Die mittlere Entfernung, die das Körnchen zurücklegt, ist in vorhersehbarer Weise mit der Beobachtungszeit verknüpft. Zwar bleibt die

## 6. Die Brownsche Bewegung

Bewegung des einzelnen Rußkörnchens völlig unbestimmt, da die Stöße der einzelnen Moleküle selbst unter dem Mikroskop unbeobachtbar bleiben, doch läßt sich zumindest die Durchschnittsbewegung einer großen Zahl von Rußkörnchen vorhersagen. Einstein und Smoluchowski haben die wahre Geschwindigkeit, die meßbar ist, ganz beiseite gelassen, haben sich auch um den unendlich verwickelten Weg, den ein Teilchen während einer gegebenen Zeit beschreibt, nicht gekümmert, sondern haben als charakteristische Größe der Bewegung die gerade Verbindungslinie gewählt, welche den Ausgangs- und Endpunkt der Bahn verbindet und im Mittel augenscheinlich um so größer ist, je lebhafter die Bewegung ist. Diese Gerade soll die Verschiebung eines Teilchens während der betrachteten Zeit genannt werden. Ursache der Brownschen Bewegung sind die Stöße der Flüssigkeitsmoleküle. Einstein promovierte 1905 an der Universität Zürich mit der Arbeit "Eine neue Bestimmung der Moleküldimension". Frage von Einsteins Dissertation: Was bedeutet die Endlichkeit der Loschmidtschen Zahl  $N_A \approx 6 \cdot 10^{23} \text{ mol}^{-1} < \infty$  physikalisch? Das Ende Pascalscher Reisen!

### 6.3 Experimentelle Bestätigung der Theorie von Einstein – Smoluchowski

- 1907 erste Messungen in Gasen durch Ehrenhaft.
- 1908 Jean-Baptiste Perrin (1870-1942): Die Brownsche Bewegung und die wahre Existenz der Moleküle (1911).

" ... die Kurven, welche keine Tangente besitzen, sind die Regel, und die höchst regelmäßigen Kurven, wie der Kreis, sind zwar interessante, aber ganz spezielle Fälle. ... experimentelle Befunde bietet das Studium der Kolloide in Hülle und Fülle. Wir wollen z.B. eine jener weißer Flocken beobachten, die man erhält, wenn man Seifenwasser mit Kochsalz versetzt. Von weitem erscheint ihre Oberfläche scharf umrissen zu sein, sobald man sich aber nähert, verschwindet diese Schärfe. Das Auge vermag keine Tangente mehr an einen Punkt der Oberfläche zu anzulegen ... "

$$\langle m \rangle = \frac{1}{\tau} \int_0^{\tau} m dt \quad (6.1)$$

- Kochschen Schneeflocke
- Wiener-Prozeß (mathematisches Modell der Brownschen Bewegung)
- Ornstein-Uhlenbeck-Prozeß

### 6.4 Kurze Geschichte der Atomtheorie

#### 6.4.1 Antiker Atomismus

- Leukipp aus Milet (\* um 500, † um 440 v.Chr.)
- Demokrit von Abdera (\* um 460 v.Chr., †370 v.Chr.) " Urgründe des Alls sind die Atome und das Leere"
- Lukrez (\* um 96, †55 v.Chr.) überliefert dem römischen Denken die epikureische Atomlehre durch sein umfangreiches Lehrgedicht "Von der Natur der Dinge"

Gegner der körnigen Struktur der Materie:

- Anaxagoras (\* um 500, †428 v.Chr.) Warum sollte es eine Grenze der Teilbarkeit geben?

- Aristoteles (\* 384 v.Chr., †322 v.Chr.) kontinuierlicher, stetiger und ungeteilter Aufbau der Materie, es gibt weder Atome noch leeren Raum. Im leeren Raume ist keine Richtung ausgezeichnet, folglich auch keine Bewegung möglich. Leerer Raum ist undenkbar.
- Thomas von Aquin (13.Jh.): Aristoteles als Vorläufer Christi im weltlichen Bereich. Die plan- und ziellose Bewegung der Atome widersprach dem göttlichen Walten in unserer Welt. Die Atomisten lehren ein mechanisches, gottloses Universum.

Christianisierung der Atome im 17.Jh., Pierre Gassendi (1592-1655): Die Bewegung der Atome ist vom Schöpfer festgelegt worden.

#### 6.4.2 Erste Abschätzung der Loschmidtschen Zahl

Johann Chrysostomus Magnien (um 1650): "Mehr als einmal habe ich beobachtet, wie sich der Rauch eines verbrannten Weihrauchkornes so verbreitet, daß er einen Raum erfüllt, der mehr als 700 Millionen mal größer ist als das Korn selbst ... Das Weihrauchkorn kann in mindestens 1000 noch vom Auge feststellbare Teilchen geteilt werden. → 700,000,000,000 duftende Teilchen. Aber auch jene einzelnen Teilchen waren ein Konglomerat von verschiedenen Partikeln, und mit großer Wahrscheinlichkeit enthielt jedes von ihnen mindestens eine Million Atome." → 700,000,000,000,000,000 =  $7 \cdot 10^{17}$  Atome pro Weihrauchkorn.

#### 6.4.3 Die Chemie entdeckt das Atom

Antoine Lavoisier (1743-1794) Theorie der chemischen Elemente 1789 23 chemische Elemente  
Amadeo Avogadro (1776-1856) Bei Gasen stehen die Reaktionspartner stets in besonders einfachen Volumenverhältnissen  
John Dalton (1766-1844) Stöchiometrische Verhältnisse  
1860 Kongreß der Chemiker in Karlsruhe (für die Chemie gibt es Atome): Atome → Stöchiometrische Verhältnisse  
Dimitrij Mendelejew (1834-1907) 64 chemische Elemente

#### 6.4.4 Kinetische Gastheorie 1856

Statistik ohne Zeitentwicklung

- Karl August Krönig (1822-1879)
- Rudolf Emanuel Clausius (1822-1888): Hauptsätze der Wärmelehre
- James Clerk Maxwell (1831-1879)
- Ludwig Boltzmann (1844-1906)
- Gasdruck (Boyle: Berührung elastischer Teilchen; Newton: Abstoßende Fernwirkung; Daniel Bernoulli: Atomhagel)
- Pendel im luftgefüllten Raum (1860):  $N_{Stöße} \propto p, \lambda \propto 1/p$  Atome → Gaskinetik

Solvey-Kongreß 1911 zu Brüssel: Durchbruch der Atomtheorie auch in der Physik! Moleküle → Bewegung der Rußteilchen

## 6.5 Herleitungen

### 6.5.1 Plausible Herleitung der Einsteinschen Gleichung

$\langle x^2 \rangle$  bezeichnet das mittlere Quadrat der ein-dimensionalen Verschiebung nach  $N$  Einzelbewegungen der Länge  $\lambda$  mit zufälliger Bewegungsrichtung

$$\langle x^2 \rangle = \lambda^2 N \quad (6.2)$$

$v$  mittlere Geschwindigkeit des Teilchens  $N$  Zahl der Schritte in der Zeit  $t$

$$N = \frac{v}{t} \lambda \quad (6.3)$$

Gleichverteilungssatz der Energie. Das makroskopische Brownsche Teilchen sei im Gleichgewicht mit seiner mikroskopischen Umgebung.  $T$  ist die Temperatur der mikroskopischen Umgebung.

$$\langle \frac{1}{2} m v^2 \rangle = \frac{3}{2} k T \quad (6.4)$$

$\langle v \rangle$  selbst verschwindet, nicht aber  $\langle v^2 \rangle$ . Wegen

$$\langle v_x^2 \rangle = \frac{k T}{m} \quad (6.5)$$

und der "besseren" Statistik scheinen große Teilchen in Ruhe zu sein. Auf Grund ganz zufälliger Schwankungen werden sich nämlich die von allen Seiten erfolgenden Stöße von Zeit zu Zeit nicht kompensieren und so dem beobachtbaren Teilchen zu einer kurzen und abrupten Bewegung verhelfen.  $k$  bezeichnet die Boltzmann-Konstante mit dem Wert  $1.380,658(12) \cdot 10^{-23} \text{ JK}^{-1}$ .

Abschätzung der mittleren freien Weglänge  $\lambda$  des Brownschen Teilchens, mit dem Radius  $a$ . Die über die Strecke  $\lambda$  wirkende Stokessche Reibungskraft

$$F = 6\pi a \eta v \quad (6.6)$$

bringt das Teilchen mit einer bestimmten kinetische Energie zum Stillstand.

$$6\pi a \eta v \lambda = \frac{3}{2} k T \quad (6.7)$$

oder

$$\lambda = \frac{k}{t} 4\pi \eta a v \quad (6.8)$$

→

$$\langle x^2 \rangle = \frac{T}{k} 4\pi a \eta t \quad (6.9)$$

Einsteins Rechnung ergibt:

$$\langle x^2 \rangle = \frac{T}{k} 3\pi a \eta t \quad (6.10)$$

### 6.5.2 Einsteins Herleitung

$f(x, t)$  sei die Teilchenzahl im Einheitsvolumen.  $p(\Delta)$  sei die Wahrscheinlichkeit dafür, daß ein Teil während der Zeit  $\tau$  eine Ortsverschiebung um den Wert  $\Delta$  erfährt. Es gilt die Normierungsbedingung

$$\int_{-\infty}^{\infty} p(\Delta) d\Delta = 1. \quad (6.11)$$

Außerdem gilt

$$p(\Delta) = p(-\Delta). \quad (6.12)$$



Somit gilt auch

$$\int_{-\infty}^{\infty} \Delta p(\Delta) d\Delta = 0. \quad (6.13)$$

Dann ist  $dn$  die Anzahl der Teilchen, die eine solche Verschiebung erfährt

$$dn = np(\Delta)d\Delta. \quad (6.14)$$

In einem festen räumlichen Intervall ergibt sich nach einer Zeit  $\tau$  die Veränderung der Teilchenzahl dadurch, daß in allen um jeweils  $\Delta$  verschobenen Einheitsvolumina Veränderungen der Teilchenzahlen vor sich gehen:

$$f(x, t + \tau)dx = dx \int_{-\infty}^{\infty} f(x + \Delta, t)p(\Delta)d\Delta \quad (6.15)$$

Nun gilt obige Gleichung auch für kleine Veränderungen der Teilchenzahlen.

$$f(x, t + \tau) = f(x, t) + \tau \frac{\partial f}{\partial t} \quad (6.16)$$

$$f(x + \Delta, t) = f(x, t) + \Delta \frac{\partial f}{\partial x} + \frac{\Delta^2}{2!} \frac{\partial^2 f}{\partial x^2}. \quad (6.17)$$

Aus Gleichung 6.15 folgt damit

$$f + \frac{\partial f}{\partial t} \tau = f \int_{-\infty}^{\infty} p(\Delta) d\Delta + \frac{\partial f}{\partial x} \int_{-\infty}^{\infty} \Delta p(\Delta) d\Delta + \frac{\partial^2 f}{\partial x^2} \int_{-\infty}^{\infty} \frac{\Delta^2}{2} p(\Delta) d\Delta. \quad (6.18)$$

Wenn man nun

$$\frac{1}{\tau} \int_{-\infty}^{\infty} \frac{\Delta^2}{2} p(\Delta) d\Delta = D \quad (6.19)$$

setzt, so bekommt man in niedrigster Näherung eine Gleichung, die dem Fickschen Gesetz entspricht

$$\frac{\partial f}{\partial t} = D \frac{\partial^2 f}{\partial x^2} \quad (6.20)$$

und deren Lösung

$$f(x, t) = \frac{n}{\sqrt{4\pi Dt}} \exp - \frac{x^2}{4Dt} \quad (6.21)$$

ist. Damit ist

$$\sigma_x^2 = 2Dt. \quad (6.22)$$

### 6.5.3 Langevins Herleitung (1908)

Newtonsches Grundgesetz für eine Kugel mit dem Durchmesser  $a$ , die sich in einer Flüssigkeit mit der dynamischen Viskosität  $\eta$  unter dem Einfluß einer zufälligen Kraft  $X$  bewegt:

$$m\ddot{x} = -6\pi\eta a\dot{x} + X \quad (6.23)$$

Erste stochastische Differentialgleichung zur Beschreibung der Natur.

$$\frac{m}{2} \frac{d^2}{dt^2} x^2 - mv^2 = -3\pi\eta a \frac{d}{dt} x^2 + Xx \quad (6.24)$$

Bei der Mittelung  $\langle Xx \rangle$  für eine große Teilchenzahl gilt, wegen der Unabhängigkeit der Bewegung von mikroskopischen und makroskopischen Teilchen

$$\langle Xx \rangle = 0, \quad (6.25)$$

## 6. Die Brownsche Bewegung

und damit

$$\langle \frac{1}{2}mv_x^2 \rangle = \frac{1}{2}kT \quad (6.26)$$

$$\frac{m}{2} \frac{d^2 \langle x^2 \rangle}{dt^2} + 3\pi\eta a \frac{d \langle x^2 \rangle}{dt} = kT \quad (6.27)$$

→

$$\frac{d \langle x^2 \rangle}{dt} = \frac{kT}{3\pi\eta a} + C \exp - \frac{6\pi\eta a t}{m} \quad (6.28)$$

$$\frac{d \langle x^2 \rangle}{dt} = \frac{kT}{3\pi\eta a} \quad (6.29)$$

$$\langle x^2 \rangle = 2Dt \quad (6.30)$$

mit

$$D = \frac{k}{T} 6\pi\eta a \quad (6.31)$$

## 6.6 Simulation der Brownschen Bewegung

Zufallszahlengeneratoren. Gauß-Verteilung. Mittelwert. Standardabweichung. Realisierungen.

## 6.7 Bestimmung der Loschmidtschen Zahl

$$N_A = \frac{R}{k} \quad (6.32)$$

$R =$  universelle Gaskonstante  $= 8.314,510(70) \text{ Jmol}^{-1}\text{K}^{-1}$

Bestimmung von  $a$ ,  $x$  und  $t$ .

→ prinzipielle Begrenzung der Genauigkeit bestimmter Meßverfahren:

- Drehspiegelschwingung (Zeigerinstrument)
- Widerstandsrauschen (elektronische Meßverfahren)

## 6.8 Literatur

Brown, Robert (1828) A brief account of microscopical observations made in the month of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and anorganic bodies, Philos. Mag. Ann. of Philos. New ser. 4, 161-178

Chandrasekhar, Subrahmanyan (1986) Marian Smoluchowski, Warschau (Biographie, Theorie, Originalarbeiten)

Einstein, Albert (1922) Untersuchungen über die Theorie der Brownschen Bewegung. Ostwald's Klassiker der exakten Wissenschaften, Nr. 199, Leipzig (Originalarbeiten)

Einstein, Albert und Leopold Infeld (1950) Die Evolution der Physik, Paul Zsolnay Verlag, Wien-Hamburg, S.74-78 (elementare Einführung zur Brownschen Bewegung)

Feynman, Richard P., Robert B. Leighton, Matthew Sands (1973) Feynman Vorlesungen über Physik. Mechanik, Strahlung und Wärme, Band I.2, R. Oldenbourg Verlag, München-Wien, S. 41-1ff. (Einführung)

Gardiner, C.W. (1985<sup>2</sup>) Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, Springer-Verlag, Berlin, Kapitel 1.2 (Einführung in die stochastischen Methoden)

Griechische Atomisten. Texte und Kommentare zum materialistischen Denken der Antike (1977) Hergegeben von Fritz Jüss, Reimer Müller und Ernst Gunther Schmidt, Verlag Philipp Reclam jun., Leipzig

Honerkamp, Josef (1990) Stochastische Dynamische Systeme. Konzepte, numerische Methoden, Datenanalyse, VCH Verlagsgesellschaft, Weinheim (Brownsche Bewegung als stochastischer Prozeß)

Lavenda, Bernhard H. (1985) Die Brownsche Bewegung, Spektrum der Wissenschaft, Weinheim, S.58-68 (Brownsche Bewegung heute)

Mandelbrot, Benoît B. (1987) Die fraktale Geometrie der Natur, Akademie-Verlag, Berlin, S.24ff. 364ff., Kap. IX

Perrin, Jean (1914) Die Atome, Verlag von Theodor Steinkopff, Dresden 1914 (hervorragende Beschreibung des Themas)

Reif, Frederik (1976) Grundlagen der Physikalischen Statistik und der Physik der Wärme, Walter de Gruyter, Berlin-New York, S.295, 661ff. (ausführliche Darstellung)

Röpke, Gerd (1987) Statistische Mechanik für das Nichtgleichgewicht, VEB Deutscher Verlag der Wissenschaften, Berlin (Brownsche Bewegung als stochastischer Prozeß)

Schallreuter, W. (1971<sup>21</sup>) Grimsehl. Lehrbuch der Physik. Bd. 1, Mechanik- Akustik-Wärmelehre, BSB B.G. Teubner Verlagsgesellschaft, Leipzig, S. 433f., 536ff. (Einführung in die experimentellen Befunde)

Schrödinger, Erwin (1989) Die Natur und die Griechen, Diogenes, Zürich (Denkansätze der Antike und moderne Naturwissenschaft)

Schrödinger, Erwin (1989<sup>3</sup>) Was ist Leben? Die lebende Zelle mit den Augen des Physikers betrachtet, Piper, München-Wien (Uhr beim Anfall Brownscher Bewegung)

Segrè, Emilio (1981) Die großen Physiker und ihre Entdeckungen, Bd.2, Von den Röntgenstrahlen zu den Quarks, R.Piper Verlag & Co. München, S.99, 323f. (plausible Herleitung)

Sommerfeld, Arnold (1965) Vorlesungen über Theoretische Physik. Thermodynamik und Statistik, Akademische Verlagsgesellschaft Geest & Portig K.-G., Leipzig, S.154-159 (das Beste, alle anderen haben hier ungenau abgeschrieben)

Sexl, Roman S. (1982) Was die Welt zusammenhält. Physik auf der Suche nach dem Bauplan der Natur, Deutsche Verlags-Anstalt, Stuttgart, S. 84, 121ff. (sehr gute Beschreibung der Ideengeschichte der Physik)

Treder, Hans-Jürgen (1983) Große Physiker und ihre Probleme. Studien zur Geschichte der Physik, Akademie-Verlag, Berlin, S.122-125 (tiefgründige Darstellung der physikalischen Problemstellungen)

## 6. *Die Brownsche Bewegung*

Trigg, George L. (1984) Experimente der modernen Physik. Die Existenz der Atome, Akademie-Verlag, Berlin, S.33-50 (kurze Beschreibung der Details der Perrinschen Experimente)

Uhlmann, Armin (1979) Das wissenschaftliche Erbe Albert Einsteins, Leipziger Universitätsreden, Neue Folge, Heft 54, 11-19 (kurze Darstellung und Einordnung der Brownschen Bewegung)

Wiener, Norbert (1963<sup>2</sup>) Kybernetik. Regelung und Nachrichtenübertragung im Lebewesen und in der Maschine, Econ-Verlag, Düsseldorf-Wien, S.12f., 97-123, 215f., 229ff. (Einführung in die Idee des Wiener-Prozesses)

# 7. Stöße & Billards

## 7.1 Stoßgesetze für Punktmassen

Punktmasse – Die Kopplung von Schwerpunktbewegung und innere Bewegung ist vernachlässigbar.  
Stoß – Austausch von Energie, Impuls

Gesamtimpulserhaltung:

$$m_1 \vec{v}_1 + m_2 \vec{v}_2 = m_1 \vec{v}'_1 + m_2 \vec{v}'_2 \quad (7.1)$$

Gesamtenergieerhaltung beim elastischen Stoß:

$$m_1 v_1^2 + m_2 v_2^2 = m_1 v_1'^2 + m_2 v_2'^2 \quad (7.2)$$

4 Gleichungen, 6 Unbekannte ( $\vec{v}'_1, \vec{v}'_2$ ) → die Geschwindigkeitsvektoren müssen nicht alle in einer Ebene liegen.

Auswertung im Ruhssystem/Schwerpunktsystem

$$m_1 \vec{v}_1 + m_2 \vec{v}_2 = m_1 \vec{v}'_1 + m_2 \vec{v}'_2 = \vec{0} \quad (7.3)$$

$$m_1 v_1^2 + m_2 v_2^2 = m_1 v_1'^2 + m_2 v_2'^2 = 0 \quad (7.4)$$

$$\Rightarrow m_1 v_1^2 = -m_2 v_2^2, \quad m_1 v_1'^2 = -m_2 v_2'^2, \quad v_1^2 = v_1'^2 \quad (7.5)$$

### 7.1.1 Gerader Stoß für $v_2 = 0$

Für zwei Punktmassen folgt mit den Gl. (7.1) und (7.2):

$$2v_1'v_2' = 0 \quad \rightarrow v_1' = 0 \quad (7.6)$$

$$v_2' = \frac{2m_1}{m_1 + m_2} v_1 \quad (7.7)$$

Wegen

$$v_2' = \frac{2}{1 + m_2/m_1} v_1 \quad (7.8)$$

und

$$v_1' = \frac{m_1 - m_2}{m_1 + m_2} v_1 \quad (7.9)$$

gilt für  $m_1 \ll m_2$

$$v_2' = 0, \quad v_1' = -v_1 \quad (\text{Reflexion}) \quad (7.10)$$

Speziell findet für  $m_1 = m_2$  ein totaler Impulsaustausch statt:

$$v_1' = 0, \quad v_2' = v_1 \quad (7.11)$$

Für drei Punktmassen folgt mit den Gl. (7.1) und (7.2) mit  $v_2 = 0$  und  $v_3 = 0$ :

$$2v_1'v_2' + 2v_1'v_3' + 2v_2'v_3' = 0 \quad \rightarrow v_1' = v_2' = 0 \quad (7.12)$$

### 7.1.2 Kugelstoß

Stoß – plötzliches Zusammentreffen zweier oder mehrerer fester Körper (Austausch von Energie, Impuls, Drehimpuls)

- Massenpunkte – homogene Kugeln ohne Rotation
- elastisch: Neutronen durch Metalle, Fußball, Billard, ideales Gas
- plastisch: Schneeball, Ringmaterie, Galaxien, Wirbel, Asteroideneinschlag, Kernmaterie

Während einer sehr kurzen Zeit wirken sehr große Kräfte (Stoßkräfte). Statt

$$m\dot{\vec{v}} = \vec{F} \quad (7.13)$$

und

$$m d\vec{v} = \vec{F} dt \quad (7.14)$$

wird die Änderung der Bewegungsgröße beim Stoß (gleich Kraftstoß) betrachtet:

$$m\vec{v} - m\vec{v}_0 = \int_0^T \vec{F} dt \quad (7.15)$$

- zentral: Stoßnormale  $\parallel$  Schwerpunktverbindungsrichtung
- gerade (schief):  $\vec{v}_1 \parallel \vec{v}_2$

### 7.1.3 Streu-Phänomene

$$\sigma(\theta, \phi) \sin \theta d\theta d\phi \quad (7.16)$$

- Franck-Hertz-Versuch (unelastische Streuung ab Mindestenergie, diskrete Übergänge)
- Rutherford-Streuung
- Gravitative Streuung (Saturn-Ringstruktur)

## 7.2 Elastische Festkörper

### 7.2.1 Kugelstoßkette – Heinrich Hertz

In der Physikalischen Gesellschaft war mehrfach über die Newtonschen Ringe diskutiert worden. Dabei war die Frage nach der Form der aufeinander gepreßten Glasflächen in der unmittelbaren Nähe ihres Berührungspunktes aufgetaucht. Dies nahm Hertz zum Anlaß zu einer theoretischen Untersuchung über die Berührung elastischer Körper (Berührungsfläche, Spannungsverteilung, geodätische Messungen, Härte)

#### Über die Berührung fester elastischer Körper<sup>1</sup>

Die Schlußbetrachtungen betreffen den Stoßvorgang

$$p = k\alpha^{\frac{3}{2}} \quad (7.17)$$

wobei gilt  $p$  = Druck,  $k$  = Konstante,  $\alpha$  = Verschiebung der Kugeloberfläche in radialer Richtung.

<sup>1</sup>Journal für reine und angewandte Mathematik **92** (1881) 156-171

Über die Berührung fester elastischer Körper und über die Härte<sup>2</sup>

### 7.3 Unelastische Festkörper

### 7.4 Billards

### 7.5 Literatur

Budó, A. (1974) Theoretische Mechanik, VEB Deutscher Verlag der Wissenschaften, Berlin

Herrmann, F., Schmälzle, P. (1981) Amer.J.Phys. 49, 761

Hertz, Heinrich (1895) Gesammelte Werke, Band 1, Schriften vermischten Inhalts, Herausgegeben von Ph. Lenard, Johann Ambrosius Barth, Leipzig

Hertz, Heinrich (1971) Über sehr schnelle elektrische Schwingungen, Vier Arbeiten von Heinrich Hertz, Akademische Verlagsgesellschaft Geest Portig, Leipzig

Landau, L.D., Lifschitz, E.M. (1973) Lehrbuch der Theoretischen Physik, Bd.1, Mechanik, Akademie-Verlag, Berlin

Korsch, Hans Jürgen, Bruno Mirbach & Hans-Jörg Jodl 1987 Chaos und Determinismus in der klassischen Dynamik: Billard-Systeme als Modell, PdN-Physik **36**, 2-10

Jung, Christov 1992 Chaos & Streuung, Wissenschaft und Fortschritt **42**, 110-113

---

<sup>2</sup>Aus den Verhandlungen des Vereins zur Beförderung des Gewerbefleißes, Berlin, November 1882

# 8. Körper mit gravitativer Wechselwirkung

Stabilität des Sonnensystems: Monde (Poincaré, Wisdom)

## 8.1 Zwei - Körper - Problem

Kepler-Gesetze  
Gezeiten

## 8.2 Drei - Körper - Problem

Kleine Stern- oder Galaxiengruppen (Ann.Rev.Astron.Astrophys.)

## 8.3 Beschränktes Drei - Körper - Problem

Kleine Stern- oder Galaxiengruppen (Ann.Rev.Astron.Astrophys.)  
Monde, Planetensystem, Saturnring

## 8.4 Teilchen Simulation

Kugelhaufen, Galaxien, Galaxienhaufen

### Teilchen – Teilchen – Methode

$O(N^2)$  paarweise Wechselwirkungen.

### Teilchen in Zelle Methode

(Particle Mesh Method or Particle - in - Cell Method)

$O(N + M \log M)$  Berechnungen nötig, wobei  $M$  die Gitterpunktzahl ist. Das Gitter führt zu einer beschränkten Auflösung und stark ungleichförmige Körperverteilungen schränken die Leistung dieses Ansatzes ein.

### Die Methode von Greengard & Rokhlin

$O(N)$  Berechnungen



**Teilchen - Teilchen – Teilchen - Zellen Methode**

**Finite - Particle Method**

**8.5 Literatur**

G.W. Collins, *The Foundations of Celestial Mechanics*, Pachart Publishing House, Tucson, 1987

L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, The MIT Press, Cambridge, Massachusetts, London, England, 1988

R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, McGraw Hill, New York, 1981

M. Schneider, *Himmelsmechanik*, B-I Wissenschaftsverlag, Mannheim, Wien, Zürich, 1981

## 9. GLOSSAR

**Apfelmännchen** Spitzname einer ebenen grafischen Darstellung der nach dem Mathematiker Benoît Mandelbrot benannten Mandelbrotmenge. Sie erinnert an mehrere aufeinandergesetzte Äpfel verschiedener Größe.

**Bifurkation** lat: bifurcum = gegabelt

**Brownsche Bewegung**(engl. Brownian motion, kurz Bm) Der englische Botaniker Robert Brown entdeckte 1827 bei seinen Forschungen, daß - unter einem Mikroskop betrachtet - kleinste Teilchen, die in einer Flüssigkeit schweben, heftige, ungeordnete Zickzackbewegungen vollführen. Die Brownsche Bewegung läßt sich als stochastischer Prozeß einer eindimensionalen Variablen auffassen, wenn man die Bewegungen eines Teilchens im Raum aufzeichnet und dabei nur eine Dimension berücksichtigt. Die Brownsche Bewegung ist statistisch selbstähnlich und besitzt die fraktale Dimension 1.5.

**Cantor-Menge** Wer aus einer Linie das mittlere Drittel ausschneidet, aus den verbliebenen Stücken jeweils das mittlere Drittel herausschneidet und so weiter, der wandelt auf den Spuren des Mathematikers Georg Cantor. Wird das Ausschneiden unendlich oft wiederholt, entsteht eine Menge mit unendlich vielen vereinzelt Punkten: Die Cantor-Menge, später auch "STAUB-Fraktal" genannt. Cantor hat es Benoît Mandelbrot zu verdanken, daß seine Entdeckung nicht im Kabinett mathematischer Kuriositäten verblieb. Mandelbrot fand in den 60iger Jahren eine Anwendung für die Cantor-Menge: Mit ihr ließ sich die Häufigkeit von Störungen, bei der Fernübertragung von Computerdaten beschreiben. Die Fehler treten gehäuft auf, unterbrochen von Phasen störungsfreier Übertragung. In jeder "Fehlerphase" finden sich wiederum Pausen und Häufungen. Die fraktale Dimension der Cantor-Menge beträgt  $D=0.6309$ .

**Chaos** (griech: gestaltlose Urmasse) In der Alltagssprache gilt Chaos als das Gegenteil von Ordnung oder die Abwesenheit von Ordnung. Das Wort selbst stammt aus der griechischen Mythologie: Nach Hesiods Theogonie um 700 v. Chr. war im Anfang das Chaos - eine grenzenlose Leere, aus der dann Gaia entstand, die Erde. Ähnlich heißt es in der Genesis, dem Eröffnungskapitel der Bibel: "Und die Erde war wüst und leer." Auch die Schöpfungsmythologie vieler anderer Völker beginnt im Chaos. Mathematiker und Naturwissenschaftler sprechen dagegen von "chaotischen" Systemen, wenn deren Entwicklung zwar streng determiniert, aber nur beschränkt vorhersagbar ist. Paradox erscheint daher der Begriff deterministisches Chaos. Dieses Chaos - auf den ersten Blick bloßer Zufall - entsteht streng gesetzmäßig etwa in Turbulenzen oder bestimmten Doppelpendeln. Trotzdem ist das Verhalten deterministischer chaotischer Systeme nicht berechenbar, da sie äußerst empfindlich auf kleinste Veränderungen der Anfangsbedingungen reagieren. Weil sich Startwerte jedoch prinzipiell nicht beliebig genau ermitteln und damit nie exakt wieder herstellen lassen, ist eine Vorausberechnung unmöglich. Die Diagnose deterministisches Chaos bedeutet aber nicht nur Enttäuschung über die langzeitigen Vorhersagen - in der scheinbaren Regellosigkeit existieren geordnete Strukturen, etwa das Feigenbaumszenario.

**Dendritisches Wachstum** Baumartig verzweigt wachsen beispielsweise Dendriten (griech: dendron = baum) von Nervenzellen - Fortsätze, über die sie Erregungen empfangen: von den Hauptsträngen spalten sich kleinere Äste ab, daraus sprießen Zweige, aus diesen Zweiglein und so fort. Nicht nur biologische Gebilde, sondern auch anorganische wie Blitze oder Schneeflocken "wachsen" dendritisch. Ein Wechselspiel von mikroskopischen und makroskopischen Kräften bestimmt, in welche

Richtung die weitere Ausdehnung gerade am günstigsten ist. Bei der Schneeflocke wirkt beispielsweise auch die winzige Menge Wärme, die entsteht, wenn ein Wassermolekül "festfriert". Auch die Wind- und Temperaturverhältnisse beim Fall aus den Wolken beeinflussen die Form der Flocke. Schließlich gleicht keine der anderen.

**Determinismus** Philosophische Auffassung, daß alle Geschehnisse vorherbestimmt sind. Der Determinismus erlebte einen Aufschwung in der Zeit der klassischen Physik, als das Weltgeschehen mit Hilfe der Newton-Mechanik zumindest im Prinzip berechenbar zu sein schien. Zugespißt formulierte der Mathematiker Pierre Simon de Laplace diese Weltsicht im 18. Jahrhundert (Laplacescher Dämon). Erst Anfang unseres Jahrhunderts erkannte der französische Mathematiker Henri Poincaré, daß das Verhalten nichtlinearer Systeme unvorhersehbar sein kann - auch wenn die physikalischen Gesetze bekannt sind, nach denen es sich entwickelt.

**Dimension** Wir unterscheiden zwischen fraktaler und topologischer Dimension. Während die topologische Dimension nur ganzzahlige Werte annehmen kann, ist die fraktale Dimension eine reelle Zahl. Dabei stellt die fraktale Dimension eine Verbindung zwischen zwei benachbarten topologischen Dimensionen dar. Nehmen wir beispielsweise eine fraktale Dimension mit Werten zwischen 1 und 2 : Es wird hierbei der Übergang von einer geraden Linie zu einer Fläche beschrieben. Die fraktale Dimension ist hier ein Maß dafür, wie sehr eine kurvige Linie eine Fläche füllt. Je größer die fraktale Dimension, desto "wilder", "zackiger", "flächenfüllender" die Linie. Beispiel: Man nehme ein Blatt Papier ( zweidimensional ), und schwärze es mit einem Stift, ohne auch nur einmal abzusetzen. Das Ergebnis ist eine Linie von höchster fraktaler Dimension ( eventl. 1.9999 ). Natürlich beißt sich die theoretische Vorstellung, daß eine Linie keine Dicke hat, mit der Vorstellung, daß damit eine Fläche überdeckt werden könne. Eine Million Linien haben schließlich immer noch keine Dicke.

**Diskretisierung** Der Schritt vom Unendlichen (und damit nicht Aufzählbaren) ins Abzählbare. Diskretisieren bedeutet die Anwendung einer Abbildung, die eine unendliche Menge in eine abzählbare Menge überführt. Beispielsweise könnte man die reellen Zahlen von 0 bis 9 auf die Zahl 1 abbilden, die Zahlen von 9 bis 19 auf die Zahl 2 usw... . Also eine Abbildung von reellen Zahlen auf natürliche Zahlen. Die natürlichen Zahlen sind abzählbar, denn jede Zahl ist gerade ihr eigener Abzählwert, aber nicht endlich: Es gibt keine größte natürliche Zahl. Tatsächlich gibt es in einem Computer keine reellen Zahlen. Der Computer ist nur eine endliche, diskrete Maschine (zwangsläufig, da ein Computer digital arbeitet). Reelle Zahlen werden nur angenähert wiedergegeben.

**Dissipative Systeme** lat: dissipare = zerstreut

**Entropie** griech: entrepein = verändern

**Fraktal** Der Begriff Fraktal (fractal) ist von Mandelbrot erfunden worden. Er leitete Fraktal vom lateinischen Adjektiv fractus ab. Das entsprechende Verb frangere bedeutet: brechen, unregelmäßige Fragmente bilden. Fraktale sind selbstähnliche mit Struktur versehene Gebilde. Sie lassen sich beliebig vergrößern und verlieren nichts von ihrer Komplexität. Mit dem Begriff Fraktal können wir eine ganze Reihe von Objekten klassifizieren.

Die Gebilde der klassischen Geometrie, wie Linie, Quadrat, Kreis, Würfel, Kegel usw. gehören nicht dazu.

**Fraktale Brownsche Bewegung** (engl. fractional Brownian motion, kurz fBm) Die Fraktionale Brownsche Bewegung ist eine Verallgemeinerung der Brownschen Bewegung, die zu selbstähnlichen, stochastischen Prozessen mit beliebigen fraktalen Dimensionen führt. Sie ist das fraktale Modell, das als Grundlage für Computer-Algorithmen zur Erzeugung von Landschaften und Planeten dient.

**Iteration** lat: iterare = wiederholen. Wiederholte Anwendung derselben Rechenvorschrift, wobei jedes Ergebnis der Rechnung wiederum als Ausgangswert dient ("endliche Rekursion").

**Kochsche Schneeflocke** Fraktales Gebilde, das nach einem einfachen Rezept entsteht: Bei einem gleichseitigen Dreieck wird auf das mittlere Drittel jeder Seite wieder ein gleichseitiges Dreieck gezeichnet, für das die Regel wieder gilt: so entwickelt sich nach unendlich vielen Wiederholungen ein mathematisches Monster - ein Schneeflockenähnliches Gebilde, das von einer unendlich langen Kurve umgrenzt wird.

**Laplacescher Dämon** Fiktives Wesen, erdacht von dem französischen Mathematiker Pierre Simon de Laplace im Jahre 1776 als radikale Konsequenz des Determinismus. Der Dämon braucht nur zu einer Zeit Ort und Geschwindigkeit aller Materialteilchen im Universum zu kennen, um - als wissender Historiker und Prophet zugleich - Vergangenheit und Zukunft berechnen zu können. Der freie Wille wäre damit eine Illusion.

**Lindemayer-System** Ein Lindemayer-System ist ein mathematisches formales System. Die Bezeichnung Lindemayer-System ist auf den Biologen Aristid Lindenmayer (1925-1989) zurückzuführen. Lindemayer konstruierte ein formales System, als theoretisches Werkzeug, um die Entwicklung von Zellorganismen zu erforschen. Mit einem Lindemayer-System lassen sich eine Vielzahl von Fraktalen definieren und erzeugen. Insbesondere lassen sich damit Pflanzen und pflanzenähnliche Gebilde beschreiben. Prinzipielle Arbeitsweise: Wir ersetzen gegebene Symbole nach bestimmten vorgegebenen Regeln. Das wesentliche Merkmal dieser Ersetzung ist ihre Parallelität: Alle Ersetzungsregeln werden gleichzeitig angewandt. Auf Zellstrukturen oder Pflanzen bezogen, bedeutet die Anwendung der Regeln eine Fortentwicklung oder Wachstum des Organismus.

**Lorenz-Attraktor** Chaotischer Attraktor benannt nach seinem Entdecker, dem US-Meteorologen Edward Lorenz, und berühmt wegen seiner an ein Eulengesicht erinnernden Form.

**Mittelpunktverschiebung ( zufällige )** Es gibt verschiedene Methoden, um die fraktale Brownsche Bewegung zu approximieren. Die zufällige Mittelpunktverschiebung ist ein sehr verbreiteter Algorithmus, denn er ist kurz - und damit leicht zu implementieren - und er ist schnell.

**Multifraktal** Natürliche Objekte haben in bestimmten Bereichen ( Skalierungen ) ein bestimmtes charakteristisches Aussehen. Das können wir in einer fraktalen Dimension ausdrücken, die aber nur für diesen bestimmten Bereich gültig ist. Fraktale mit wechselnder fraktaler Dimension bezeichnen wir also als Multifraktale.

**Neuronale Netze** Neuronen sind Hirnzellen. Über ihre Fortsätze, die Signale empfangenden Dendriten und die Signale weiterleitenden Axone, sind diese Nervenzellen im Gehirn zu einem verzweigten Netzwerk verschaltet. Es enthält alle Erfahrung und alles Wissen seines Besitzers und entsteht durch Selbstorganisation: Neuronale Verbindungen stärken sich oder schwächen sich ab - je nach Aufgabe des Netzes, von der die einzelnen Nervenzellen nichts "wissen". Die Arbeitsprinzipien der neuronalen Netze des Gehirns unterscheiden sich prinzipiell von denen herkömmlicher Computer.

**Nichtlineare Systeme** Systeme, deren Entwicklung durch nichtlineare Gleichungen beschreibbar sind. Wenn die Elemente eines Systems nichtlinear Wechselwirken, kann das Ganze mehr sein als die Summe seiner Teile. Nur in nichtlinearen Systemen ist Chaos möglich.

**Poincaré-Schnitt** Ein nach dem französischen Mathematiker Henri Poincaré benannter Schnitt durch einen Attraktor. Er wird zum Beispiel sichtbar, wenn ein Computer alle Schnittpunkte eines dreidimensionalen Attraktors mit einer Ebene berechnet.

**Poincarésche Wiederkehr** Dieser Begriff aus der statischen Mechanik bezeichnet das erneute Durchlaufen eines Zustandes, in dem sich ein dynamisches System bereits einmal befunden hat.

**Quasi-periodisch** heißt eine Bewegung, die sich aus mehreren periodischen Teilbewegungen zusammensetzt, ohne selbst periodisch zu sein. Quasiperiodik tritt auf, wenn die Perioden der einzelnen Bewegungen kein gemeinsames Vielfaches haben, so daß eine exakte Poincarésche Wiederkehr ausgeschlossen ist.

**Rekursionsgleichungen** modellieren Rückkopplungen mathematisch.

**Schmetterlingseffekt** Sinnbild für die Unvorhersagbarkeit des Wettergeschehens. Da die Atmosphäre ein nichtlineares System ist, können auch kleinste Störungen den Wetterverlauf grundlegend ändern, die Luftwirbel, die ein Schmetterling in China verursacht, können Auslöser eines Hurrikans sein, der Wochen später über der Karibik tobt.

**Selbstähnlichkeit** ist eine Eigenschaft, die viele Fraktale zeigen: Verschieden starke Vergrößerungen der Objekte ähneln einander.

**Stochastik** Die Stochastik ist die Lehre vom Zufall.

**Zufall** Zufällige Vorgänge lassen sich prinzipiell nicht vorausberechnen, während für deterministisches

Chaos wenigstens kurzfristige Vorhersagen möglich sind. Oft ist es schwer zu entscheiden, ob echtes Chaos vorliegt. So gibt es Glückspielautomaten mit elektronisch gespeicherten Listen von "Zufallszahlen", die sämtliche Spielergebnisse vorherbestimmen.

**Voxel** (engl. volume pixel) Die Bezeichnung eines Punktes im Raum, dessen Position durch drei Koordinaten  $(x, y, z)$  bestimmt ist.

## 9. GLOSSAR

# 10. Turbo-Pascal-Texte

## Einführungen

Franken, Gerhard 1990 Einsteigerseminar Turbo Pascal 5.0/5.5, BHV-Verlag, Korschenbroich

Paulin, Gerhard 1988 Turbo-Pascal, VEB Verlag Technik, Berlin

Rosenbaum, Gabriele & Reiner Schölles 1989 Das große Turbo Pascal 5.5 Buch, Data Becker

Lehmann, Eberhard 1988 Mathematik-Unterricht mit Computer-Einsatz, Bd. 1, Grundlagen. Didaktische und methodische Hinweise für die Sekundarstufen I und II, Dümmler<sup>s</sup> Verlag, Bonn

Hillebrand, H.-P. 1992 Grundkurs Personal-Computer. Hard- und Software verstehen, Dümmler<sup>s</sup> Verlag, Bonn

```

:.....:
ATTRAKTO.PAS
:.....:
PROGRAM attraktor ;

USES crt, graph ;

CONST
    dt = 0.01 ;

VAR
    x_alt , x_neu , y_alt , y_neu , z_alt , z_neu : real ;
    gd , gm                                       : integer ;
    farbe                                         : integer ;

PROCEDURE farb(farbe:integer) ;
BEGIN
    IF z_neu > 90 THEN farbe := 1 ; exit;
    IF z_neu > 80 THEN farbe := 2 ; exit;
    IF z_neu > 70 THEN farbe := 3 ; exit;
    IF z_neu > 60 THEN farbe := 4 ; exit;
    IF z_neu > 50 THEN farbe := 5 ; exit;
    IF z_neu > 40 THEN farbe := 6 ; exit;
    IF z_neu > 30 THEN farbe := 7 ; exit;
    IF z_neu > 20 THEN farbe := 8 ; exit;
    IF z_neu > 10 THEN farbe := 9 ; exit;
    IF z_neu > 0 THEN farbe := 10; exit;
    IF z_neu > (-10) THEN farbe := 6; exit;
    IF z_neu > (-20) THEN farbe := 7; exit;
    IF z_neu > (-30) THEN farbe := 8; exit;

END;

BEGIN
    x_alt := 1 ;
    y_alt := 1 ;
    z_alt := 1 ;

    gd:=detect;
    initgraph(gd, gm, 'c:\tp\bgi');

    REPEAT

        x_neu := x_alt + dt*10*(y_alt-x_alt) ;
        y_neu := y_alt + dt*(x_alt*(28-z_alt)-y_alt) ;
        z_neu := z_alt + dt*(x_alt*y_alt-8/3*z_alt) ;

    {   writeln(x_neu, ' ', y_neu, ' ', z_neu);   }

        line (440,0,440,640) ; { x-achse }
        line (0,280,640,280) ; { y-achse }

        farb(farbe) ;

        outtextxy(10,305,'x') ;
        outtextxy(400,5,'y') ;

    {   putpixel ( getmaxx - round(x_neu+300) , getmaxy-round(y_neu+230) ,
        random(7)-6 );   }

        putpixel ( getmaxx - round((x_neu)*6+200) , getmaxy-round((y_neu)
        *6+200) , farbe );

        x_alt := x_neu ;
        z_alt := z_neu ;
        y_alt := y_neu ;

    UNTIL keypressed;

```



```

    closegraph ;
END.

::::::::::::::::::
BIFU.PAS
::::::::::::::::::
PROGRAM Bifurkationsdiagramm ;

{$n+}

USES graph,crt ;

VAR
    xn      : extended ;
    x       : extended ;
    a       : extended ;
    b       : extended ;
    xn1     : extended ;
    count   : integer ;
    lauf    : integer ;

PROCEDURE initunitgraph ;

VAR
    gm,gd : integer ;

BEGIN
    gd:=detect ;
    initgraph(gd, gm, 'C:\TP\BGI') ;
    cleardevice ;
    moveto(0,0) ;
END;

BEGIN
    CLRSCR ;

    x := 0.7 ;

    initunitgraph ;

    FOR lauf := 1 TO 640 DO
        BEGIN
            xn :=x ;
            count :=0 ;
            a := lauf/640+3 ;
            REPEAT
                xn1 := a*xn*(1-xn) ;
                xn := xn1 ;
                IF count > 300 THEN
                    BEGIN
                        putpixel ( lauf , getmaxy-round(xn*getmaxy) , green ) ;
                    END;
                count := count + 1 ;
            UNTIL count > 500

            END;

        readln;
        closegraph;
    END.

::::::::::::::::::
BROWN.PAS
::::::::::::::::::
program brown;
uses crt,graph;

```

```

type punkt=record
  x,y:integer;
end;

const anz=2;
  gd:integer=vga;
  gm:integer=vghi;

var teil:array[1..anz]of punkt;
  i,k,iter:integer;
  c,j:longint;
  sx,sy,dx,dy,mx,my:real;

procedure stat;
begin;
  j:=j+1;
  dx:=teil[1].x-teil[anz].x;
  mx:=mx+dx;
  dx:=dx*dx;
  dy:=teil[1].y-teil[anz].y;
  my:=my+dy;
  dy:=dy*dy;
  sx:=sx+dx;sy:=sy+dy;
  teil[anz].x:=teil[1].x;
  teil[anz].y:=teil[1].y;
end;

begin;
  randomize;sx:=0;sy:=0;mx:=0;my:=0;j:=0;c:=0;
  write('Meãintervall:');readln(iter);
  initgraph(gd,gm,'c:\tp\bgi');
  for i:=1 to anz-1 do begin
    with teil[i] do begin
      x:=random(640);
      y:=random(480);
    end;
  end;
  repeat
    c:=c+1;if c mod iter=0 then stat;
    for i:=1 to anz-1 do begin
      with teil[i] do begin
        putpixel((x+640) mod 640,(y+480) mod 480,lightgray);
        k:=random(2)+random(2);
        case k of
          0:x:=x+1;
          2:x:=x-1;
        end;
        k:=random(2)+random(2);
      case k of
        0:y:=y+1;
        2:y:=y-1;
      end;
      putpixel((x+640) mod 640,(y+480) mod 480,white);
    end;
  end;
  until j=10000;
  readln;
  restorecrtmode;
  sx:=sx/j;sy:=sy/j;
  writeln('Varianz-x: ',sx:10:4);
  writeln('Varianz-y: ',sy:10:4);
  writeln('Mittelwert-x: ',mx/j:10:4);
  writeln('Mittelwert-y: ',my/j:10:4);
  writeln('Gesamtzeit: ',c);
  writeln('Anzahl der Messungen: ',j);
  writeln('Zeitintervall: ',iter);
  readln;
end.

:::
FBM1.PAS
:::

```

```

PROGRAM fbm;

USES graph,crt;

CONST
    dtkonst    = 1 ;
    v          = 1 ;

VAR
    gd,gm      : integer ;
    x,y        : integer ;
    dtx,dty    : integer ;

PROCEDURE bildlayout;
BEGIN
    line(0,0,640,0);
    line(639,479,639,0);
    line(0,479,640,479);
    line(0,0,0,640);
END;

PROCEDURE zufallskoordinaten;
BEGIN
    dtx:=+dtkonst;
    dty:=+dtkonst;
    x:=round(random(640));
    y:=round(random(480));
END;

PROCEDURE nextpoint;
BEGIN
    putpixel(x,y,red);
    IF x>=638 THEN dtx:=-dtkonst;
    IF x<=2 THEN dtx:=+dtkonst;
    IF y>=478 THEN dty:=-dtkonst;
    IF y<=2 THEN dty:=+dtkonst;
    x:=x+dtx*v ;
    y:=y+dty*v;
    putpixel(x,y,lightgreen);
    delay(2);
END;

BEGIN
    gd:=vga;gm:=vgahi;
    initgraph(gd,gm,'c:\tp\bgi');
    bildlayout;

    zufallskoordinaten;

    REPEAT
        nextpoint;
    UNTIL keypressed;

    closegraph;

END.

::::::::::::::::::
FBM2.PAS
::::::::::::::::::
{ 2 ohne Verbindungslinie }

PROGRAM fbm;

USES graph,crt;

CONST
    dtkonst    = 1 ;
    v          = 1 ;

VAR
    gd,gm      : integer ;

```

```

        x1,y1      : integer ;
        x2,y2      : integer ;
        dtx1,dty1  : integer ;
        dtx2,dty2  : integer ;

PROCEDURE bildlayout;
BEGIN
    line(0,0,640,0);
    line(639,479,639,0);
    line(0,479,640,479);
    line(0,0,0,640);
END;

PROCEDURE zufallskoordinaten;
BEGIN
    randomize;
    dtx1:=+dtkonst;
    dty1:=+dtkonst;
    x1:=round(random(629))+2;
    y1:=round(random(471))+2;
    dtx2:=+dtkonst;
    dty2:=+dtkonst;
    randomize;
    delay(20);
    randomize;
    x2:=round(random(631))+2;
    y2:=round(random(479))+2;
END;

PROCEDURE nextpoint1;
BEGIN
    putpixel(x1,y1,red);
    IF x1>=638 THEN dtx1:=-dtkonst;
    IF x1<=2 THEN dtx1:=+dtkonst;
    IF y1>=478 THEN dty1:=-dtkonst;
    IF y1<=2 THEN dty1:=+dtkonst;
    x1:=x1+dtx1*v ;
    y1:=y1+dty1*v ;
    putpixel(x1,y1,lightgreen);
END;

PROCEDURE nextpoint2;
BEGIN
    putpixel(x2,y2,lightgreen);
    IF x2>=638 THEN dtx2:=-dtkonst;
    IF x2<=2 THEN dtx2:=+dtkonst;
    IF y2>=478 THEN dty2:=-dtkonst;
    IF y2<=2 THEN dty2:=+dtkonst;
    x2:=x2+dtx2*v ;
    y2:=y2+dty2*v ;
    putpixel(x2,y2,red);
END;

BEGIN
    gd:=vga;gm:=vgahi;
    initgraph(gd,gm,'c:\tp\bgi');
    bildlayout;

    zufallskoordinaten;

    REPEAT
        nextpoint2;
        nextpoint1;
        delay(3);
    UNTIL keypressed;

    closegraph;

END.

:::
FBM3.PAS

```

```

:.....:
{ 2 mit Verbindungslinie }

PROGRAM fbm;

USES graph,crt;

CONST
    dtkonst    = 1 ;
    v          = 1 ;

VAR
    gd,gm      : integer ;
    x1,y1      : integer ;
    x2,y2      : integer ;
    dtx1,dy1   : integer ;
    dtx2,dy2   : integer ;

PROCEDURE bildlayout;
BEGIN
    line(0,0,640,0);
    line(639,479,639,0);
    line(0,479,640,479);
    line(0,0,0,640);
END;

PROCEDURE zufallskoordinaten;
BEGIN
    dtx1:=+dtkonst;
    dty1:=+dtkonst;
    x1:=round(random(640));
    y1:=round(random(480));
    x2:=round(random(640));
    y2:=round(random(480));
END;

PROCEDURE nextpoint1;
BEGIN
    putpixel(x1,y1,red);
    IF x1>=638 THEN dtx1:=-dtkonst;
    IF x1<=2 THEN dtx1:=+dtkonst;
    IF y1>=478 THEN dty1:=-dtkonst;
    IF y1<=2 THEN dty1:=+dtkonst;
    x1:=x1+dtx1*v;
    y1:=y1+dty1*v;
    putpixel(x1,y1,lightgreen);
END;

PROCEDURE nextpoint2;
BEGIN
    putpixel(x2,y2,lightgreen);
    IF x2>=638 THEN dtx2:=-dtkonst;
    IF x2<=2 THEN dtx2:=+dtkonst;
    IF y2>=478 THEN dty2:=-dtkonst;
    IF y2<=2 THEN dty2:=+dtkonst;
    x2:=x2+dtx2*v;
    y2:=y2+dty2*v;
    putpixel(x2,y2,red);
END;

BEGIN
    randomize;
    gd:=vga;gm:=vgahi;
    initgraph(gd,gm,'c:\tp\bgi');
    bildlayout;

    zufallskoordinaten;

    REPEAT
        nextpoint1;
        nextpoint2;
        setcolor(yellow);

```

```

    line(x1,y1,x2,y2);
    setcolor(black);
    delay(20);
    line(x1,y1,x2,y2);

UNTIL keypressed;

closegraph;

END.

:::::::::::::
FBM4.PAS
:::::::::::::
{ 2 mit Verbindungslinie effekte }

PROGRAM fbm;

USES graph,crt;

CONST
    dtkonst    = 1 ;
    v          = 1 ;

VAR
    gd,gm      : integer ;
    x1,y1      : integer ;
    x2,y2      : integer ;
    dtx1,dty1  : integer ;
    dtx2,dty2  : integer ;

PROCEDURE bildlayout;
BEGIN
    line(0,0,640,0);
    line(639,479,639,0);
    line(0,479,640,479);
    line(0,0,0,640);
END;

PROCEDURE zufallskoordinaten;
BEGIN
    dtx1:=+dtkonst;
    dty1:=+dtkonst;
    x1:=round(random(640));
    y1:=round(random(480));
    x2:=round(random(640));
    y2:=round(random(480));
END;

PROCEDURE nextpoint1;
BEGIN
    putpixel(x1,y1,red);
    IF x1>=638 THEN dtx1:=-dtkonst;
    IF x1<=2 THEN dtx1:=+dtkonst;
    IF y1>=478 THEN dty1:=-dtkonst;
    IF y1<=2 THEN dty1:=+dtkonst;
    x1:=x1+dtx1*v;
    y1:=y1+dty1*v;
    putpixel(x1,y1,lightgreen);
END;

PROCEDURE nextpoint2;
BEGIN
    putpixel(x2,y2,lightgreen);
    IF x2>=638 THEN dtx2:=-dtkonst;
    IF x2<=2 THEN dtx2:=+dtkonst;
    IF y2>=478 THEN dty2:=-dtkonst;
    IF y2<=2 THEN dty2:=+dtkonst;
    x2:=x2+dtx2*v;
    y2:=y2+dty2*v;
    putpixel(x2,y2,red);
END;

```

```

BEGIN
  randomize;
  gd:=vga;gm:=vgahi;
  initgraph(gd,gm,'c:\tp\bgi');
  bildlayout;

  zufallskordinaten;

  REPEAT
    nextpoint1;
    nextpoint2;
    setcolor(yellow);
    line(x1,y1,x2,y1);
    setcolor(black);
    delay(7);
    line(x1,y1,x2,y2);

  UNTIL keypressed;

  closegraph;

END.

:::::::::::::
FEIGBAUM.PAS
:::::::::::::
program feighoch;
uses crt,dos,tpvga,tpdac;

const maxx=360;
      maxy=480;

type
  tergebnis=array[0..maxy] of word;
  pergebnis=^tergebnis;

var
  palette:dac all registers;
  f:file of dac all registers;
  x,y,z,i,leer,it:word;
  amin,amax,afak,xanf,rfak:real;
  ergebnis:tergebnis;

procedure berech(a:real);
var
  x1,x0:word;
  x,f:real;
begin
  x:=xanf;
  fillchar(ergebnis,sizeof(ergebnis),#0);
  for x1:=1 to leer+it do begin
    x:=a*x*(1-x);
    if x1>it then inc(ergebnis[round(x*maxy)]);
  end;
  for x1:=0 to maxy-1 do begin
    f:=ergebnis[x1]*rfak;
    ergebnis[x1]:=round(f);
    if ergebnis[x1]>253 then ergebnis[x1]:=254;
  end;
end;

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  write(' Relationsfaktor : ');
  readln(rfak);

```

```

write(' Anfangsschleifen : ');
readln(leer);
write(' I.-Tiefe : ');
readln(it);
grafik360x480;
assign(f, 'palette.pal');
reset(f);
read(f,palette);
close(f);
palette.register[254].rot:=45;
palette.register[254].blau:=45;
palette.register[254].gruen:=0;
palette.anzahl:=255;
palette.start:=0;
for x:=0 to 254 do
  linie360x480(x,0,x,1,x);
for x:=255 to maxx-1 do
  linie360x480(x,0,x,1,254);
setdacregisters(palette);
afak:=(amax-amin)/maxx;
for i:=0 to maxx-1 do begin
  berech(amin+i*afak);
  for y:=1 to maxy-1 do setze360x480(i,y,ergebnis[y]);
end;
repeat until readkey<>#0;
altermodus;
end.

```

```

::::::::::::::::::

```

```

FEIGEMPF.PAS

```

```

::::::::::::::::::

```

```

program feighoch;
uses crt, dos, tppvga, tppdac;

```

```

const maxx=360;
      maxy=480;

```

```

type

```

```

  tergebnis=array[0..maxy] of word;
  pergebnis=^tergebnis;

```

```

var

```

```

  palette:dac all registers;
  f:file of dac all registers;
  x,y,z,i,leer,it:word;
  amin,amax,afak,xanf,rfak,xend:real;
  ergebnis:tergebnis;

```

```

procedure berech(a:real);

```

```

var

```

```

  x1,x0,i,j:word;
  x,f,dx,x3:real;

```

```

begin

```

```

  dx:=(xend-xanf)/maxy;
  x:=xanf-dx;
  i:=0;
  repeat
    inc(i);
    x:=x+dx;
    x3:=x;
    for j:=1 to it do x3:=a*x3*(1-x3);
    f:=x3*rfak;
    x0:=round(f)+1;
    if x0>250 then x0:=254;
    ergebnis[i]:=x0;
  until i>=maxy
end;

```

```

begin

```

```

  writeln;
  write(' a min : ');
  readln(amin);

```



```

write(' a max : ');
readln(amax);
write(' x anf : ');
readln(xanf);
write(' x end : ');
readln(xend);
write(' Relationsfaktor : ');
readln(rfak);
write(' Anfangsschleifen : ');
readln(leer);
write(' I.-Tiefe : ');
readln(it);
grafik360x480;
assign(f, 'palette.pal');
reset(f);
read(f, palette);
close(f);
palette.register[254].rot:=45;
palette.register[254].blau:=45;
palette.register[254].gruen:=0;
palette.anzahl:=255;
palette.start:=0;
for x:=0 to 254 do
  linie360x480(x,0,x,1,x);
for x:=255 to maxx-1 do
  linie360x480(x,0,x,1,254);
setdacregisters(palette);
afak:=(amax-amin)/maxx;
for i:=0 to maxx-1 do begin
  berech(amin+i*afak);
  for y:=1 to maxy-1 do
    setze360x480(i,y,ergebnis[y]);
  end;
  repeat until readkey<>#0;
  altermodus;
end.

:::::::::::::::::::
FEIGHOC1.PAS
:::::::::::::::::::
program feighoch;
uses crt, dos, tppvga, tppdac;

const maxx=360;
      maxy=480;

type
  tergebnis=array[0..maxy] of word;
  pergebnis=^tergebnis;

var
  palette:dac all registers;
  f:file of dac all registers;
  x,y,z,i,leer,it:word;
  amin,amax,afak,xanf,rfak:real;
  ergebnis:tergebnis;

procedure berech(a:real);
var
  x1,x0:word;
  x,f:real;
begin
  x:=xanf;
  fillchar(ergebnis,sizeof(ergebnis),#0);
  for x1:=1 to leer+it do begin
    x:=a*x*(1-x);
    if x1>it then inc(ergebnis[round(x*maxy)]);
  end;
  for x1:=0 to maxy-1 do begin
    f:=ergebnis[x1]*rfak;
    ergebnis[x1]:=round(f);
    if ergebnis[x1]>253 then ergebnis[x1]:=254;
  end;
end.

```

```

    end;
end;

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  write(' Relationsfaktor : ');
  readln(rfak);
  write(' Anfangsschleifen : ');
  readln(leer);
  write(' I.-Tiefe : ');
  readln(it);
  grafik360x480;
  assign(f, 'palette.pal');
  reset(f);
  read(f, palette);
  close(f);
  palette.register[254].rot:=45;
  palette.register[254].blau:=45;
  palette.register[254].gruen:=0;
  palette.anzahl:=255;
  palette.start:=0;
  for x:=0 to 254 do
    linie360x480(x,0,x,1,x);
  for x:=255 to maxx-1 do
    linie360x480(x,0,x,1,254);
  setdacregisters(palette);
  afak:=(amax-amin)/maxx;
  for i:=0 to maxx-1 do begin
    berech(amin+i*afak);
    for y:=1 to maxy-1 do setze360x480(i,y,ergebnis[y]);
  end;
  repeat until readkey<>#0;
  altermodus;
end.

```

```

:::::::::::::
FEIGHOC2.PAS
:::::::::::::
program feighoch;
uses crt, dos, tppvga, tppdac;

const maxx=360;
      maxy=480;

type
  tergebnis=array[0..maxy] of word;
  pergebnis=^tergebnis;

var
  palette          : dac all_registers;
  f                : file of dac_all_registers;
  x,y,z,i,leer,it  : word;
  amin,amax,afak,xanf,rfak,xend : real;
  ergebnis        : tergebnis;

procedure berech(a:real);
var
  x1,x0,i,j:word;
  x,f,dx,x3:real;
begin
  dx:=(xend-xanf)/maxy;
  x:=xanf-dx;
  i:=0;
  repeat
    inc(i);
    x:=x+dx;

```

```

    x3:=x;
    for j:=1 to it do x3:=a*x3*(1-x3);
    f:=x3*rfak;
    x0:=round(f)+1;
    if x0>250 then x0:=254;
    ergebnis[i]:=x0;
  until i>=maxy
end;

```

```

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  write(' x end : ');
  readln(xend);
  write(' Relationsfaktor : ');
  readln(rfak);
  write(' Anfangsschleifen : ');
  readln(leer);
  write(' I.-Tiefe : ');
  readln(it);
  grafik360x480;
  assign(f, 'palette.pal');
  reset(f);
  read(f,palette);
  close(f);
  palette.register[254].rot:=45;
  palette.register[254].blau:=45;
  palette.register[254].gruen:=0;
  palette.anzahl:=255;
  palette.start:=0;
  for x:=0 to 254 do
    linie360x480(x,0,x,1,x);
  for x:=255 to maxx-1 do
    linie360x480(x,0,x,1,254);
  setdacregisters(palette);
  afak:=(amax-amin)/maxx;
  for i:=0 to maxx-1 do begin
    berech(amin+i*afak);
    for y:=1 to maxy-1 do
      setze360x480(i,y,ergebnis[y]);
    end;
  repeat until readkey<>#0;
  altermodus;
end.

```

```

::::::::::::::::::
FEIGHOCH.PAS
::::::::::::::::::
program freighoch;
uses crt,dos, tpp256, tppdac;

type tbild=array[1..32767] of word;
   fbild=file of tbild;
   tergebnis=array[0..200] of word;
   pergebnis=^tergebnis;

var bild:tbild absolute $A000:0000;
    palette:dac all registers;
    f:file of dac_all_registers;
    x,y,z,i:word;
    amin,amax,afak,xanf:real;
    ergebnis:tergebnis;

procedure berech(a:real);
var
  x1,x0:word;
  x,f:real;

```

```

begin
  x:=xanf;
  fillchar(ergebnis,sizeof(ergebnis),#0);
  for x1:=1 to 1000 do begin
    x:=a*x*(1-x);
    if x1>100 then inc(ergebnis[round(x*200)]);
  end;
  for x1:=0 to 199 do begin
    f:=ergebnis[x1]*0.2;
    ergebnis[x1]:=round(f);
  end;
end;

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  mode320x200;
  assign(f,'palette.pal');
  reset(f);
  read(f,palette);
  close(f);
  for x:=0 to 255 do
    line320x200(x,0,x,1,x);
  setdacregisters(palette);
  afak:=(amax-amin)/320;
  for i:=0 to 319 do begin
    berech(amin+i*afak);
    for y:=1 to 199 do set320x200(i,y,ergebnis[y]);
  end;
  repeat until readkey<>#0;
  modetext;
end.

:::::::::::::
FEIGH A.PAS
:::::::::::::
program freighoch;
uses crt,dos,tp256,tpdac;

type tbild=array[1..32767] of word;
  fbild=file of tbild;
  tergebnis=array[0..200] of word;
  pergebnis=^tergebnis;

var bild:tbild absolute $A000:0000;
  palette:dac all registers;
  f:file of dac_all_registers;
  x,y,z,i:word;
  amin,amax,afak,xanf:real;
  ergebnis:tergebnis;

procedure berech(a:real);
var
  x1,x0:word;
  x,f:real;
begin
  x:=xanf;
  fillchar(ergebnis,sizeof(ergebnis),#0);
  for x1:=1 to 1000 do begin
    x:=a*x*(1-x);
    if x1>100 then inc(ergebnis[round(x*200)]);
  end;
  for x1:=0 to 199 do begin
    f:=ergebnis[x1]*0.2;
    ergebnis[x1]:=round(f);
  end;
end;
end;

```

```

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  mode320x200;
  assign(f, 'palette.pal');
  reset(f);
  read(f, palette);
  close(f);
  for x:=0 to 255 do
    line320x200(x, 0, x, 1, x);
  setdacregisters(palette);
  afak:=(amax-amin)/320;
  for i:=0 to 319 do begin
    berech(amin+i*afak);
    for y:=1 to 199 do set320x200(i, y, ergebnis[y]);
  end;
  repeat until readkey<>#0;
  modetext;
end.

::::::::::::::::::
FEIGH B.PAS
::::::::::::::::::
program freighoch;
uses crt, dos, tpp256, tppdac;

type tbild=array[1..32767] of word;
   fbild=file of tbild;
   tergebnis=array[0..200] of word;
   pergebnis=^tergebnis;

var bild:tbild absolute $A000:0000;
    palette:dac all registers;
    f:file of dac_all_registers;
    x, y, z, i:word;
    amin, amax, afak, xanf:real;
    ergebnis:tergebnis;

procedure berech(a:real);
var
  x1, x0:word;
  x, f:real;
begin
  x:=xanf;
  fillchar(ergebnis, sizeof(ergebnis), #0);
  for x1:=1 to 1000 do begin
    x:=a*x*(1-x);
    if x1>100 then inc(ergebnis[round(x*200)]);
  end;
  for x1:=0 to 199 do begin
    f:=ergebnis[x1]*0.2;
    ergebnis[x1]:=round(f);
  end;
end;

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  mode320x200;
  assign(f, 'palette.pal');
  reset(f);

```

```

    read(f,palette);
    close(f);
    {for x:=0 to 255 do
      line320x200(x,0,x,1,x);}
    setdacregisters(palette);
    afak:=(amax-amin)/320;
    for i:=0 to 319 do begin
      berech(amin+i*afak);
      for y:=1 to 199 do set320x200(i,y,ergebnis[y]);
    end;
    repeat until readkey<>#0;
    modetext;
end.

::::::::::::::::::
FEIG VGA.PAS
::::::::::::::::::
program feighoch;

uses crt,graph,dos,tpdac;

type
  tergebnis = array[0..480] of word;
  pergebnis = ^tergebnis;

var
  gd,gm                : integer;
  palette              : dac_all_registers;
  f                    : file of _dac_all_registers;
  x,y,z,i,leer,it     : word;
  amin,amax,afak,xanf,rfak,xend : real;
  ergebnis            : tergebnis;

procedure berech(a:real);
var
  x1,x0,i,j:word;
  x,f,dx,x3:real;
begin
  dx:=(xend-xanf)/480;
  x:=xanf-dx;
  i:=0;
  repeat
    inc(i);
    x:=x+dx;
    x3:=x;
    for j:=1 to it do x3:=a*x3*(1-x3);
    f:=x3*rfak;
    x0:=round(f)+1;
    if x0>250 then x0:=254;
    ergebnis[i]:=x0;
  until i>=480
end;

begin
  writeln;
  write(' a min : ');
  readln(amin);
  write(' a max : ');
  readln(amax);
  write(' x anf : ');
  readln(xanf);
  write(' x end : ');
  readln(xend);
  write(' Relationsfaktor : ');
  readln(rfak);
  write(' Anfangsschleifen : ');
  readln(leer);
  write(' I.-Tiefe : ');
  readln(it);
  gd:=detect;
  initgraph(gd,gm,'C:\tp\bgi');
  assign(f,'palette.pal');

```

```

reset(f);
read(f,palette);
close(f);
palette.register[254].rot:=45;
palette.register[254].blau:=45;
palette.register[254].gruen:=0;
palette.anzahl:=255;
palette.start:=0;
for x:=0 to 254 do
  BEGIN
    setcolor(x);
    line(x,0,x,1);
  END;
for x:=255 to 640-1 do
  BEGIN
    setcolor(254);
    line(x,0,x,1);
  END;
setdaregisters(palette);
afak:=(amax-amin)/640;
for i:=0 to 640-1 do begin
  berech(amin+i*afak);
  for y:=1 to 480-1 do
    BEGIN
      putpixel(i,y,ergebnis[y]);
    END;
end;
repeat until readkey<>#0;
closegraph;
end.

```

```

::::::::::::::::::
GRAF.PAS
::::::::::::::::::
{GRAF.PAS}

```

```
{ $X+ }
```

```
program graf ;
```

```
uses crt,graph;
```

```
var
```

```
  gd,gm,x    : integer;
  r,y,x0,y0  : real;
```

```
begin
```

```
  clrscr;
  write('r (z.b. 3bis4) : ');
  readln(r);
  write('x0 (z.b. 0.777) : ');
  readln(x0);
  y0:=0;
```

```
  gd:=detect;
  initgraph(gd,gm,'c:\tp\bgi');
  setbkcolor(white);
```

```
  for x:=0 to 1000 do
    begin
      y:=r*(x/1000)*(1-(x/1000));
      putpixel(round(x*0.48),round(479-y*480),red);
      putpixel(round(x*0.48),round(479-x*0.48),lightgreen);
    end;
```

```
  for x:=1 to 10 do
    begin
      setbkcolor(white);
      setcolor(lightred);
      line(round(x0*480),round(480-y0*480),round(x0*480),round(480-r*x0*(1-x0)*480));
    end;
```

```

    y0:=r*x0*(1-x0);
    line(round(x0*480),round(480-y0*480),round(y0*480),round(480-
    y0*480));
    x0:=y0;
end;

readkey;

end.

::::::::::::::::::
HISTOGRA.PAS
::::::::::::::::::
PROGRAM HistogrammMay;

{Geschrieben von J"rgen Kosche in Turbo Pascal 6.0}
{17.01.1993}

{Erstellt ein Histogramm zur Maykurve}

{Compiler-Befehle}
{$N+}
{$E+}

USES
  CRT, Graph;

CONST
  Max = 500000;      { +++ }
  r = 4;
  x0 = 0.777;
  Faktor = 50;      { f•r skalierung f•r bildschirm }

VAR
  feld : ARRAY[0..479] OF LongInt;

FUNCTION MayGleich (x : Extended) : Extended;
BEGIN
  MayGleich:=x*r*(1-x);
END;

PROCEDURE Init;
  VAR
    gd, gm : Integer;
BEGIN
  DetectGraph (gd, gm);
  InitGraph (gd, gm, 'C:\TP\BGI\');
END;

PROCEDURE Zeichne;
  VAR
    l : Word;
BEGIN
  FOR l:=0 TO 479 DO BEGIN
    MoveTo (0, 479-l);
    LineTo (Round (Faktor*639*feld[l]/Max), 479-l);
  END;
END;

PROCEDURE Histogramm;
  VAR
    l : LongInt;
    x : Extended;
BEGIN
  x:=x0;
  FOR l:=0 TO 479 DO feld[l]:=0;
  FOR l:=0 TO Max DO BEGIN
    x:=MayGleich (x);
    Inc (feld[Round (x*479)]);
  END;
  Zeichne;
END;

```



```

BEGIN
    Init;
    Histogramm;
    REPEAT UNTIL KeyPressed;
    Closegraph;
END.

::::::::::::::::::
ITER.PAS
::::::::::::::::::
PROGRAM iter ;

{$n+}

USES graph,crt ;

VAR
    xn      : extended ;
    x       : extended ;
    a       : extended ;
    b       : extended ;
    xn1     : extended ;
    count   : integer ;

PROCEDURE initunitgraph ;

VAR
    gm,gd : integer ;

BEGIN
    gd:=detect ;
    initgraph(gd, gm, 'C:\TP\BGI') ;
    cleardevice ;
    moveto(0,0) ;
END;

BEGIN

    CLRSCR ;

    write('X = ') ;
    readln(x) ;

    write('Amax ist kleiner als : ') ;
    b:=1/(x-x*x) ;
    writeln(b:3:2);

    write('A = ') ;
    readln(a) ;

    initunitgraph ;

    xn := x ;
    count := 0 ;

    REPEAT
        xn1 := a*xn*(1-xn) ;
        xn := xn1 ;
        IF count > 300 THEN
            BEGIN
                lineto( round((count-300)*6.4) , getmaxy-round(xn*getmaxy) ) ;
            END;
        count := count + 1 ;
    UNTIL count > 400 ;

    readln;
    closegraph;

END.

::::::::::::::::::

```

```

KOCH.PAS
::::::::::::::::::

{$n+}
program koch;
  uses crt,dos,graph;
  var i,rlauf,lauf,b,index,p,schleife:word;
      r:byte;
      gdr,grm,xn,yn,y,x,sarray,max:integer;
      q,g,yg,weiter:char;
      l,w,laufz:longint;

      s:array[1..10000] of char;

begin
repeat
for sarray:=1 to 10000 do begin
  s[sarray] := '0';
end;
write('Anzahl der Iterationen ');readln(i);lauf:=1;
x:=0;y:=475;r:=1;rlauf:=1;s[1] := 'f';schleife:=1;sarray:=0;max:=1;
  for p:=1 to i do begin
    repeat
      q:=s[schleife];
      if q='f' then begin

        for sarray:=max+1 downto schleife+1 do begin
          s[sarray+7] :=s[sarray];
        end;
        schleife:=schleife+1;s[schleife] := '+';
        schleife:=schleife+1;s[schleife] := 'f';
        schleife:=schleife+1;s[schleife] := '-';
        schleife:=schleife+1;s[schleife] := '-';
        schleife:=schleife+1;s[schleife] := 'f';
        schleife:=schleife+1;s[schleife] := '+';
        schleife:=schleife+1;s[schleife] := 'f';
        max:=max+7;
      end;

      schleife:=schleife+1;
    until (q='0');
    schleife:=1;
  end;

detectgraph(gdr,grm);
  initgraph(gdr,grm,'C:\tp\bgi');
  w:=round(exp(i*ln(4))+1);
  l:=round(600/(exp(i*ln(3))));
  laufz:=0;
  repeat
    repeat
      g:=s[rlauf];
      if g='+' then r:=r+1;
      if g='-' then r:=r-1;
      if r=7 then r:=1;
      if r=0 then r:=6;
      rlauf:=rlauf+1;
    until (g='f');
    case r of 1:begin
      xn:=x+1;
      yn:=y;
    end;
    2:
  begin

```

```

    xn:=x+round(1/2);
    yn:=round(y-1*0.5*sqrt(3));
end;
        3:
begin
    xn:=x-round(1/2);
    yn:=y-round(1*0.5*sqrt(3));
end;
        4:
begin
    xn:=x-1;
    yn:=y;
end;

        5:
begin
    xn:=x-round(1/2);
    yn:=y+round(1*0.5*sqrt(3));
end;
        6:
begin
    xn:=x+round(1/2);
    yn:=y+round(1*0.5*sqrt(3));
end;
    end;
    line(x,y,xn,yn);
    x:=xn;y:=yn;laufz:=laufz+1;
until laufz=w;

    yg:=readkey;
    closegraph;

writeln('Weiter?');
read(weiter);
until weiter='n'

```

end.

```

::::::::::::::::::
LORENZ.PAS
::::::::::::::::::
{$X+}

```

```

program lorenz;

```

```

uses crt,graph;

```

```

const

```

```

    anz    = 2      ;
    sigma  = 10     ;
    a      = 28     ;
    dt     = 0.001 ;

```

```

var

```

```

    gd,gm,zahl      : integer;
    x,y,z,xa,ya,za  : array[1..anz] of real;

```

```

begin

```

```

    for zahl:=1 to anz do
    begin
        read(x[zahl]);
        read(y[zahl]);
        read(z[zahl]);
    end;

```

```

    gd:=vga;gm:=vgahi;
    initgraph(gd,gm,'c:\tp\bgi');

```

```

    repeat
        for zahl:=1 to anz do

```

```

begin
  xa[zahl]:=x[zahl];ya[zahl]:=y[zahl];za[zahl]:=z[zahl];
  x[zahl]:=xa[zahl]+sigma*dt*(ya[zahl]-xa[zahl]);
  y[zahl]:=ya[zahl]+dt*(xa[zahl]*(a-za[zahl])-ya[zahl]);
  z[zahl]:=za[zahl]+dt*(xa[zahl]*ya[zahl]-8/3*za[zahl]);
  setcolor(0);
  line(round(10*xa[1])+320,round(10*ya[1])+240,round(10*xa[2])
+320,round(10*ya[2])+240);
  setcolor(15);
  line(round(10*x[1])+320,round(10*y[1])+240,round(10*x[2])
+320,round(10*y[2])+240);
  putpixel(round(10*xa[1])+320,round(10*ya[1])+240,13);
  putpixel(round(10*xa[2])+320,round(10*ya[2])+240,10)
end;
until keypressed;
end.

```

```

:~::~:
LORENZ2.PAS
:~::~:
{$X+}

```

```

program lorenz;

```

```

uses crt,graph;

```

```

const

```

```

  anz   = 2      ;
  sigma = {10}0.1 ;
  a     = 28     ;
  dt    = 0.001 ;
  sch   = 1      ;
  an    = {30000} 1000;

```

```

var

```

```

  gd,gm,h,i,j,k   : integer;
  x,y,z,xa,ya,za  : array[1..anz] of real;
  exp,dx,dy,dz    : real;
  s               : string;

```

```

begin

```

```

  clrscr;
  write('Startkoordinate (x) : ');
  readln(x[1]);
  write('Startkoordinate (y) : ');
  readln(y[1]);
  write('Startkoordinate (z) : ');
  readln(z[1]);
  x[2]:=x[1]+0.01;y[2]:=y[1];z[2]:=z[1];
  clrscr;
  for h:=1 to 3 do
  for i:=1 to an do begin
    for j:=1 to sch do begin
      for k:=1 to 2 do begin
        xa[k]:=x[k];ya[k]:=y[k];za[k]:=z[k];
        x[k]:=xa[k]+sigma*dt*(ya[k]-xa[k]);
        y[k]:=ya[k]+dt*(xa[k]*(a-za[k])-ya[k]);
        z[k]:=za[k]+dt*(xa[k]*ya[k]-8/3*za[k]);
      end;
    end;
    dx:=x[2]-x[1];dy:=y[2]-y[1];dz:=z[2]-z[1];
    exp:=exp+100*sqrt(dx*dx+dy*dy+dz*dz)/sch;
    x[2]:=x[1]+0.01;y[2]:=y[1];z[2]:=z[1];
    if i mod 100=0 then begin gotoxy(0,0);writeln(i*h);end;
  end;
  exp:=ln(exp)/(an*dt);
  gotoxy(10,10);write(exp:9:6);
  readkey;
end.

```

```

:~::~:
LYAPUNOW.PAS

```

```

:.....:
{$n+}

PROGRAM LYAPUNOV_EXPONENT;

uses dos,crt,graph ;

var aanf,a,aend,da,x0,x00,s,f,lamp:extended;
    l,n,nend,gdr,grm,xalt,yalt,xgraph,ygraph:integer;

BEGIN
  write('Aanf(3.5)=');
  readln(aanf);
  write('deltaA(0.0005)=');
  readln(da);
  write('Aend(4)=');
  readln(aend);
  write('Xanf(egal(0.3))=');
  readln(x00);
  write('Summationen(hoch(500)):');
  readln(nend);
  detectgraph(gdr,grm);
  initgraph(gdr,grm,'C:\tp\bgi');
  line(0,240,640,240);
  xgraph:=0; ygraph:=0;a:=aanf;
  REPEAT
    f:=0;n:=0;s:=0;x0:=x00;xalt:=xgraph;yalt:=ygraph;
    FOR l:=1 TO 300 DO
      x0:=a*x0*(1-x0);
    REPEAT
      x0:=a*x0*(1-x0);
      f:=abs(a*(1-2*x0));
      s:=s+ln(f);
      n:=n+1;
    UNTIL (n=nend);
    lamp:=(s/n/ln(2))/ln(2);
    {lampgraph:=160+round(lamp);agraph:=round(da);
    setze360x480(agraph,lampgraph,50);}
    xgraph:=round((a-aanf)/da*640*da/(aend-aanf));
    ygraph:=240-round(160*lamp);
    line(xalt,yalt,xgraph,ygraph);
    a:=a+da;
    {writeln('A:',a,' Lambda=',lamp);}
  UNTIL (a>=aend-da);
  readln;
  closegraph;
END.

```

```

:.....:
NEWTON.PAS
:.....:
PROGRAM Zweidimensionales_Newtonverfahren;

USES crt,printer,graph;

{
geg:  $x^2 + y^2 = 4$ 
       $y = 3 * \sin(2*x)$ 
====>  $f(x,y) = x^2 - y^2 = 0$ 
       $g(x,y) = y - 3 * \sin(2*x) = 0$ 

====>  $fx = 2x$ 
       $fy = 2y$ 
       $gx = -6 * \cos(2x)$ 
       $gy = 1$ 

====> mit (5.86)
       $x[k+1] = x[k] + (y^2-6y*\sin(2x)-x^2+4)/(2x+12y*\cos(2x))$ 
       $y[k+1] = y[k] + (6x*\sin(2x) - 2xy - 6*\cos(2x)*(x^2+y^2-4))$ 
       $/ (2x+12y*\cos(2x))$ 
}

```

```

VAR    x          :real ;
       y          :real ;
       i          :integer ;
       d          :integer ;
       m          :integer ;
       maxx      :integer ;
       maxy       :integer ;
       s          :string ;
       x1         :array[1..6]of real ;
       y1         :array[1..6]of real ;

PROCEDURE iteration(var a,b:real);{Annäherung der Lösungen}
  var x,y:real;
  begin
    repeat
      x:=a;y:=b;
      a:= x+(y*y-6*y*sin(2*x)-x*x+4)/(2*x+12*y*cos(2*x));
      b:= y+(6*x*sin(2*x)-2*x*y-6*cos(2*x)*(x*x+y*y-4))/(2*x+12*y*cos(2*x));
    until (abs(a-x)<1e-1) and (abs(b-y)<1e-1);
    {Abbruch, wenn sich die Werte um weniger als 0,1 ändern}
  end;

PROCEDURE iteration1(var a,b:real);
  {Berechnung der speziellen Iterationsfolgen}
  var x,y:real;
  begin
    x:=a;y:=b;
    a:= x+(y*y-6*y*sin(2*x)-x*x+4)/(2*x+12*y*cos(2*x));
    b:=y+(6*x*sin(2*x)-2*x*y-6*cos(2*x)*(x*x+y*y-4))/(2*x+12*y*cos(2*x));
  end;

FUNCTION farbe(x:real):word;
  begin
    case trunc(3*x) of
      -5:farbe:=6;  {braun}
      -3:farbe:=7;  {hellgrau}
      -1:farbe:=0;  {schwarz}
       5:farbe:=12; {hellrot}
       3:farbe:=15; {weiß}
       1:farbe:=14; {gelb}
    end;
  end;

BEGIN

  detectgraph(d,m);
  initgraph(d,m,'c:\tp\bgi');

  maxx:=300;           {maximale Pixelanzahl}
  maxy:=300;           {in x- und y- Richtung}

  for d:= 0 to maxx do      {erstellt die Grafik}
    for m:= 0 to maxy do
      begin
        x:=-1+d*6/maxx;
        y:=-1+m*26/maxy;
        iteration(x,y);
        putpixel(d,maxy-m, farbe(x));
      end;
    setcolor(1);           {Koordinatensystem}
    line(0,maxy-trunc(maxy/26),maxx,maxy-trunc(maxy/26));
    line(trunc(maxx/6),0,trunc(maxx/6),maxy);
    moveto(maxx,maxy-trunc(maxy/26));
    lineto(maxx-10,maxy+4-trunc(maxy/26));
    moveto(maxx,maxy-trunc(maxy/26));
    lineto(maxx-10,maxy-4-trunc(maxy/26));
    moveto(trunc(maxx/6),0);
    lineto(trunc(maxx/6)-4,10);
    moveto(trunc(maxx/6),0);
    lineto(trunc(maxx/6)+4,10);
  for d:=0 to 5 do
    line(trunc(d*maxx/6),maxy+4-trunc(maxy/26),trunc(d*maxx/6),maxy-4-

```

```

    trunc(maxy/26));
  for m:=0 to 25 do
    line(trunc(maxx/6)-4,maxy-trunc(m*maxy/26),trunc(maxx/6)+4,maxy-
      trunc(m*maxy/26));
  outtextxy(trunc(maxx/6)-10,maxy-trunc(maxy/26)+5,'0');
  outtextxy(trunc(2*maxx/6),maxy-trunc(maxy/26)+5,'1');
  outtextxy(trunc(3*maxx/6),maxy-trunc(maxy/26)+5,'2');
  outtextxy(trunc(4*maxx/6),maxy-trunc(maxy/26)+5,'3');
  outtextxy(trunc(5*maxx/6),maxy-trunc(maxy/26)+5,'4');
  outtextxy(maxx-10,maxy-trunc(maxy/26)+5,'x');
  outtextxy(trunc(maxx/6)-14,maxy-trunc(6*maxy/26)-4,'5');
  outtextxy(trunc(maxx/6)-22,maxy-trunc(11*maxy/26)-4,'10');
  outtextxy(trunc(maxx/6)-22,maxy-trunc(16*maxy/26)-4,'15');
  outtextxy(trunc(maxx/6)-22,maxy-trunc(21*maxy/26)-4,'20');
  outtextxy(trunc(maxx/6)-14,4,'y');
  {Anfangswerte f•r Iterationsbeispiele
  Nicht alle Werte sind geeignet, weil nicht alle L"sungen im Intervall
  liegen
  bzw. die Iterationsfolge zeitweilig das Intervall verl„ät}
  x1[1]:=0.5;x1[2]:=0.55; x1[3]:=1.5; x1[4]:=1.55; x1[5]:=1.8;x1[6]:=1.8;
  y1[1]:=4;y1[2]:=4;y1[3]:=19; y1[4]:=19; y1[5]:=-0.9;y1[6]:=-1;
  for i:=1 to 6 do {berechnet Werte und zeichnet Linien}
    begin
      outtextxy(trunc(maxx*(x1[i]+1)/6),maxy-trunc(maxy*(y1[i]+1)
        /26)-5,'S');
      moveto(trunc(maxx*(x1[i]+1)/6),maxy-trunc(maxy*(y1[i]+1)/26));
      repeat
        x:=x1[i];y:=y1[i];
        iteration1(x1[i],y1[i]);
        lineto(trunc(maxx*(x1[i]+1)/6),maxy-trunc(maxy*(y1[i]+1)/26));
      until (abs(x-x1[i])<1e-2) and (abs(y-y1[i])<1e-2);
    end;
  readln;
  closegraph;
END.

::::::::::::::::::
TPP256.PAS
::::::::::::::::::
unit tpp256;
{ -----
                                tpp256.pas
                                -----
Pascal-Unit f•r den Zugriff auf den
320x200-Modus in 256 Farben (Modus 13)

von Martin Althaus 3/90
-----
}
interface
procedure mode320x200; {Setzt den 320x200-Modus}
procedure modetext;   {Zur•ck in den Textmodus}
procedure set320x200(x,y:integer;farbe:byte);
                                {Setzt einen Punkt}
procedure line320x200(x1,y1,x2,y2:integer;
                                farbe:byte);
                                {Zeichnet eine Linie}
function read320x200(x,y:integer):byte;
                                {Liest einen Punkt}
implementation
uses dos;                       {f•r die Register-Def.}
var
cpu: registers;
modus: byte;
benutzt: boolean;
                                {Zeigt an, ob MODE320X200 benutzt}

procedure mode320x200;
{-----
Setze die VFA/MCGA-Karte Modus 13h mit 320x200
Punkten in 256 Farben.
-----}

```

```

}
begin
  cpu.ah:=$0f;           {Lese Videomodus aus}
  intr($10,cpu);        {•ber Int 10hex}
  modus:=cpu.al;        {Merke den Modus}
  benutzt:=true;       {zeige gemerkten Modus an}
  cpu.ah:=0;           {Setze Modus 13hex •ber}
  cpu.al:=$13;         {Funktion 00hex von Int. 10hex}
  intr($10,cpu);
end;

procedure modetext;
{-----}
  Kehre zur•ck in den Textmodus, der vor dem Setzen
  mit MODE320X200 aktiv war.
{-----}
}
begin
  if benutzt then      {Ist ein Modus gemerkt?}
  begin                {Ja, setze ihn}
    cpu.ah:=0;         {Benutze Funktion 00hex}
    cpu.al:=modus;    {von Int. 10hex}
    intr($10,cpu);
  end;
end;

procedure set320x200(x,y:integer;farbe:byte);
{-----}
  Setze einen Punkt im 320x200-Modus an der Stelle
  X,Y und gebe ihm die Farbe FARBE
{-----}
}
var
  offset: word;
begin                {X-Koordinate g•ltig?}
  if x<0 then x:=0
    else if x>319 then x:=319;
                {Y-Koordinaten g•ltig}
  if y<0 then y:=0
    else if y>199 then y:=199;
  offset:=320*y+x;   {Berechne Punkt-Adresse}
  mem[$a000:offset]:=farbe; {Und setze Punkt}
end;

function read320x200(x,y:integer):byte;
{-----}
  Lese einen Punkt aus dem Bildschirmspeicher aus
  und liefere seine Farbe als Funktionsergebnis
  zur•ck.
{-----}
}
var
  offset: word;
begin                {justiere X-Koordinate}
  if x<0 then x:=0
    else if x>319 then x:=319;
                {justiere Y-Koordinate}
  if y<0 then y:=0
    else if y>199 then y:=199;
  offset:=320*y+x;   {Berechne Punkt-Adresse}
  read320x200:=mem[$a000:offset]; {Lese Punkt aus}
end;

procedure tausche(x,y:integer);
{-----}
  Interne Prozedur zum Tauschen von zwei Integers
{-----}
}
var
  help:integer;      {Hilfsvariable}
begin
  help:=x;           {X merken}

```



```

x:=y;           {Y wird X}
y:=help;       {X wird Y}
end;

procedure line320x200(x1,y1,x2,y2:integer;
                     farbe:byte);
{-----}
Zeichnet eine Linie im 320x200-Modus vom
Anfangspunkt X1,Y1 zum Endpunkt X2,Y2 und gibt
ihr die Farbe FARBE
{-----}
}
var
  y: integer;           {f•r das Funktionsergebnis}
  m,b: real;           {Steigung und Achsenabschnitt}

begin
  if x2<x1 then        {M•ssen die Koordianten ge-}
    begin              {tauscht werden?}
      tausche(x1,x2);   {ja, also tausche sie}
      tausche(y1,y2);   {beide aus}
    end;
  if x2<>x1 then        {kann man Geradengleichung}
    begin              {anwenden?}
      m:=(y2-y1)/(x2-x1); {ja, berechne M}
      b:=y1-(m*x1);      {und B}
      for x1:=x1 to x2 do {Berechne in einer}
        begin           {Schleife zu jedem}
          y:=trunc(m*x1+b); {X ein Y und}
          set320x200(x1,y,farbe); {zeichne den Punkt}
        end;           {Schleife schlie•en}
      end;
    else                {Es ist eine Senkrechte}
      begin
        if y2<y1 then tausche(y1,y2);
        for y1:=y1 to y2 do {setze in einer Schleife}
          set320x200(x1,y1,farbe); {die Senkrechte}
        end;
      end;
    end;

begin
{-----}
  Initialisierung
{-----}
}
benutzt:=false;      {MODE320X200 ist unbenutzt}
end.

:::
TPPDAC.PAS
:::
unit tppdac;
{-----}
                               tppdac.pas
                               -----}

Unit zur vollen Kontrolle der Farbgebung in den
Videomodi mit 256 Farben der EGA- und VGA-Karte

von Martin Althaus 5/90
{-----}
}
interface
{-----}
  Neue Datentypen der Unit
{-----}
}
type
  dac_one_register =
    record
      rot,gruen,blau:byte;
    end;
  dac_all_registers =

```

```

    record
      register: array[0..255] of dac_one_register;
      anzahl, start: byte;
    end;
  {-----
  Konstanten f•r Fehlermeldungen
  -----}
const
  dac ok = 0;
  dac_no_vga = 1;
  dac register wrong = 2;
  {-----
  Variable f•r Fehlerabfrage
  -----}
}
var
  dac status: byte;
  {-----
  Funktionen und Prozeduren
  -----}
function getdacstatus:byte;      {Status ermitteln}
procedure setdacregister(nr: byte; reg:
                          dac one register);
      {Ein einzelnen DAC-Register setzen}
procedure setdacregisters(reg:dac all registers);
      {Eine Reihe von DAC-Registern setzen}
procedure getdacregister(nr:byte; var reg:
                          dac one register);
      {Ein einzelnes AC-Register lesen}
procedure getdacregisters(var reg:
                          dac_all registers);
      {Eine Reihe von DAC-Registern lesen}
function getdacmask:byte;
      {DAC-Maskierungs-Register lesen}
procedure setdacmask(mask:byte);
      {DAC-Maskierungs-Register setzen}
  {-----
  Einbinden der neuen Routinen
  -----}
}
implementation
  uses dos;          {F•r Register-Variable CPU}
var
  cpu:registers;
  co:byte;
  hauptreg:dac_all_registers;

function getdacstatus:byte;
  {-----
  Ermittelt den Status der Unit TPPDAC. Als Return-
  Werte sind die als Konstanten definierten Werte
  m"glich.
  -----}
}
begin
  getdacstatus:=dac_status;      {Status •bertragen}
end;

procedure setdacregister(nr:byte; reg:
                          dac one register);
  {-----
  Gibt einem einzelnen Register in der DAC-Palette
  einen neuen Wert. Der •bergebene Parameter ist
  vom Datentyp DAC ONE REGISTER.
  -----}
}
begin
  cpu.ah:=$10;          {Funktion und Unterfunktion}
  cpu.al:=$10;          {ausw,,hlen}
  cpu.bx:=nr;

```

```

cpu.cl:=reg.blau;           {Farbanteile •bertragen}
cpu.ch:=reg.gruen;
cpu.dh:=reg.rot;
intr($10,cpu);           {Interrupt aufrufen}

end;

procedure setdacregisters(reg:dac all registers);
{-----
Setze mehrere DAC-Register in der DAC-
Farbpalette. Die Register werden mit der Daten-
struktur DAC ALL REGISTERS •bergeben.
-----}
begin
cpu.ah:=$10;           {Funktion und Unterfunktion}
cpu.al:=$12;           {ausw,,hlen}
cpu.bx:=reg.start;
cpu.cx:=reg.anzahl;
cpu.es:=seg(reg.register[reg.start]);
                        {Startadresse der}
cpu.dx:=ofs(reg.register[reg.start]); {Tabelle}
intr($10,cpu);
end;

procedure getdacregister(nr:byte;var reg:
                        dac one register);
{-----
Lies den Inhalt eines einzelnen DAC-Registers
aus. Der •bergebene Wert ist mit dem Datentyp
DAC ONE REGISTER definiert.
-----}
begin
cpu.ah:=$10;           {Funktion und Unterfunktion}
cpu.al:=$15;           {ausw,,hlen}
cpu.bx:=nr;
intr($10,cpu);
reg.rot:=cpu.dh;       {Farbanteile •ber-}
reg.blau:=cpu.cl;      {tragen}
reg.gruen:=cpu.ch;
end;

procedure getdacregisters(var reg:
                        dac all registers);
{-----
Lies eine Reihe von DAC-Registern aus der Palette
aus. Die Register-Inhalte werden in einer Daten-
struktur des Typs DAC ALL REGISTER gespeichert.
-----}
begin
cpu.ah:=$10;           {Funktion und Unterfunktion}
cpu.al:=$17;           {ausw,,hlen}
cpu.bx:=reg.start;
cpu.cx:=reg.anzahl;
cpu.es:=seg(reg.register[reg.start]);
                        {Startadresse der}
cpu.dx:=ofs(reg.register[reg.start]); {Tabelle}
intr($10,cpu);
end;

function getdacmask:byte;
{-----
Ermittle den Inhalt des DAC-Maskierungs-Registers
-----}
begin
cpu.ah:=$10;           {Funktion und Unterfunktion}
cpu.al:=$19;           {ausw,,hlen}
intr($19,cpu);
getdacmask:=cpu.bl;   {Maskierung •bergeben}

```

```

end;

procedure setdacmask(mask:byte);
{-----
  Setze das DAC-Maskierungs-Register
-----}
begin
  cpu.ah:=$10;           {Funktion und Unterfunktion}
  cpu.al:=$18;           {ausw„hlen}
  cpu.bl:=mask;         {Maskierung setzen}
  intr($10,cpu);
end;

{-----
  Initialisierungsteil der Unit
-----}
begin
  cpu.ah:=$1a;           {šber Funktion 1Ah testen,}
  cpu.al:=0;             {ob eine MCGA- oder VGA-}
  intr($10,cpu);
  if (cpu.al=$1a)       {Karte installiert ist}
  then
    begin
      dac_status:=dac_no_vga; {Funktion ist unterst•tzt}
      {R•ckgabe: nicht bekannt}
      case cpu.bl of
        $07,$08,$0a,$0b,$0c: dac_status:=dac_ok;
          {R•ckgabe: VGA oder MCGA}
      end;
    end;
  if dac_status=dac_ok then
    begin
      {DAC-Funktionen unterst•tzt?}
      {ja}
      hauptreg.anzahl:=255; {alle 255 Register ab}
      hauptreg.start:=0;   {Register 0}
      getdacregisters(hauptreg); {auslesen}
    end;
end.

:::
TPPVGA.PAS
:::
unit tppvga;
{-----
                                tppvga.pas
-----}

Pascal-Unit zum Zugriff auf zwei neue Videomodi
der VGA-Karte:

    320x400 Punkte in 256 Farben
    360x480 Punkte in 256 Farben

von Martin Althaus 2/90

Vor Benutzung der Routinen muá die Boolean-
Variable VGA OK abgefragt werden. Nur wenn sie
den Wert TRUE besitzt, ist eine VGA-Karte
installiert.
-----}
interface

var
  vga ok:boolean;           {Fehlervariable}
{-----}
  Routinen f•r den 320x400-Modus in 256 Farben
-----}
procedure grafik320x400;    {Modus 320x400 setzen}
procedure wahl320x400(seite:byte);

```

```

    {aktive Seite im Modus 320x400 w,,hlen}
procedure zeige320x400(seite:byte);
    {Seite im Modus 320x400 anzeigen}
procedure setze320x400(x,y:integer;farbe:byte);
    {Einen Punkt im Modus 320x400 setzen}
function lese320x400(x,y:integer):byte;
    {Einen Punkt im Modus 320x400 auslesen}
procedure linie320x400(x1,y1,x2,y2:word;farbe:
    byte);
    {Eine Linie im 320x400-Modus zeichnen}
{-----}
Routinen f•r den 360x480-Modus in 256 Farben
-----
}
procedure grafik360x480;    {Modus 360x480 setzen}
procedure setze360x480(x,y:integer;farbe:byte);
    {Einen Punkt im Modus 360x480 setzen}
function lese360x480(x,y:integer):byte;
    {Einen Punkt im Modus 360x480 auslesen}
procedure linie360x480(x1,y1,x2,y2:word;farbe:
    byte);
    {Eine Linie im 360x480-Modus zeichnen}
{-----}
Gemeinsame Routine
-----
}
procedure altermodus;    {alten Modus zur•cksetzen}

implementation

uses dos;                {Benutzt Interrupt-Aufrufe}

var
    benutzt:boolean;
    {Zeigt an, ob ALTMODUS g•ltigen Modus enth,,lt}
    cpu:registers;        {Zugriff auf CPU-Register}
    ramseg:word;          {Startadresse des VGA-RAMS}
    altmodus,index,zseite,sseite:byte;

{-----}
Den Objekt-Code des Moduls in Maschinensprache
einbinden
-----
}

{$! c:\asm\tpplinie}

procedure grafik320x400;
{-----}
Setzt den Videomodus mit 320x400 Punkten in 256
Farben. Dazu werden Register der VGA-Karte direkt
programmiert
-----
}
var
    lese:byte;
    zaehler:word;
begin
    cpu.ah:=$0f;          {Videomodus •ber das}
    intr($10,cpu);
    {BIOS auslesen und in der Variablen}
    altmodus:=cpu.al;     {ALTMODUS speichern}
    benutzt:=true;       {Flag setzen: ALTMODUS g•ltig}
    cpu.ah:=0;           {Videomodus 13hex}
    cpu.al:=$13;         {•ber das BIOS}
    intr($10,cpu);       {setzen}
    {Nun werden die Register ge,,ndert}
    port[$3c4]:=4;
    lese:=(port[$3c5] and (255-8)) or 4;
    port[$3c5]:=lese;
    port[$3ce]:=5;
    lese:=(port[$3cf] and (255-$10));

```

```

port[$3cf]:=lese;
port[$3ce]:=6;
lese:=(port[$3cf] and (255-2));
port[$3cf]:=lese;
port[$3c4]:=2;
port[$3c5]:=$0f;
for zaehler:=0 to $ffff do mem[$0a000:
                                zaehler]:=0;
                                {Bildschirmspeicher 1"schon}
port[$3d4]:=9;
lese:=(port[$3d5] and (255-$1f));
port[$3d5]:=lese;
port[$3d4]:=$14;
lese:=(port[$3d5] and (255-$40));
port[$3d5]:=lese;
port[$3d4]:=$17;
lese:=(port[$3d5] or $40);
port[$3d5]:=lese;
wahl320x400(0);                                {Seite 0 adressiert}
zeige320x400(0);                                {und dargestellt}
end;

procedure wahl320x400(seite:byte); external;
{-----
W,,hlt die Bildschirmseite aus, in der alle
Befehle zum Zeichnen und Lesen von Punkten usw.
durchgef•hrt werden. Als Seiten sind 0 und 1
erlaubt. Die Prozedur ist im Maschinensprache-
Modul deklariert.
-----
}

procedure zeige320x400(seite:byte);
{-----
W,,hlt eine der beiden Bildschirmseiten zur
Darstellung aus. Die gew,,hlte Seite muá nicht die
sein, in der die Zeichenbefehle ausgef•hrt
werden. Als Seiten sind 0 und 1 erlaubt.
-----
}
var
  start:byte;                                {Nimmt Startadresse auf}
begin
  if seite=0 then start:=0
    else start:=$80;
    {Lege Startadresse fest}
  port[$3d4]:=$0c;                            {Darstellungsoffset}
  port[$3d5]:=start;    {Setze Startadresse im CRTC}
end;

procedure setze320x400(x,y:integer;farbe:
                      byte); external;
{-----
Setzt einen Punkt im Bildschirmspeicher an den
Koordinaten X,Y in der aktiven Seite und gibt ihm
die die Farbe FARBE. Die Prozedur ist in
Maschinensprache kodiert.
-----
}

function lese320x400(x,y:integer):byte; external;
{-----
Liest einen Punkt mit den Koordinaten X,Y aus der
aktiven Seite aus und •bergibt seine Farbe als
Byte-Funktionsergebnis. Die Funktion ist in
Maschinensprache kodiert.
-----
}

procedure linie320x400(x1,y1,x2,y2:word;farbe:
                      byte); external;
{-----

```

Zeichnet eine Linie im Modus 320x400 vom Anfangspunkt X1, Y1 zum Endpunkt X2, Y2. Die Linie wird in der Farbe FARBE gezeichnet. Die Routine ist im Maschinensprache-Modul formuliert.

```

}
-----
procedure grafik360x480;
{-----
  Setzt den Videomodus mit 360x480 Punkten in 256
  Farben. Dazu werden Register der VGA-Karte direkt
  programmiert
  -----}
const
  reg wert: array[0..$17] of byte=
    ($6b,$59,$5a,$8e,$5e,$8a,$0d,$3e,$ff,$40,$ff,
     $ff,$ff,$ff,$ff,$ff,$ea,$ac,$df,$2d,$00,$e7,
     $06,$e3);          {Werte f•r die CRTC-Register}
var
  lese,zaehler:byte;
begin
  cpu.ah:=$0f;          {Den aktuellen Videomodus}
  intr($10,cpu);       {auslesen und in der Variablen}
  altmodus:=cpu.al;    {ALTMODUS speichern}
  benutzt:=true;      {Flag setzen: Funktion benutzt}
  cpu.ah:=0;           {VGA-RAM •ber}
  cpu.al:=$12;         {Videomodus 12hex}
  intr($10,cpu);       {weitm"glichst 1"schen}
  cpu.ah:=0;          {Modus 13hex(320x200, 256 Farben)}
  cpu.al:=$13;        {•ber das BIOS}
  intr($10,cpu);       {setzen}
  {Nun werden die Register ge„ndert}

  port[$3c4]:=4;
  port[$3c5]:=6;          {Bit-Planes}
  port[$3c4]:=0;
  port[$3c5]:=1;         {Reset, Kathodenstrahl aus}
  port[$3c2]:=$e7;      {28 MHz Taktgenerator einschalten}

  port[$3c4]:=0;
  port[$3c5]:=3;         {Reset, Kathodenstrahl an}
  port[$3d4]:=$11;      {CRTC R/O-Bit adressieren}
  lese:=(port[$3d5] and $7f);
  port[$3d5]:=lese;     {CRTC-Register werden schreibbar}
  for zaehler:=0 to $17 do
  begin
    {Die wichtigsten}
    if reg_wert[zaehler]<>$ff then
    begin
      {CRTC-Register in}
      port[$3d4]:=zaehler;          {einer}
      port[$3d5]:=reg_wert[zaehler];
    end;
    {Schleife „ndern}
  end;
end;

procedure altermodus;
{-----
  Setzt den vor Einschalten eines der beiden
  Grafikmodi aktiven Videomodus wieder zur•ck
  -----}
begin
  if benutzt then {Ist ein Modus in der Variablen}
  begin
    {ALTMODUS gespeichert?}
    cpu.ah:=0;          {ja, setze}
    cpu.al:=altmodus;  {ihn •ber}
    intr($10,cpu);     {das BIOS}
  end;
end;

procedure setze360x480(x,y:integer;farbe:
  byte); external;

```

```

{-----
Setzt einen Punkt im Bildschirmspeicher an den
Koordinaten X,Y in der aktiven Seite und gibt ihm
die die Farbe FARBE.
-----
}

function lese360x480(x,y:integer):byte; external;
{-----
Liest einen Punkt mit den Koordinaten X,Y aus der
aktiven Seite aus und •ber gibt seine Farbe als
Byte-Funktionsergebnis.
-----
}

procedure linie360x480(x1,y1,x2,y2:word; farbe:
                    byte); external;
{-----
Zeichnet eine Linie im Modus 320x400 vom Anfangs-
punkt X1, Y1 zum Endpunkt X2, Y2. Die Linie wird
in der Farbe FARBE gezeichnet. Die Routine ist im
Maschinensprache-Modul formuliert.
-----
}

{-----
Initialisierungsteil der Unit. Hier wird die
Variable VGA OK gesetzt, mit der vo Aufrufen fest-
gestellt werden muá, ob die Funktionen •berhaupt
ansprechbar sind.
-----
}
begin
  cpu.ah:=$1a;           {Ist eine VGA-Karte}
  cpu.al:=0;             {am System angeschlossen?}
  intr($10,cpu);        {Teste •ber das VGA-BIOS}
  if (cpu.al=$1a) and ((cpu.bl=7) or (cpu.bl=8))
  then vga ok:=true     {ja, zeige es an}
  else vga_ok:=false;   {nein, zeige es an}
end.

```



## 11. IDL-Texte

```

:.....:
Plot
:.....:
; Plot of the tracking data
close,1
set plot,'X'
n_sum=10
;
pp=0
ccc=''
fnl=''
fnl old=''
!p.multi=[0,1,1]
st=''
sx='Time (sec)'
sy='Radio flux (sfu)'
f:
read,'File name: ',fnl
columns = 1
;read,'Number of columns: ',columns
read,'Title (91032308) ',st
dt=0.05 ;sample time
read,'sample time (0.05) : ',dt
if n elements(ppt) eq 0 then ppt=0
openr,1,fnl,error=ex ; Datei zum lesen oeffnen
numb = long(0)
while not eof(1) do begin
readf,1,dummy ; count of input row number
numb = numb + 1
endwhile
close,1
;
f = fltarr(columns)
print,numb
print,'total time',float(numb)*dt
openr,1,fnl,error=ex ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
if numb le 0 then goto, raus
x=fltarr(numb) ; Vektor x als Feld (0..ix-1) vereinbaren
y=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
z=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
for i=long(0),numb-1 do begin
readf,1,f ; read the current row
x(i) = i
y(i) = f(0)
endfor
print,'Radio flux (integer): min',min(y),' max',max(y)
;
;goto, 1100
case fnl of
'7' : begin
zero = -900 ; mV
on_off=745 ; mV
sfu = 35.36 ; mV
end
'8': begin
zero = -900 ; mV
on off=745 ; mV
sfu = 35.36 ; mV
end
'10': begin
zero = -900 ; mV
on off=745 ; mV
sfu = 35.36 ; mV
end
'16': begin
zero = -4200 ; mV
on off = 1550 ; mV
sfu = 73.56 ; mV
end
'17': begin

```

```

        zero      = -4200 ; mV
        on off    = 1550 ; mV
        sfu       = 73.56 ; mV
    end
'20': begin
    zero      = -4000 ; mV
    on off    = 4550 ; mV
    sfu       =215.95 ; mV
end
endcase
mV=3.2768
print,'on off',on_off,'mV , 1sfu=',sfu,'mV'
;y=y/mV+on off      ; integer to mV
y=y/mV+zero        ; integer to mV
y=y/sfu            ; mV to sfu
y=y+200
;y=alog10(y)
l100:
x=x*dt
;
xa=0
xe=0
ya=0
ye=0
x:
if fn1 eq fn1_old then read,'xrange: xa xe ',xa,xe
goto ,z
y:
if fn1 eq fn1_old then read,'yrange: ya ye ',ya,ye
z:
if xa eq 0 and xe eq 0 and ya eq 0 and ye eq 0 then begin
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
endif
print,'min',ya,' max',ye
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,yticks=10,nsum=
n sum
close,1
;
fn1 old=fn1
if pp eq 0 then begin
    while ccc ne 'q' do begin
        print,'                x change the xrange'
        print,'                y change the yrange'
        print,'                f change the data file'
        print,'                p prints current picture'
        print,'                q leaves procedure'
        ccc=get kbrd(1)
        if ccc eq 'p' then begin
            pp=1
            goto,PRINT
        endif
        if ccc eq 'x' then goto, x
        if ccc eq 'y' then goto, y
        if ccc eq 'f' then goto, f
    endwhile
endif

PRINT:
if pp eq 1 then begin
print,'printing picture'

set plot,'ps'
;device, /landscape
devfn=getenv('USER')+fn1
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn

plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17',
xtitle='!17'+sx ytitle='!17'+sy /xstyle /ystyle xticks=10 yticks=10 nsum=

```

```

xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,yticks=10,nsum=
n_sum

idltime=systemtime()
;xyouts,-100,-90,'!17USER: '+getenv('USER')+ ' FILE:'+' '+fn1+' .!17 grf by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close
set plot,'X'
;spawn,'lpr -Plps '+devfn
;spawn,'rm '+devfn
endif

raus:
end
::::::::::::::::::
Plot
::::::::::::::::::
; Plot of the tracking data
close,1
set plot,'X'
n_sum=10
;
pp=0
ccc=''
fn1=''
fn1 old=''
!p.multi=[0,1,1]
st=''
sx='Time (sec)'
sy='Radio flux (sfu)'
f:
read,'File name: ',fn1
columns = 1
;read,'Number of columns: ',columns
read,'Title (91032308) ',st
dt=0.05 ;sample time
read,'sample time (0.05) : ',dt
if n elements(ppt) eq 0 then ppt=0
openr,1,fn1,error=ex ; Datei zum lesen oeffnen
numb = long(0)
while not eof(1) do begin
readf,1,dummy ; count of input row number
numb = numb + 1
endwhile
close,1
;
f = fltarr(columns)
print,numb
print,'total time',float(numb)*dt
openr,1,fn1,error=ex ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
if numb le 0 then goto, raus
x=fltarr(numb) ; Vektor x als Feld (0..ix-1) vereinbaren
y=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
z=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
for i=long(0),numb-1 do begin
readf,1,f ; read the current row
x(i) = i
y(i) = f(0)
endfor
print,'Radio flux (integer): min',min(y),' max',max(y)
;
;goto, 1100
case fn1 of
'7' : begin
zero = -900 ; mV
on_off=745 ; mV
sfu = 35.36 ; mV
end
'8': begin
zero = -900 ; mV

```

```

        on off=745      ; mV
        sfu      = 35.36 ; mV
    end
'10': begin
    zero      = -900  ; mV
    on off=745      ; mV
    sfu      = 35.36 ; mV
    end
'16': begin
    zero      = -4200 ; mV
    on_off    = 1550  ; mV
    sfu      = 73.56 ; mV
    end
'17': begin
    zero      = -4200 ; mV
    on off    = 1550  ; mV
    sfu      = 73.56 ; mV
    end
'20': begin
    zero      = -4000 ; mV
    on off    = 4550  ; mV
    sfu      =215.95 ; mV
    end
endcase
mV=3.2768
print,'on off',on_off,'mV , 1sfu=',sfu,'mV'
;y=y/mV+on off      ; integer to mV
y=y/mV+zero        ; integer to mV
y=y/sfu            ; mV to sfu
y=y+200
;y=alog10(y)
ll00:
x=x*dt
;
xa=0
xe=0
ya=0
ye=0
x:
if fn1 eq fn1_old then read,'xrange: xa xe ',xa,xe
goto ,z
y:
if fn1 eq fn1_old then read,'yrange: ya ye ',ya,ye
z:
if xa eq 0 and xe eq 0 and ya eq 0 and ye eq 0 then begin
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
endif
print,'min',ya,' max',ye
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,yticks=10,nsum=
n_sum
close,1
;
fn1 old=fn1
if pp eq 0 then begin
    while ccc ne 'q' do begin
        print,'          x  change the xrange'
        print,'          y  change the yrange'
        print,'          f  change the data file'
        print,'          p  prints current picture'
        print,'          q  leaves procedure'
        ccc=get kbrd(1)
        if ccc eq 'p' then begin
            pp=1
            goto,PRINT
        endif
        if ccc eq 'x' then goto, x
        if ccc eq 'y' then goto, y
        if ccc eq 'f' then goto, f
    endwhile
endbegin

```

```

        endwhile
    endif

    PRINT:
    if pp eq 1 then begin
    print, 'printing picture'

    set plot, 'ps'
    ;device, /landscape
    devfn=getenv('USER')+fn1
    devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
    device, filename=devfn

    plot, x, y, xrange=[xa, xe], yrange=[ya, ye], title='!17',
    xtitle='!17'+sx, ytitle='!17'+sy, /xstyle, /ystyle, xticks=10, yticks=10, nsum=
    n_sum

    idltime=systime()
    ;xyouts, -100, -90, '!17USER: '+getenv('USER')+ ' FILE:+' '+fn1+' '!17 grf by
    !12Schwarz!17 IDL output from '+idltime, /dev
    device, /close
    set plot, 'X'
    ;spawn, 'lpr -Plps '+devfn
    ;spawn, 'rm '+devfn
    endif

    raus:
    end
    :::::::::::::::
    g
    :::::::::::::::
    ;pro grf
    yt=''
    ;
    psymbol=0
    n_sum=1
    threshold=0.1
    pp=0
    ccc=''
    fn1=''
    fn1 old=''
    !p.multi=[0,1,1]
    st=''
    sx=''
    sy=''
    f:
    ;read, 'File name: ', fn1
    fn1='Climate.d'
    columns = 2
    ;read, 'Number of columns: ', columns
    f = fltarr(columns)
    xa=0
    xe=0
    ya=0
    ye=0
    if n_elements(ppt) eq 0 then ppt=0

    ;print, 1, strcompress(fn1+string(1), /remove all)
    openr, 1, fn1, error=ex ; Datei zum lesen oeffnen
    readf, 1, st ; read title
    readf, 1, st ; read title
    readf, 1, st ; read title
    numb = long(0)
    while not eof(1) do begin
    readf, 1, dummy ; count of input row number
    ;print, dummy
    numb = numb + 1
    endwhile
    print, numb
    close, 1
    openr, 1, fn1, error=ex ; Datei zum lesen oeffnen
    print, 'file number ', 1, ' successful'

```

```

if ex ne 0 then goto,raus
readf,1,st                ; read title
readf,1,sx                ; x-axis notation
readf,1,sy                ; x-axis notation
if numb le 0 then goto, raus
x=fltarr(numb)            ; Vektor x als Feld (0..ix-1) vereinbaren
y=fltarr(numb)            ; Vektor y als Feld (0..iy-1) vereinbaren
z=fltarr(numb)            ; Vektor y als Feld (0..iy-1) vereinbaren
sos=intarr(numb)
for i=long(0),numb-1 do begin
readf,1,f                ; read the current row
if columns eq 2 then begin
x(i) = f(0)
y(i) = f(1)
endif else begin
x(i) = i
y(i) = f(0)
endelse
;print,i,x(i),y(i)
endfor
close,1
y=y+100
standy=stdev(y,meany)
print,'Mean ',meany,' standard dev ',standy
x:
if fn1 eq fn1_old then read,'xrange: xa xe ',xa,xe
goto ,z
y:
if fn1 eq fn1_old then read,'yrange: ya ye ',ya,ye
z:
;y=alog10(y)
if xa eq 0 and xe eq 0 and ya eq 0 and ye eq 0 then begin
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
ya=-60
ye=300
threshold=ya
endif
print,'xa xe ya ye',xa,xe,ya,ye
s:
print,'symbol (0=without, 10=histo) ',psymbol
;read,'new symbols ',psymbol
b:
print,'point number ',numb
print,'old binsize (1=default) ', n_sum
;read,'new binsize: ',n_sum
sy='Relative Einheiten'
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st, xtitle='!17'+sx, $
yttitle='!17'+sy,/xstyle,/ystyle,xticks=10,nsum=n_sum,psym=psymbol,
$
yticks=10
oy=50
xyouts,1020,oy, 'Oort',/data
xyouts,1290,oy, 'Wolf',/data
xyouts,1430,oy, 'Spoerer',/data
xyouts,1650,oy, 'Maunder',/data
xyouts,1780,oy, 'Dalton',/data
xyouts,1100,100, 'Grosse Minima',/data
xyouts,1030,160, 'Sonnenfleckenbeobachtungen',/data
xyouts,1550,270, 'Sonnenfleckenrelativzahlen',/data
xyouts,1090,20, '!7D!17!u14!nC-Messungen',/data

fn1='Naked'
openr,1,fn1,error=ex      ; Datei zum lesen oeffnen
readf,1,yt                ; read title
readf,1,yt                ; read title
readf,1,yt                ; read title
numb = long(0)
while not eof(1) do begin
readf,1,dummy            ; count of input row number

```

```

;print,dummy
numb = numb + 1
endwhile
print,numb
close,1
openr,1,fn1,error=ex          ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
readf,1,yt                    ; read title
readf,1,yt                    ; x-axis notation
readf,1,yt                    ; x-axis notation
if numb le 0 then goto, raus
x2=fltarr(numb)               ; Vektor x als Feld (0..ix-1)
vereinbaren
y2=fltarr(numb)               ; Vektor y als Feld (0..iy-1)
vereinbaren
for i=long(0),numb-1 do begin
readf,1,f                     ; read the current row
if columns eq 2 then begin
x2(i) = f(0)
y2(i) = f(1)
endif else begin
x(i) = i
y2(i) = f(0)
endelse
;print,i,x(i),y(i)
endfor
close,1
standy=stdev(y2,meany)
y2=y2*4+180
print,'Mean ',meany,' standard dev ',standy
oplot,x2,y2,psym=10

fn1='sunspoty.dat'
openr,1,fn1,error=ex          ; Datei zum lesen oeffnen
readf,1,yt                    ; read title
readf,1,yt                    ; read title
readf,1,yt                    ; read title
numb = long(0)
while not eof(1) do begin
readf,1,dummy                 ; count of input row number
;print,dummy
numb = numb + 1
endwhile
print,numb
close,1
openr,1,fn1,error=ex          ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
readf,1,yt                    ; read title
readf,1,yt                    ; x-axis notation
readf,1,yt                    ; x-axis notation
if numb le 0 then goto, raus
x3=fltarr(numb)               ; Vektor x als Feld (0..ix-1)
vereinbaren
y3=fltarr(numb)               ; Vektor y als Feld (0..iy-1)
vereinbaren
for i=long(0),numb-1 do begin
readf,1,f                     ; read the current row
if columns eq 2 then begin
x3(i) = f(0)
y3(i) = f(1)
endif else begin
x(i) = i
y3(i) = f(0)
endelse
;print,i,x(i),y(i)
endfor
close,1
standy=stdev(y3,meany)
y3=y3+100
print,'Mean ',meany,' standard dev ',standy

```



```

oplot,x3,y3,psym=10

fn1='C14'
openr,1,fn1,error=ex           ; Datei zum lesen oeffnen
readf,1,yt                     ; read title
readf,1,yt                     ; read title
readf,1,yt                     ; read title
numb = long(0)
while not eof(1) do begin
readf,1,dummy                 ; count of input row number
;print,dummy
numb = numb + 1
endwhile
print,numb
close,1
openr,1,fn1,error=ex           ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
readf,1,yt                     ; read title
readf,1,yt                     ; x-axis notation
readf,1,yt                     ; x-axis notation
if numb le 0 then goto, raus
x4=fltarr(numb)                ; Vektor x als Feld (0..ix-1)
vereinbaren
y4=fltarr(numb)                ; Vektor y als Feld (0..iy-1)
vereinbaren
for i=long(0),numb-1 do begin
readf,1,f                     ; read the current row
if columns eq 2 then begin
x4(i) = f(0)
y4(i) = f(1)
endif else begin
x(i) = i
y4(i) = f(0)
endelse
;print,i,x(i),y(i)
endifor
close,1
standy=stdev(y4,meany)
y4=y4*2
print,'Mean ',meany,' standard dev ',standy
oplot,x4,y4,psym=10

t:
jjj=1
if jjj eq 1 then goto , jjj
print,'y-axis greater than ', threshold
read,'lower threshold: ',threshold
for i=0,numb-1 do if y(sos(i)) ge threshold then $
print,x(sos(i)),1./x(sos(i)),y(sos(i))
;
jjj:
fn1 old=fn1
if pp eq 0 then begin
while ccc ne 'q' do begin
print,'
print,' x change the xrange'
print,' y change the yrange'
print,' b change the binsize'
print,' s change the symbols'
print,' f change the data file'
print,' t change the Threshold'
print,' p prints current picture'
print,' q leaves procedure'
ccc=get kbrd(1)
if ccc eq 'p' then begin
pp=1
goto,PRINT
endif
if ccc eq 'x' then goto, x
if ccc eq 'y' then goto, y
if ccc eq 'b' then goto, b
if ccc eq 's' then goto, s

```

```

        if ccc eq 'f' then goto, f
        if ccc eq 't' then goto, t
    endwhile
endif

PRINT:
if pp eq 1 then begin
print, 'printing picture'

set plot, 'ps'
;device, /landscape, /inches, xsize=6.452, ysize=6.0
;device, /landscape
idltime=systime()
devfn=getenv('USER')+'Sunactivity'
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn

plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17', xtitle='!17'+sx, $
    ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,nsum=n_sum,psym=psymbol,
    $
    yticks=10
oplot,x2,y2,psym=10
oplot,x3,y3,psym=10
oplot,x4,y4,psym=10
oy=50
xyouts,1020,oy, 'Oort',/data
xyouts,1280,oy, 'Wolf',/data
xyouts,1420,oy, 'Spoerer',/data
xyouts,1620,oy, 'Maunder',/data
xyouts,1780,oy, 'Dalton',/data
xyouts,1100,100, 'Grosse Minima',/data
xyouts,1030,160, 'Sonnenfleckenbeobachtungen',/data
xyouts,1550,270, 'Sonnenfleckenrelativzahlen',/data
xyouts,1090,20, '!7D!17!u14!nC-Messungen',/data

;xyouts,-100,-90,'!17USER: '+getenv('USER')+' FILE:'+' '+fn1+' .!17 grf by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close
set plot,'X'
;spawn,'lpr -Plps '+devfn
;spawn,'rm '+devfn
endif

raus:
end
::::::::::::::::::
grf.pro
::::::::::::::::::
pro grf
;+
; NAME: grf.pro
; PURPOSE: printing of four grf files each with two columns
; CALLING SEQUENCE: psgrf2 4,'n1',...,'n4'
; INPUTS: title, x-axis, y-axis
;-
; input structure
; title
; x-axis
; y-axis
; grf file with two columns and any number of rows
;
psymbol=0
n sum=1
threshold=0.1
pp=0
ccc=''
fn1=''
fn1_old=''
!p.multi=[0,1,1]
st=''
sx=''
sy=''

```

```

f:
read,'File name: ',fn1
columns = 2
read,'Number of columns: ',columns
f = fltarr(columns)
xa=0
xe=0
ya=0
ye=0
if n_elements(ppt) eq 0 then ppt=0

;print,1,strcompress(fn1+string(1), /remove all)
openr,1,fn1,error=ex ; Datei zum lesen oeffnen
readf,1,st ; read title
readf,1,st ; read title
readf,1,st ; read title
numb = long(0)
while not eof(1) do begin
readf,1,dummy ; count of input row number
;print,dummy
numb = numb + 1
endwhile
print,numb
close,1
openr,1,fn1,error=ex ; Datei zum lesen oeffnen
print,'file number ',1,' successful'
if ex ne 0 then goto,raus
readf,1,st ; read title
readf,1,sx ; x-axis notation
readf,1,sy ; x-axis notation
if numb le 0 then goto, raus
x=fltarr(numb) ; Vektor x als Feld (0..ix-1) vereinbaren
y=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
z=fltarr(numb) ; Vektor z als Feld (0..iz-1) vereinbaren
sos=intarr(numb)
for i=long(0),numb-1 do begin
readf,1,f ; read the current row
if columns eq 2 then begin
x(i) = f(0)
y(i) = f(1)
endif else begin
x(i) = i
y(i) = f(0)
endelse
;print,i,x(i),y(i)
endfor
close,1
standy=stdev(y,meany)
print,'Mean ',meany,' standard dev ',standy
x:
if fn1 eq fn1_old then read,'xrange: xa xe ',xa,xe
goto ,z
y:
if fn1 eq fn1_old then read,'yrange: ya ye ',ya,ye
z:
;y=alog10(y)
if xa eq 0 and xe eq 0 and ya eq 0 and ye eq 0 then begin
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
threshold=ya
endif
print,'xa xe ya ye',xa,xe,ya,ye
s:
print,'symbol (0=without, 10=histo) ',psymbol
read,'new symbols ',psymbol
b:
print,'point number ',numb
print,'old binsize (1=default) ', n_sum
read,'new binsize: ',n_sum
zero=intarr(numb)

```

```

plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st, xtitle='!17'+sx, $
  ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,nsum=n_sum,psym=psymbol,
  $
  yticks=10
;oplot,x,zero
;plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,psym=3
;plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,psym=-1,xticks=10,yticks=10
;oplot,x,z,linestyle=1
sos=reverse(sort(y))
t:
jjj=1
if jjj eq 1 then goto , jjj
print,'y-axis greater than ', threshold
read,'lower threshold: ',threshold
for i=0,numb-1 do if y(sos(i)) ge threshold then $
  print,x(sos(i)),1./x(sos(i)),y(sos(i))
;
jjj:
fn1 old=fn1
if pp eq 0 then begin
  while ccc ne 'q' do begin
    print,'          x  change the xrange'
    print,'          y  change the yrange'
    print,'          b  change the binsize'
    print,'          s  change the symbols'
    print,'          f  change the data file'
    print,'          t  change the Threshold'
    print,'          p  prints current picture'
    print,'          q  leaves procedure'
    ccc=get kbrd(1)
    if ccc eq 'p' then begin
      pp=1
      goto,PRINT
    endif
    if ccc eq 'x' then goto, x
    if ccc eq 'y' then goto, y
    if ccc eq 'b' then goto, b
    if ccc eq 's' then goto, s
    if ccc eq 'f' then goto, f
    if ccc eq 't' then goto, t
  endwhile
endif
PRINT:
if pp eq 1 then begin
print,'printing picture'

set plot,'ps'
;device, /inches,xsize=5.3,ysize=5.0
;device, /landscape, /inches,xsize=6.452,ysize=6.0
;device, /landscape
devfn=getenv('USER')+fn1
devfn=devfn+strtrim(string(fix(systemtime(1) mod 1000000)),2)+'.ps'
device,filename=devfn

;plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,yticks=10
;st='Rebinned, interpolated and detrended data'
;plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17', xtitle='!17'+sx, $
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17', xtitle='!17'+sx, $
  ytitle='!17'+sy,/xstyle,/ystyle,xticks=10,nsum=n_sum,psym=psymbol,
  $
  yticks=10
;  ytitle='!17'+sy,/xstyle,/ystyle,psym=-1,xticks=10
;oplot,x,zero

idltime=systemtime()
;xyouts,-100,-90,'!17USER: '+getenv('USER')+ ' FILE:+' '+' +fn1+' .!17 grf by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close

```

```

set plot, 'X'
;spawn, 'lpr -Plps '+devfn
;spawn, 'rm '+devfn
endif

raus:
end
::::::::::::::::::
h
::::::::::::::::::
;Feigenbaum
set plot, 'X'
pp=0
ccc=''
;+
!p.multi=[0,1,1]
a= double(1)
st=''
sx='Kontrollparameter r'
sy='x'
index_total=512
control_total=640
multi=1
einhalb=intarr(control_total)+0.5
transient=300
xx00=double(0.33)
dv=dblarr(index_total,control_total)
ds=intarr(index_total,control_total)
d=intarr(index_total,control_total)
x = findgen(index_total)/float(index_total)
xf = findgen(control_total)/float(control_total)
xa=0
xe=4
ya=0
ye=1

xf=xf+3.
;window,0,title='Histogram'
for ctrl=0,control_total-1 do begin
a=double(3.+ float(ctrl)/float(control_total))
;print,'control ',ctrl,a
xi=xx00
    ; transient
    for itr=0,transient-1 do xi=a*xi*(1-xi)
    for itr=long(0),multi*long(index_total-1) do begin
    ; iteration loop
    xi=a*xi*(1-xi)
    index=fix(index_total*xi)
    d(index,ctrl)=d(index,ctrl)+1
    dv(index,ctrl)=xi
    ds(index,ctrl)=-1
    ;print,xi,index
    endfor
da=0
de=max(d)
;plot,x,d(0:index_total-1,ctrl),xrange=[ya,ye],yrange=
[da,de],title='!17'+st, xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle
;xyouts, 0.1, da/2, string(format='("r = ",f5.3)',a),/data
endifor

;window,1,title='Feigenbaum diagram'
;plot,xf,dv(0,0:control_total-1),psym=3,ytitle='!17 x',xtitle='!17 a'
;for itr=0,index_total-1 do begin
;oplot,xf,dv(itr,0:control_total-1),psym=3
;endifor
d_max=max(d(0:index_total-1,control_total-1))
;d=d<256
d=d<d_max
ds=log((d+1))
dss=ds
;window,2

```

```

;surface,dss
ds max=max(ds)
;einhalb line
ds(index_total/2-1,0:control_total-1)=ds_max
bb=bytsc1(transpose(-1*ds))
;window,0
;tv,bb
k=3
l=4
ll=1
kk=0
print,'printing picture'
set_plot,'ps'

;;device, /inches,xsize=5.3,ysize=5.0
;;device, /landscape, /inches,xsize=6.452,ysize=6.0
;;device, /landscape
idltime=systime()
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn

plot, [k, l], [kk, ll], xrange=[k, l], yrange=[kk, ll], title='!17', xtitle='!17'+sx,
yttitle='!17'+sy, ticklen=-0.02, xticks=10, yticks=10, ystyle=1, xstyle=1, /nodata
oplot,xf,einhalb
tvsc1,bb,2240,1420,xsize=14860,ysize=10550 ; portr"at
;tvsc1,bb,2240,1420,xsize=21000,ysize=15600 ; landscape
xyouts, 3.4, 0.3, 'r!I2!N = 3.4495',/data
xyouts, 3.5, 0.25, 'r!I3!N = 3.5441',/data
xyouts, 3.5, 0.2, 'r!I4y!17!N = 3.5699',/data
xyouts, 3.7, 0.1, 'r!IT5!N = 3.7379',/data
xyouts, 3.8, 0.05, 'r!IT3!N = 3.8283',/data

device,/close

set_plot,'X'
endif

raus:
;spawn,'lpr -Plps '+devfn
;spawn,'rm '+devfn
end
:::::::::::::
i
:::::::::::::
;pro i
itera=''
read,'Iteration (1-5): ',itera
threshold=0.1
pp=0
ccc=''
;+
!p.multi=[0,1,1]
a= double(1)
read,'Kontrollparameter (0-4): ',a
print, 'a ',a
st=''
sx='x!In!N'
sy='x!In+1!N'
beginprint:
length=100
dt = 1./double(length)
limit = 10
x = dblarr(length)
x(0)=0
y = dblarr(length)

;for jj=1,limit do begin
for ii =0,length-1 do x(ii) = double(ii+1)*dt
;f = ...
case itera of
'1' : begin

```

```

y = x*(1-x)*a
end

'2' : begin
y = A^2*X*( - A*X^3 + 2*A*X^2 - A*X - X + 1)
end

'3' : begin
y = A^3*X*( - A^4*X^7 + 4*A^4*X^6 - 6*A^4*X^5 + 4*A^4*X^4 - A^4*X^3
- 2*A^3*X^5 + 6*A^3*X^4 - 6*A^3*X^3 + 2*A^3*X^2 - A^2*X^3 +
2*A^2*X^2 - A^2*X - A*X^3 + 2*A*X^2 - A*X - X + 1)
end

'4' : begin
y1=-
A^11*X^15+8*A^11*X^14-28*A^11*X^13+56*A^11*X^12-70*A^11*X^11+56*A^11*X^10-2
8*A^11*X^9+8*A^11*X^8-A^11*X^7-4*A^10*X^13+28*A^10*X^12-84*A^10*X^11
y2=
140*A^10*X^10-140*A^10*X^9+84*A^10*X^8-28*A^10*X^7+4*A^10*X^6-6*A^9*X^11+36
*A^9*X^10-90*A^9*X^9+120*A^9*X^8-90*A^9*X^7+36*A^9*X^6-6*A^9*X^5-2*A^8*X^11
y3=
12*A^8*X^10-34*A^8*X^9+60*A^8*X^8-70*A^8*X^7+52*A^8*X^6-22*A^8*X^5+4*A^8*X^
4-6*A^7*X^9+30*A^7*X^8-61*A^7*X^7+64*A^7*X^6-36*A^7*X^5+10*A^7*X^4-A^7*X^3
y4=-6*A^6*X^7+24*A^6*X^6-36*A^6*X^5+24*A^6*X^4-6*A^6*X^3-
A^5*X^7+4*A^5*X^6-8*A^5*X^5+10*A^5*X^4-7*A^5*X^3+2*A^5*X^2-
A^4*X^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4
y5=-7*A^4*X^3+2*A^4*X^2-2*A^3*X^5+6*A^3*X^4-7*A^3*X^3+4*A^3*X^2-A^3*X-
A^2*X^3+2*A^2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1
;
y=A^4*X*(y1+y2+y3+y4+y5)
end

'5' : begin

; y=-A^26*X^31+16*A^26*X^30-120*A^26*X^29+560* $
; A^26*X^28-1820*A^26*X^27+4368*A^26*X^26-8008 $
; *A^26*X^25+11440*A^26*X^24-12870*A^26*X^23+ $
; 11440*A^26*X^22-8008*A^26*X^21+4368*A^26*X^20 $
; -1820*A^26*X^19+560*A^26*X^18-120*A^26*X^17+ $
; 16*A^26*X^16-A^26*X^15-8*A^25*X^29+120*A^ $
; 25*X^28-840*A^25*X^27+3640*A^25*X^26-10920*A $
; ^25*X^25+24024*A^25*X^24-40040*A^25*X^23+ $
; 51480*A^25*X^22-51480*A^25*X^21+40040*A^25*X^ $
; 20-24024*A^25*X^19+10920*A^25*X^18-3640*A^25* $
; X^17+840*A^25*X^16-120*A^25*X^15+8*A^25*X^ $
; 14-28*A^24*X^27+392*A^24*X^26-2548*A^24*X^25 $
; +10192*A^24*X^24-28028*A^24*X^23+56056*A^24*X $
; ^22-84084*A^24*X^21+96096*A^24*X^20-84084*A^ $
; 24*X^19+56056*A^24*X^18-28028*A^24*X^17+10192 $
; *A^24*X^16-2548*A^24*X^15+392*A^24*X^14-28*A $
; ^24*X^13-4*A^23*X^27+56*A^23*X^26-420*A^23* $
; X^25+2184*A^23*X^24-8372*A^23*X^23+24024*A^ $
; 23*X^22-52052*A^23*X^21+85800*A^23*X^20- $
; 108108*A^23*X^19+104104*A^23*X^18-76076*A^23*X $
; ^17+41496*A^23*X^16-16380*A^23*X^15+4424*A^ $
; 23*X^14-732*A^23*X^13+56*A^23*X^12-28*A^22*X $
; ^25+364*A^22*X^24-2254*A^22*X^23+8848*A^22*X $
; ^22-24640*A^22*X^21+51436*A^22*X^20-82698*A^ $
; 22*X^19+103488*A^22*X^18-100716*A^22*X^17+ $
; 75460*A^22*X^16-42658*A^22*X^15+17584*A^22*X^ $
; 14-4984*A^22*X^13+868*A^22*X^12-70*A^22*X^11 $
; -84*A^21*X^23+1008*A^21*X^22-5600*A^21*X^21 $
; +19096*A^21*X^20-44660*A^21*X^19+75768*A^21*X $
; ^18-96096*A^21*X^17+92400*A^21*X^16-67452*A^ $
; 21*X^15+36960*A^21*X^14-14784*A^21*X^13+4088* $
; A^21*X^12-700*A^21*X^11+56*A^21*X^10-6*A^20 $
; *X^23+72*A^20*X^22-536*A^20*X^21+2860*A^20*X $
; ^20-10698*A^20*X^19+28132*A^20*X^18-53004*A^ $
; 20*X^17+72792*A^20*X^16-73530*A^20*X^15+54576 $
; *A^20*X^14-29376*A^20*X^13+11132*A^20*X^12- $
; 2806*A^20*X^11+420*A^20*X^10-28*A^20*X^9-2*A $

```

```

;^19*X^23+24*A^19*X^22-168*A^19*X^21+836*A^ $
;19*X^20-3110*A^19*X^19+8924*A^19*X^18-20036*A $
;^19*X^17+35088*A^19*X^16-47310*A^19*X^15+ $
;48272*A^19*X^14-36480*A^19*X^13+19812*A^19*X^ $
;12-7370*A^19*X^11+1724*A^19*X^10-212*A^19*X^ $
;9+8*A^19*X^8-12*A^18*X^21+132*A^18*X^20- $
;750*A^18*X^19+2880*A^18*X^18-8094*A^18*X^17+ $
;17100*A^18*X^16-27469*A^18*X^15+33704*A^18*X^ $
;14-31492*A^18*X^13+22100*A^18*X^12-11308*A^18 $
;X^11+3992*A^18*X^10-874*A^18*X^9+92*A^18*X $
;^8-A^18*X^7-34*A^17*X^19+340*A^17*X^18- $
;1650*A^17*X^17+5160*A^17*X^16-11488*A^17*X^15 $
;+18872*A^17*X^14-23044*A^17*X^13+20768*A^17*X $
;^12-13570*A^17*X^11+6228*A^17*X^10-1898*A^17 $
;X^9+344*A^17*X^8-28*A^17*X^7-6*A^16*X^19+ $
;60*A^16*X^18-330*A^16*X^17+1260*A^16*X^16- $
;3510*A^16*X^15+7272*A^16*X^14-11344*A^16*X^13 $
;+13348*A^16*X^12-11694*A^16*X^11+7400*A^16*X^ $
;10-3206*A^16*X^9+864*A^16*X^8-118*A^16*X^7+ $
;4*A^16*X^6-30*A^15*X^17+270*A^15*X^16-1150* $
;A^15*X^15+3080*A^15*X^14-5776*A^15*X^13+7952 $
;A^15*X^12-8176*A^15*X^11+6260*A^15*X^10- $
;3490*A^15*X^9+1346*A^15*X^8-322*A^15*X^7+36* $
;A^15*X^6-61*A^14*X^15+488*A^14*X^14-1760*A^ $
;14*X^13+3780*A^14*X^12-5368*A^14*X^11+5272*A $
;^14*X^10-3618*A^14*X^9+1700*A^14*X^8-515*A^ $
;14*X^7+88*A^14*X^6-6*A^14*X^5-6*A^13*X^15+ $
;48*A^13*X^14-232*A^13*X^13+784*A^13*X^12- $
;1786*A^13*X^11+2708*A^13*X^10-2738*A^13*X^9+ $
;1832*A^13*X^8-784*A^13*X^7+196*A^13*X^6-22*A $
;^13*X^5-A^12*X^15+8*A^12*X^14-52*A^12*X^13 $
;+224*A^12*X^12-610*A^12*X^11+1112*A^12*X^10 $
;-1412*A^12*X^9+1252*A^12*X^8-749*A^12*X^7+ $
;280*A^12*X^6-56*A^12*X^5+4*A^12*X^4-A^11*X $
;^15+8*A^11*X^14-32*A^11*X^13+84*A^11*X^12- $
;190*A^11*X^11+412*A^11*X^10-718*A^11*X^9+ $
;862*A^11*X^8-669*A^11*X^7+320*A^11*X^6-86*A $
;^11*X^5+10*A^11*X^4-4*A^10*X^13+28*A^10*X^ $
;12-92*A^10*X^11+188*A^10*X^10-284*A^10*X^9+ $
;364*A^10*X^8-389*A^10*X^7+296*A^10*X^6-134* $
;A^10*X^5+28*A^10*X^4-A^10*X^3-8*A^9*X^11+ $
;48*A^9*X^10-130*A^9*X^9+210*A^9*X^8-226*A^ $
;9*X^7+172*A^9*X^6-94*A^9*X^5+34*A^9*X^4-6 $
;A^9*X^3-2*A^8*X^11+12*A^8*X^10-40*A^8*X^9 $
;+90*A^8*X^8-137*A^8*X^7+140*A^8*X^6-94*A^ $
;8*X^5+38*A^8*X^4-7*A^8*X^3-6*A^7*X^9+30*A $
;^7*X^8-67*A^7*X^7+88*A^7*X^6-74*A^7*X^5+ $
;40*A^7*X^4-13*A^7*X^3+2*A^7*X^2-7*A^6*X^7 $
;+28*A^6*X^6-44*A^6*X^5+34*A^6*X^4-13*A^6*X $
;^3+2*A^6*X^2-A^5*X^7+4*A^5*X^6-10*A^5*X $
;^5+16*A^5*X^4-13*A^5*X^3+4*A^5*X^2-A^4*X $
;^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4-8*A^4* $
;X^3+4*A^4*X^2-A^4*X-2*A^3*X^5+6*A^3*X^4 $
;-7*A^3*X^3+4*A^3*X^2-A^3*X-A^2*X^3+2*A^ $
;2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1

;L=A^5*X*()
end

endcase

;for ii =0,length-1 do print,x(ii),y(ii)
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
xa=0
xe=1
ya=0
ye=1
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle

```



```

xx00=double(0.33)
yy00=double(0.0)
n_total=1000
x2=dblarr(n_total)
y2=x2
xi=xx00
x2(0)=xx00
y2(0)=yy00
for i=1,n_total-1 do begin
if (i mod 2 eq 1) then begin
    xi=a*xi*(1-xi)
    x2(i)=x2(i-1)
    y2(i)=xi
endif
if (i mod 2 eq 0 and i gt 0) then begin
    x2(i)=xi
    y2(i)=xi
endif
endifor
x2(0)=xx00
y2(0)=yy00
;oplot,x2(1:n_total-1),y2(1:n_total-1),linestyle=1
oplot,x2,y2,linestyle=1
;wait,1
xyouts, 0.1, 0.8, string(format='("r = ",f5.3)',a),/data
;endfor

oplot,x,x
if (pp eq 1) then goto, endprint

if pp eq 0 then begin
    while ccc ne 'q' do begin
        print,'                                p prints current picture'
        print,'                                q leaves procedure'
        ccc=get_kbrd(1)
        if ccc eq 'p' then begin
            pp=1
            goto,PRINT
        endif
    endwhile
endif

PRINT:
if pp eq 1 then begin
print,'printing picture'
set_plot,'ps'
device, /inches,xsize=5.3,ysize=5.0
;device, /landscape, /inches,xsize=6.452,ysize=6.0
;device, /landscape
idltime=systemtime()
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systemtime(1) mod 1000000)),2)+'.ps'
device,filename=devfn
goto, beginprint

endprint:

;xyouts,-100,-90,'!17USER: '+getenv('USER')+' FILE:'+' .!17 xsin by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close
set_plot,'X'
endif

raus:
;spawn,'lpr -Plps '+devfn
;spawn,'rm '+devfn
end
::::::::::::::::::
psret1 4.pro
::::::::::::::::::
pro psret1_4,fn1,fn2,fn3,fn4

```

```

;+
; NAME: psmmap2 4.pro
; PURPOSE: printing of four grf files each with two columns
; CALLING SEQUENCE: psgrf2_4,'n1',...,'n4'
; INPUTS: title, x-axis, y-axis
;-
; input structure
; title
; x-axis
; y-axis
; grf file with two columns and any number of rows
;
set plot,'ps'
device, /landscape
!x.minor=10
!y.minor=5
!p.multi=[0,2,2]
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn
st=''
sx=''
sy=''
columns = 1
f = fltarr(columns)
if n elements(ppt) eq 0 then ppt=0
for ifn=1,4 do begin
print,'The four file names are ',fn1,' ',fn2,' ',fn3,' ',fn4
;print,ifn,strcompress(fn+string(ifn), /remove_all)
if ifn eq 1 then openr,ifn,fn1,error=ex ; Datei zum lesen
oeffnen
if ifn eq 2 then openr,ifn,fn2,error=ex ; Datei zum lesen
oeffnen
if ifn eq 3 then openr,ifn,fn3,error=ex ; Datei zum lesen
oeffnen
if ifn eq 4 then openr,ifn,fn4,error=ex ; Datei zum lesen
oeffnen
readf,ifn,st ; read title
readf,ifn,sx ; read title
readf,ifn,sy ; read title
numb = 0
while not eof(ifn) do begin
readf,ifn,dummy ; count of input row number
;print,dummy
if numb eq 0 then oldy = dummy
numb = numb + 1
end
print,numb
close,ifn
if ifn eq 1 then openr,ifn,fn1,error=ex ; Datei zum lesen
oeffnen
if ifn eq 2 then openr,ifn,fn2,error=ex ; Datei zum lesen
oeffnen
if ifn eq 3 then openr,ifn,fn3,error=ex ; Datei zum lesen
oeffnen
if ifn eq 4 then openr,ifn,fn4,error=ex ; Datei zum lesen
oeffnen
print,'file number ',ifn,' successful'
if ex ne 0 then goto,raus
readf,ifn,st ; read title
readf,ifn,sx ; x-axis notation
readf,ifn,sy ; x-axis notation
;readf,ifn,sl,xa,xe,ya,ye ; Anz. Funkt., Xan,Xend,Yan,Yend
einlesen
if numb le 0 then goto, raus
x=fltarr(numb) ; Vektor x als Feld (0..ix-1) vereinbaren
y=fltarr(numb) ; Vektor y als Feld (0..iy-1) vereinbaren
for i=0,numb-1 do begin
readf,ifn,f ; read the current row
x(i) = oldy
y(i) = f(0)
oldy = y(i)

```

```

;print,x(i),y(i)
endfor
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
;if ifn eq 1 then xa=1050
;if ifn eq 1 then ya=1050
plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle,psym=3
idltime=systime()
close,ifn
endfor
raus:
idltime=systime()
xyouts,-100,-90,!17USER: '+getenv('USER')+ FILE:'+' '+fn1+' '+fn2+'
'+fn3+' '+fn4+'.!17 psm2_4 by !12Sch!17 IDL psret2_4 from '+idltime,/dev
device,/close
spawn,'lpr -Plps '+devfn
spawn,'rm '+devfn
end
::::::::::::::::::
space.pro
::::::::::::::::::
pro space,file=fn,tcol=tsp,xcol=xsp,ycol=ysp,zcol=zsp,title=ts,$
  subtitle=ss1,sub2title=ss2,sub3title=ss3,sub4title=ss4,$
  sub5title=ss5,$
  print=pp,rotate=rot,rx=xr,ry=yr,rz=zr
;+
; NAME: space.pro
; PURPOSE: displaying or printing of phase spaces
; CATEGORY: Graphics
; CALLING SEQUENCE: space
;           space with keywordparameters
; INPUTS: file containing the space data
; OPTIONAL INPUT PARAMETERS:
; KEYWORD PARAMETERS:
;   title   - Title for the plot
;   subtitle - Subtitle for the plot
;   sub2title - Subtitle for the plot
;   sub3title - Subtitle for the plot
;   sub4title - Subtitle for the plot
;   sub5title - Subtitle for the plot
;   file    - file name
;   tcol    - total coloumns in file
;   xcol    - x coloumn in file
;   ycol    - y coloumn in file
;   zcol    - z coloumn in file
;   print   - if non-zero prints the picture to -Plps
; OUTPUTS: picture to screen or PostScript printer
; OPTIONAL OUTPUT PARAMETERS:
; COMMON BLOCKS:
; SIDE EFFECTS: display will be updated
;           temporary file will be written and deleted
; RESTRICTIONS:
; PROCEDURE: straightforward
; MODIFICATION HISTORY: Stefan Schmidt, 04.06.1991
;-
i=0
if n elements(fn) ne 0 then goto,eins
fn=''
read,'file name : ',fn
eins:
if n elements(tsp) ne 0 then goto,zwei
read,'total number of coloumns : ',tsp
zwei:
if n elements(xsp) ne 0 then goto,drei
read,'number x coloumn : ',xsp
drei:
if n elements(ysp) ne 0 then goto,vier
read,'number y coloumn : ',ysp
vier:

```

```

if n_elements(zsp) ne 0 then goto,fuenf
read,'number z coloumn : ',zsp
fuenf:
if n_elements(pp) eq 0 then pp=0 else pp=1
if n_elements(rot) eq 0 then rot=0 else rot=1
if n_elements(xr) eq 0 then xr=0
if n_elements(yr) eq 0 then yr=0
if n_elements(zr) eq 0 then zr=0
xi=fltarr(tsp)
openr,1,fn
while not eof(1) do begin
  readf,1,xi
  i=i+1
end
print,'file contains ',i,' lines.'
point lun,1,0
xx=fltarr(3,i)
for j=0,2 do for k=0,i-1 do xx(j,k)=0.
for k=0,i-1 do begin
  readf,1,xi
  xx(0,k)=xi(xsp-1)
  xx(1,k)=xi(ysp-1)
  xx(2,k)=xi(zsp-1)
endfor
close,1
ccc=''
x1=xr
y1=yr
z1=zr
if pp eq 0 then begin
  if rot eq 1 then begin
    while ccc ne 'q' do begin
      print,'select rotation :  x  set new x rotation'
      print,'                      y  set new y rotation'
      print,'                      z  set new z rotation'
      print,'                      p  prints current picture'
      print,'                      q  leaves procedure'
      erase,0
      t3d,/reset,rotate=[x1,y1,z1] & scale3d & phase,xx
      xyouts,10,!d.y vsize-10,'x = '+string(fix(x1)),/dev
      xyouts,10,!d.y vsize-30,'y = '+string(fix(y1)),/dev
      xyouts,10,!d.y vsize-50,'z = '+string(fix(z1)),/dev
      ccc=get kbrd(1)
      if ccc eq 'p' then begin
        xr=x1
        yr=y1
        zr=z1
        pp=1
        goto,PRINT
      end
      if ccc eq 'x' then read,'new x rotation : ',x1
      if ccc eq 'y' then read,'new y rotation : ',y1
      if ccc eq 'z' then read,'new z rotation : ',z1
    end
  end
  if rot eq 0 then begin
    erase,0
    t3d,/reset,rotate=[xr,yr,zr] & scale3d & phase,xx
  end
end
PRINT:
if pp eq 1 then begin
  print,'printing picture'
  if n_elements(ts) eq 0 then begin
    ts=''
    read,'title for picture : ',ts
  end
  if n_elements(ss1) eq 0 then begin
    ss1=''
    read,'subtitle 1 for picture : ',ss1
  end
  if n_elements(ss2) eq 0 then begin

```

```

    ss2=''
    read,'subtitle 2 for picture : ',ss2
end
if n_elements(ss3) eq 0 then begin
    ss3=''
    read,'subtitle 3 for picture : ',ss3
end
if n_elements(ss4) eq 0 then begin
    ss4=''
    read,'subtitle 4 for picture : ',ss4
end
if n_elements(ss5) eq 0 then begin
    ss5=''
    read,'subtitle 5 for picture : ',ss5
end
grde=!d.name
set plot,'ps'
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,file=devfn
t3d,/reset,rotate=[xr,yr,zr] & scale3d & phase,xx
; xyouts,2000,13000,'!17'+ts,/dev
; xyouts,2000,0,'!17'+ss1,/dev
; xyouts,2000,-500,'!17'+ss2,/dev
; xyouts,2000,-1000,'!17'+ss3,/dev
; xyouts,2000,-1500,'!17'+ss4,/dev
; xyouts,2000,-2000,'!17'+ss5,/dev
; xyouts,2000,-3500,'!17USER :',/dev
; xyouts,4000,-3500,'!17'+getenv('USER'),/dev
; xyouts,2000,-4000,'!17FILE :',/dev
; xyouts,4000,-4000,'!17'+fn,/dev
; xyouts,2000,-5000,'!17X coloumn :',/dev
; xyouts,6000,-5000,'!17'+string(fix(xsp)),/dev
; xyouts,2000,-5500,'!17Y coloumn :',/dev
; xyouts,6000,-5500,'!17'+string(fix(ysp)),/dev
; xyouts,2000,-6000,'!17Z coloumn :',/dev
; xyouts,6000,-6000,'!17'+string(fix(zsp)),/dev
; xyouts,2000,-6500,'!17X rotation :',/dev
; xyouts,6000,-6500,'!17'+string(fix(xr)),/dev
; xyouts,2000,-7000,'!17Y rotation :',/dev
; xyouts,6000,-7000,'!17'+string(fix(yr)),/dev
; xyouts,2000,-7500,'!17Z rotation :',/dev
; xyouts,6000,-7500,'!17'+string(fix(zr)),/dev
; xyouts,2000,-8500,'!17IDL routine SPACE by !12St. Schmidt!17,
04.07.1991',/dev
; xyouts,2000,-9000,'!17IDL output from '+systime(),/dev
device,/close
set plot,grde
spawn,'lpr -Plps '+devfn
spawn,'rm '+devfn
end
print,'bye.'
schluss:
end
::::::::::::::::::
x.pro
::::::::::::::::::
pro x
threshold=0.1
pp=0
ccc=''
;+
; NAME: psgrf2 1.pro
; PURPOSE: printing of four grf files each with two columns
; CALLING SEQUENCE: psgrf2 4,'n1',...,'n4'
; INPUTS: title, x-axis, y-axis
;-
; input structure
; title
; x-axis
; y-axis
; grf file with two columns and any number of rows

```

```

;
!p.multi=[0,1,1]
st=''
sx='x!In!N'
sy='x!In+1!N'
length=100
dt = 1./double(length)
limit = 10
x = dblarr(length)
x(0)=0
y = dblarr(length)
beginprint:
for jj=1,limit do begin
a= double(jj)*4/double(limit)
print, 'a ',a
for ii =0,length-1 do x(ii) = double(ii+1)*dt
;f = ...
y = x*(1-x)*a
;g = ...
;y = A^2*X*( - A*X^3 + 2*A*X^2 - A*X - X + 1)
;h = ...
;y = A^3*X*( - A^4*X^7 + 4*A^4*X^6 - 6*A^4*X^5 + 4*A^4*X^4 - A^4*X^3 -
2*A^3*X^5 + 6*A^3*X^4 - 6*A^3*X^3 + 2*A^3*X^2 - A^2*X^3 + 2*A^2*X^2 -
A^2*X - A*X^3 + 2*A*X^2 - A*X - X + 1)

;k = ...
;y1=-
A^11*X^15+8*A^11*X^14-28*A^11*X^13+56*A^11*X^12-70*A^11*X^11+56*A^11*X^10-2
8*A^11*X^9+8*A^11*X^8-A^11*X^7-4*A^10*X^13+28*A^10*X^12-84*A^10*X^11
;y2=
140*A^10*X^10-140*A^10*X^9+84*A^10*X^8-28*A^10*X^7+4*A^10*X^6-6*A^9*X^11+36
*A^9*X^10-90*A^9*X^9+120*A^9*X^8-90*A^9*X^7+36*A^9*X^6-6*A^9*X^5-2*A^8*X^11
;y3=
12*A^8*X^10-34*A^8*X^9+60*A^8*X^8-70*A^8*X^7+52*A^8*X^6-22*A^8*X^5+4*A^8*X^
4-6*A^7*X^9+30*A^7*X^8-61*A^7*X^7+64*A^7*X^6-36*A^7*X^5+10*A^7*X^4-A^7*X^3
;y4=-6*A^6*X^7+24*A^6*X^6-36*A^6*X^5+24*A^6*X^4-6*A^6*X^3-
A^5*X^7+4*A^5*X^6-8*A^5*X^5+10*A^5*X^4-7*A^5*X^3+2*A^5*X^2-
A^4*X^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4
;y5=-7*A^4*X^3+2*A^4*X^2-2*A^3*X^5+6*A^3*X^4-7*A^3*X^3+4*A^3*X^2-A^3*X-
A^2*X^3+2*A^2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1
;
;y=A^4*X*(y1+y2+y3+y4+y5)
;
; ; l = ...
;
;
;y1=-A^26*X^31+16*A^26*X^30-120*A^26*X^29+560*
;A^26*X^28-1820*A^26*X^27+4368*A^26*X^26-8008
;A^26*X^25+11440*A^26*X^24-12870*A^26*X^23+
;11440*A^26*X^22-8008*A^26*X^21+4368*A^26*X^20
;-1820*A^26*X^19+560*A^26*X^18-120*A^26*X^17+
;16*A^26*X^16-A^26*X^15-8*A^25*X^29+120*A^
;25*X^28-840*A^25*X^27+3640*A^25*X^26-10920*A
;^25*X^25+24024*A^25*X^24-40040*A^25*X^23+
;51480*A^25*X^22-51480*A^25*X^21+40040*A^25*X^
;20-24024*A^25*X^19+10920*A^25*X^18-3640*A^25*
;X^17+840*A^25*X^16-120*A^25*X^15+8*A^25*X^
;14-28*A^24*X^27+392*A^24*X^26-2548*A^24*X^25
;+10192*A^24*X^24-28028*A^24*X^23+56056*A^24*X
;^22-84084*A^24*X^21+96096*A^24*X^20-84084*A^
;24*X^19+56056*A^24*X^18-28028*A^24*X^17+10192
;A^24*X^16-2548*A^24*X^15+392*A^24*X^14-28*A
;^24*X^13-4*A^23*X^27+56*A^23*X^26-420*A^23*
;X^25+2184*A^23*X^24-8372*A^23*X^23+24024*A^
;23*X^22-52052*A^23*X^21+85800*A^23*X^20-
;108108*A^23*X^19+104104*A^23*X^18-76076*A^23*X
;^17+41496*A^23*X^16-16380*A^23*X^15+4424*A^
;23*X^14-732*A^23*X^13+56*A^23*X^12-28*A^22*X
;^25+364*A^22*X^24-2254*A^22*X^23+8848*A^22*X
;^22-24640*A^22*X^21+51436*A^22*X^20-82698*A^
;22*X^19+103488*A^22*X^18-100716*A^22*X^17+
;75460*A^22*X^16-42658*A^22*X^15+17584*A^22*X^

```

```

;14-4984*A^22*X^13+868*A^22*X^12-70*A^22*X^11
;-84*A^21*X^23+1008*A^21*X^22-5600*A^21*X^21
;+19096*A^21*X^20-44660*A^21*X^19+75768*A^21*X
;^18-96096*A^21*X^17+92400*A^21*X^16-67452*A^
;21*X^15+36960*A^21*X^14-14784*A^21*X^13+4088*
;A^21*X^12-700*A^21*X^11+56*A^21*X^10-6*A^20
;*X^23+72*A^20*X^22-536*A^20*X^21+2860*A^20*X
;^20-10698*A^20*X^19+28132*A^20*X^18-53004*A^
;20*X^17+72792*A^20*X^16-73530*A^20*X^15+54576
;*A^20*X^14-29376*A^20*X^13+11132*A^20*X^12-
;2806*A^20*X^11+420*A^20*X^10-28*A^20*X^9-2*A
;^19*X^23+24*A^19*X^22-168*A^19*X^21+836*A^
;19*X^20-3110*A^19*X^19+8924*A^19*X^18-20036*A
;^19*X^17+35088*A^19*X^16-47310*A^19*X^15+
;48272*A^19*X^14-36480*A^19*X^13+19812*A^19*X^
;12-7370*A^19*X^11+1724*A^19*X^10-212*A^19*X^
;9+8*A^19*X^8-12*A^18*X^21+132*A^18*X^20-
;750*A^18*X^19+2880*A^18*X^18-8094*A^18*X^17+
;17100*A^18*X^16-27469*A^18*X^15+33704*A^18*X^
;14-31492*A^18*X^13+22100*A^18*X^12-11308*A^18
;*X^11+3992*A^18*X^10-874*A^18*X^9+92*A^18*X
;^8-A^18*X^7-34*A^17*X^19+340*A^17*X^18-
;1650*A^17*X^17+5160*A^17*X^16-11488*A^17*X^15
;+18872*A^17*X^14-23044*A^17*X^13+20768*A^17*X
;^12-13570*A^17*X^11+6228*A^17*X^10-1898*A^17
;*X^9+344*A^17*X^8-28*A^17*X^7-6*A^16*X^19+
;60*A^16*X^18-330*A^16*X^17+1260*A^16*X^16-
;3510*A^16*X^15+7272*A^16*X^14-11344*A^16*X^13
;+13348*A^16*X^12-11694*A^16*X^11+7400*A^16*X^
;10-3206*A^16*X^9+864*A^16*X^8-118*A^16*X^7+
;4*A^16*X^6-30*A^15*X^17+270*A^15*X^16-1150*
;A^15*X^15+3080*A^15*X^14-5776*A^15*X^13+7952
;*A^15*X^12-8176*A^15*X^11+6260*A^15*X^10-
;3490*A^15*X^9+1346*A^15*X^8-322*A^15*X^7+36*
;A^15*X^6-61*A^14*X^15+488*A^14*X^14-1760*A^
;14*X^13+3780*A^14*X^12-5368*A^14*X^11+5272*A
;^14*X^10-3618*A^14*X^9+1700*A^14*X^8-515*A^
;14*X^7+88*A^14*X^6-6*A^14*X^5-6*A^13*X^15+
;48*A^13*X^14-232*A^13*X^13+784*A^13*X^12-
;1786*A^13*X^11+2708*A^13*X^10-2738*A^13*X^9+
;1832*A^13*X^8-784*A^13*X^7+196*A^13*X^6-22*A
;^13*X^5-A^12*X^15+8*A^12*X^14-52*A^12*X^13
;+224*A^12*X^12-610*A^12*X^11+1112*A^12*X^10
;-1412*A^12*X^9+1252*A^12*X^8-749*A^12*X^7+
;280*A^12*X^6-56*A^12*X^5+4*A^12*X^4-A^11*X
;^15+8*A^11*X^14-32*A^11*X^13+84*A^11*X^12-
;190*A^11*X^11+412*A^11*X^10-718*A^11*X^9+
;862*A^11*X^8-669*A^11*X^7+320*A^11*X^6-86*A
;^11*X^5+10*A^11*X^4-4*A^10*X^13+28*A^10*X^
;12-92*A^10*X^11+188*A^10*X^10-284*A^10*X^9+
;364*A^10*X^8-389*A^10*X^7+296*A^10*X^6-134*
;A^10*X^5+28*A^10*X^4-A^10*X^3-8*A^9*X^11+
;48*A^9*X^10-130*A^9*X^9+210*A^9*X^8-226*A^
;9*X^7+172*A^9*X^6-94*A^9*X^5+34*A^9*X^4-6
;*A^9*X^3-2*A^8*X^11+12*A^8*X^10-40*A^8*X^9
;+90*A^8*X^8-137*A^8*X^7+140*A^8*X^6-94*A^
;8*X^5+38*A^8*X^4-7*A^8*X^3-6*A^7*X^9+30*A
;^7*X^8-67*A^7*X^7+88*A^7*X^6-74*A^7*X^5+
;40*A^7*X^4-13*A^7*X^3+2*A^7*X^2-7*A^6*X^7
;+28*A^6*X^6-44*A^6*X^5+34*A^6*X^4-13*A^6*X
;^3+2*A^6*X^2-A^5*X^7+4*A^5*X^6-10*A^5*X
;^5+16*A^5*X^4-13*A^5*X^3+4*A^5*X^2-A^4*X
;^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4-8*A^4*
;X^3+4*A^4*X^2-A^4*X-2*A^3*X^5+6*A^3*X^4
; -7*A^3*X^3+4*A^3*X^2-A^3*X-A^2*X^3+2*A^
;2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1
;
;L=A^5*X*()

;for ii =0,length-1 do print,x(ii),y(ii)
xa=min(x)
xe=max(x)

```

```

ya=min(y)
ye=max(y)
xa=0
xe=1
ya=0
ye=1
if (jj eq 1) then plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle
idltime=systime()
if (jj ne 1) then oplot,x,y,linestyle=((jj-1) mod 5),thick=(jj-1)/5+1
;wait,1
xyouts, 0.45 , float(jj)*0.1 -0.05, string(format='("r = ",f3.1)',a),/data
endfor
oplot,x,x
if (pp eq 1) then goto, endprint

if pp eq 0 then begin
  while ccc ne 'q' do begin
    print,'                               p prints current picture'
    print,'                               q leaves procedure'
    ccc=get kbrd(1)
    if ccc eq 'p' then begin
      pp=1
      goto,PRINT
    endif
  endwhile
endif

PRINT:
if pp eq 1 then begin
print,'printing picture'
set plot,'ps'
device, /inches,xsize=5.3,ysize=5.0
;device, /landscape, /inches,xsize=6.452,ysize=6.0
;device, /landscape
idltime=systime()
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn
goto, beginprint

endprint:

;xyouts,-100,-90,'!17USER: '+getenv('USER')+' FILE:'+' .!17 xsin by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close
set_plot,'X'
endif

raus:
;spawn,'lpr -Plps '+devfn
;spawn,'rm '+devfn
end
::::::::::::::::::
y.c
::::::::::::::::::
/*  UMLAUTEpc_dec.C  -->  Wandelt in einem Text die Umlaute          *
*      " , " , • , á  in  \ "a, \ "o, \ "u, \ "s um.          *
*      (ž , ™ , š  werden nicht ber•cksichtigt.)          */

#include <stdio.h>          /* Definitionen der Konstanten EOF          */
                          /* und der Makros getchar(), putchar(). */
#include <stdlib.h>
main(int argc, char **argv[])
{
  FILE *fi,*fo;
  int  c, i, iff, index, j, k;

  if((fi=fopen(argv[1],"r"))==NULL) {printf("fehler
eingabedatei\n");exit(4);}
  if((fo=fopen(argv[2],"w"))==NULL) {printf("fehler
ausgabedatei\n");exit(4);}

```



```

    for (i=1;i<20;i++)
    {
        for (j=1;j<=34;++j) c = getc(fi);
        for (j=1;j<=61;++j)
        {
            k++;
            iff=fscanf(fi,"%4d",&index);
            printf("%4.2f\n",index*.01);
            fprintf(fo,"%d %g\n",k,index*.01);
        }
    }
/*      while( (c = getc(fi)) != EOF);      Zeichen einzeln lesen */
}
::::::::::::::::::
y.pro
::::::::::::::::::
pro y
itera=''
read,'Iteration (1-5): ',itera
threshold=0.1
pp=0
ccc=''
;+
; NAME: psgrf2 1.pro
; PURPOSE: printing of four grf files each with two columns
; CALLING SEQUENCE: psgrf2 4,'n1',...,'n4'
; INPUTS: title, x-axis, y-axis
;-
;   input structure
;   title
;   x-axis
;   y-axis
;   grf file with two columns and any number of rows
;
!p.multi=[0,1,1]
st=''
sx='x!In!N'
sy='x!In+1!N'
length=100
dt = 1./double(length)
limit = 10
x = dblarr(length)
x(0)=0
y = dblarr(length)
beginprint:
for jj=1,limit do begin
a= double(jj)*4/double(limit)
print, 'a ',a
for ii =0,length-1 do x(ii) = double(ii+1)*dt
;f = ...
case itera of
'1' : begin
    y = x*(1-x)*a
    end
'2' : begin
    y = A^2*X*( - A*X^3 + 2*A*X^2 - A*X - X + 1)
    end
'3' : begin
    y = A^3*X*( - A^4*X^7 + 4*A^4*X^6 - 6*A^4*X^5 + 4*A^4*X^4 - A^4*X^3
      - 2*A^3*X^5 + 6*A^3*X^4 - 6*A^3*X^3 + 2 *A^3*X^2 - A^2*X^3 +
      2*A^2*X^2 - A^2*X - A*X^3 + 2*A*X^2 - A*X - X + 1)
    end
'4' : begin
y1=-
A^11*X^15+8*A^11*X^14-28*A^11*X^13+56*A^11*X^12-70*A^11*X^11+56*A^11*X^10-2
8*A^11*X^9+8*A^11*X^8-A^11*X^7-4*A^10*X^13+28*A^10*X^12-84*A^10*X^11
y2=
140*A^10*X^10-140*A^10*X^9+84*A^10*X^8-28*A^10*X^7+4*A^10*X^6-6*A^9*X^11+36

```

```

*A^9*X^10-90*A^9*X^9+120*A^9*X^8-90*A^9*X^7+36*A^9*X^6-6*A^9*X^5-2*A^8*X^11
y3=
12*A^8*X^10-34*A^8*X^9+60*A^8*X^8-70*A^8*X^7+52*A^8*X^6-22*A^8*X^5+4*A^8*X^
4-6*A^7*X^9+30*A^7*X^8-61*A^7*X^7+64*A^7*X^6-36*A^7*X^5+10*A^7*X^4-A^7*X^3
y4=-6*A^6*X^7+24*A^6*X^6-36*A^6*X^5+24*A^6*X^4-6*A^6*X^3-
A^5*X^7+4*A^5*X^6-8*A^5*X^5+10*A^5*X^4-7*A^5*X^3+2*A^5*X^2-
A^4*X^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4
y5=-7*A^4*X^3+2*A^4*X^2-2*A^3*X^5+6*A^3*X^4-7*A^3*X^3+4*A^3*X^2-A^3*X-
A^2*X^3+2*A^2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1
;
y=A^4*X*(y1+y2+y3+y4+y5)
end

'5' : begin

; y=-A^26*X^31+16*A^26*X^30-120*A^26*X^29+560* $
; A^26*X^28-1820*A^26*X^27+4368*A^26*X^26-8008 $
; *A^26*X^25+11440*A^26*X^24-12870*A^26*X^23+ $
; 11440*A^26*X^22-8008*A^26*X^21+4368*A^26*X^20 $
; -1820*A^26*X^19+560*A^26*X^18-120*A^26*X^17+ $
; 16*A^26*X^16-A^26*X^15-8*A^25*X^29+120*A^ $
; 25*X^28-840*A^25*X^27+3640*A^25*X^26-10920*A $
; ^25*X^25+24024*A^25*X^24-40040*A^25*X^23+ $
; 51480*A^25*X^22-51480*A^25*X^21+40040*A^25*X^ $
; 20-24024*A^25*X^19+10920*A^25*X^18-3640*A^25* $
; X^17+840*A^25*X^16-120*A^25*X^15+8*A^25*X^ $
; 14-28*A^24*X^27+392*A^24*X^26-2548*A^24*X^25 $
; +10192*A^24*X^24-28028*A^24*X^23+56056*A^24*X $
; ^22-84084*A^24*X^21+96096*A^24*X^20-84084*A^ $
; 24*X^19+56056*A^24*X^18-28028*A^24*X^17+10192 $
; *A^24*X^16-2548*A^24*X^15+392*A^24*X^14-28*A $
; ^24*X^13-4*A^23*X^27+56*A^23*X^26-420*A^23* $
; X^25+2184*A^23*X^24-8372*A^23*X^23+24024*A^ $
; 23*X^22-52052*A^23*X^21+85800*A^23*X^20- $
; 108108*A^23*X^19+104104*A^23*X^18-76076*A^23*X $
; ^17+41496*A^23*X^16-16380*A^23*X^15+4424*A^ $
; 23*X^14-732*A^23*X^13+56*A^23*X^12-28*A^22*X $
; ^25+364*A^22*X^24-2254*A^22*X^23+8848*A^22*X $
; ^22-24640*A^22*X^21+51436*A^22*X^20-82698*A^ $
; 22*X^19+103488*A^22*X^18-100716*A^22*X^17+ $
; 75460*A^22*X^16-42658*A^22*X^15+17584*A^22*X^ $
; 14-4984*A^22*X^13+868*A^22*X^12-70*A^22*X^11 $
; -84*A^21*X^23+1008*A^21*X^22-5600*A^21*X^21 $
; +19096*A^21*X^20-44660*A^21*X^19+75768*A^21*X $
; ^18-96096*A^21*X^17+92400*A^21*X^16-67452*A^ $
; 21*X^15+36960*A^21*X^14-14784*A^21*X^13+4088* $
; A^21*X^12-700*A^21*X^11+56*A^21*X^10-6*A^20 $
; *X^23+72*A^20*X^22-536*A^20*X^21+2860*A^20*X $
; ^20-10698*A^20*X^19+28132*A^20*X^18-53004*A^ $
; 20*X^17+72792*A^20*X^16-73530*A^20*X^15+54576 $
; *A^20*X^14-29376*A^20*X^13+11132*A^20*X^12- $
; 2806*A^20*X^11+420*A^20*X^10-28*A^20*X^9-2*A $
; ^19*X^23+24*A^19*X^22-168*A^19*X^21+836*A^ $
; 19*X^20-3110*A^19*X^19+8924*A^19*X^18-20036*A $
; ^19*X^17+35088*A^19*X^16-47310*A^19*X^15+ $
; 48272*A^19*X^14-36480*A^19*X^13+19812*A^19*X^ $
; 12-7370*A^19*X^11+1724*A^19*X^10-212*A^19*X^ $
; 9+8*A^19*X^8-12*A^18*X^21+132*A^18*X^20- $
; 750*A^18*X^19+2880*A^18*X^18-8094*A^18*X^17+ $
; 17100*A^18*X^16-27469*A^18*X^15+33704*A^18*X^ $
; 14-31492*A^18*X^13+22100*A^18*X^12-11308*A^18 $
; *X^11+3992*A^18*X^10-874*A^18*X^9+92*A^18*X $
; ^8-A^18*X^7-34*A^17*X^19+340*A^17*X^18- $
; 1650*A^17*X^17+5160*A^17*X^16-11488*A^17*X^15 $
; +18872*A^17*X^14-23044*A^17*X^13+20768*A^17*X $
; ^12-13570*A^17*X^11+6228*A^17*X^10-1898*A^17 $
; *X^9+344*A^17*X^8-28*A^17*X^7-6*A^16*X^19+ $
; 60*A^16*X^18-330*A^16*X^17+1260*A^16*X^16- $
; 3510*A^16*X^15+7272*A^16*X^14-11344*A^16*X^13 $
; +13348*A^16*X^12-11694*A^16*X^11+7400*A^16*X^ $
; 10-3206*A^16*X^9+864*A^16*X^8-118*A^16*X^7+ $

```

```

;4*A^16*X^6-30*A^15*X^17+270*A^15*X^16-1150* $
;A^15*X^15+3080*A^15*X^14-5776*A^15*X^13+7952 $
;A^15*X^12-8176*A^15*X^11+6260*A^15*X^10- $
;3490*A^15*X^9+1346*A^15*X^8-322*A^15*X^7+36* $
;A^15*X^6-61*A^14*X^15+488*A^14*X^14-1760*A^ $
;14*X^13+3780*A^14*X^12-5368*A^14*X^11+5272*A $
;^14*X^10-3618*A^14*X^9+1700*A^14*X^8-515*A^ $
;14*X^7+88*A^14*X^6-6*A^14*X^5-6*A^13*X^15+ $
;48*A^13*X^14-232*A^13*X^13+784*A^13*X^12- $
;1786*A^13*X^11+2708*A^13*X^10-2738*A^13*X^9+ $
;1832*A^13*X^8-784*A^13*X^7+196*A^13*X^6-22*A $
;^13*X^5-A^12*X^15+8*A^12*X^14-52*A^12*X^13 $
;+224*A^12*X^12-610*A^12*X^11+1112*A^12*X^10 $
; -1412*A^12*X^9+1252*A^12*X^8-749*A^12*X^7+ $
;280*A^12*X^6-56*A^12*X^5+4*A^12*X^4-A^11*X $
;^15+8*A^11*X^14-32*A^11*X^13+84*A^11*X^12- $
;190*A^11*X^11+412*A^11*X^10-718*A^11*X^9+ $
;862*A^11*X^8-669*A^11*X^7+320*A^11*X^6-86*A $
;^11*X^5+10*A^11*X^4-4*A^10*X^13+28*A^10*X^ $
;12-92*A^10*X^11+188*A^10*X^10-284*A^10*X^9+ $
;364*A^10*X^8-389*A^10*X^7+296*A^10*X^6-134* $
;A^10*X^5+28*A^10*X^4-A^10*X^3-8*A^9*X^11+ $
;48*A^9*X^10-130*A^9*X^9+210*A^9*X^8-226*A^ $
;9*X^7+172*A^9*X^6-94*A^9*X^5+34*A^9*X^4-6 $
;A^9*X^3-2*A^8*X^11+12*A^8*X^10-40*A^8*X^9 $
;+90*A^8*X^8-137*A^8*X^7+140*A^8*X^6-94*A^ $
;8*X^5+38*A^8*X^4-7*A^8*X^3-6*A^7*X^9+30*A $
;^7*X^8-67*A^7*X^7+88*A^7*X^6-74*A^7*X^5+ $
;40*A^7*X^4-13*A^7*X^3+2*A^7*X^2-7*A^6*X^7 $
;+28*A^6*X^6-44*A^6*X^5+34*A^6*X^4-13*A^6*X $
;^3+2*A^6*X^2-A^5*X^7+4*A^5*X^6-10*A^5*X $
;^5+16*A^5*X^4-13*A^5*X^3+4*A^5*X^2-A^4*X $
;^7+4*A^4*X^6-8*A^4*X^5+10*A^4*X^4-8*A^4* $
;X^3+4*A^4*X^2-A^4*X-2*A^3*X^5+6*A^3*X^4 $
; -7*A^3*X^3+4*A^3*X^2-A^3*X-A^2*X^3+2*A^ $
;2*X^2-A^2*X-A*X^3+2*A*X^2-A*X-X+1

;L=A^5*X*()
      end

endcase

;for ii =0,length-1 do print,x(ii),y(ii)
xa=min(x)
xe=max(x)
ya=min(y)
ye=max(y)
xa=0
xe=1
ya=0
ye=1
if (jj eq 1) then plot,x,y,xrange=[xa,xe],yrange=[ya,ye],title='!17'+st,
xtitle='!17'+sx,ytitle='!17'+sy,/xstyle,/ystyle
idltime=systemtime()
if (jj ne 1) then oplot,x,y,linestyle=((jj-1) mod 5),thick=(jj-1)/5+1
;wait,1
;if (itera eq '1') then xyouts, 0.45 , float(jj)*0.1 -0.05,
string(format='("r = ",f3.1)',a),/data
;xyouts, 1-float(jj)*0.1 , y(45), string(format='("r = ",f3.1)',a),/data
xyouts, x(fix(5*jj)), y(fix(5*jj))-0.05, string(format='("r =
",f3.1)',a),/data
endfor
oplot,x,x
if (pp eq 1) then goto, endprint

if pp eq 0 then begin
  while ccc ne 'q' do begin
    print,'
    print,'
    ccc=get kbrd(1)
    if ccc eq 'p' then begin
      pp=1
      p prints current picture'
      q leaves procedure'
    end
  end
end

```

```

        goto, PRINT
      endif
    endwhile
  endif

PRINT:
if pp eq 1 then begin
print, 'printing picture'
set plot, 'ps'
device, /inches, xsize=5.3, ysize=5.0
;device, /landscape, /inches, xsize=6.452, ysize=6.0
;device, /landscape
idltime=systime()
devfn=getenv('USER')
devfn=devfn+strtrim(string(fix(systime(1) mod 1000000)),2)+'.ps'
device,filename=devfn
goto, beginprint

endprint:

;xyouts, -100, -90, '!17USER: '+getenv('USER')+' FILE:+' .!17 xsin by
!12Schwarz!17 IDL output from '+idltime,/dev
device,/close
set plot, 'X'
endif

raus:
;spawn, 'lpr -Plps '+devfn
;spawn, 'rm '+devfn
end
:::::::::::::
zARmodel.f
:::::::::::::
  program mf
  parameter (iacfl=100, nopera0=31, ndimm=1600)
  real*4  c0(ndimm)
  common /wiener/wie(nopera0)
  common /freq/ nprint, dx, h(2048)
  common /A/ ndim0, x(16384)
  common /ABE/ nopera, dt
  character*12  sname ! (8)  input original data file name
  character*2   armodel ! (16)AR model
  character*67  aname ! (11)  ACF
  character*17  title, xtitle, ytitle
  save
  write (*, *) '##### begin <-----#'
  nus0 = 0
  nus1 = 0
c siehe "E"
  dt = 6.04
c   write (*, 10)
  10  format('Name of the original data file : ', $)
c   read (*, '(a)') sname
  write (*, 20)
  20  format('Enter sample time: ', $)
  read (*, *) dt
  print *, ' '
  time zero=0.0
  write (*, 30)
  30  format('Enter time zero of the time axis: ', $)
  read (*, *) time_zero
  print *, ' '
  write (*, 40)
  40  format('Enter order of the AR model: ', $)
  read (*, *) nopera
  write (*, 50)
  50  format('Enter series length : ', $)
  read (*, *) ndim
  ndim=ndim+2
  nopera = nopera + 1
  print *, ' '
  d_mean = 0.

```

```

d var = 0.
x mean = 0.
x ma = 0.
x_m3 = 0.
x_m4 = 0.
x var = 0.
c open (8,file=sname,status='old')
c read (8,'(a)',end=17) title
c read (8,'(a)',end=17) xtitle
c read (8,'(a)',end=17) ytitle
c kk = 1
cc 16 read (8,*,end=17) w1,x(kk),w3,w4,w5
c16 read (8,*,end=17) x(kk)
cc write (*,*) x(kk)
c x_mean = x_mean + x(kk)
c kk = kk + 1
c if (kk.gt.ndimm) goto 17
c goto 16
c17 close (8)
c ndim = kk - 1
print *,' '
write (*,*) 'length of the series:',ndim
write (*,*) 'full time interval ',ndim*dt
write (*,*) 'length of the series for the AR operator:',ndim
write (*,*) 'Wiener operator '
write (*,*) 'AR process order ',nopera-1
do kk=1,nopera-1
write (*,60)
60 format('Enter AR coefficient : ',)$
read (*,*) wie(kk+1)
enddo
write (*,*) (wie(ir),ir=2,nopera)
c noise standard deviation
var = 10.0
write (*,*) 'noise standard deviation',var
care 4
cno = var*3.
write (*,*) 'variance ',cno
d mean = 0.
d var = 0.
x mean = 0.
x ma = 0.
x_m3 = 0.
x_m4 = 0.
x_var = 0.
c
c AR-model initialization
c
iein = 800
write (*,*) 'length of the transition phase ',iein
c
c begin of transition phase
c
do ir = 1,nopera-1
c0(ir) = (rand(kk)-0.5)*cno
enddo
do i=1,iein
c0(nopera) = (rand(kk)-0.5)*cno
do j=2,nopera
c0(nopera) = c0(nopera)+ wie(j)*c0(nopera-(j-1))
enddo
do j=2,nopera
c0(j-1) = c0(j)
enddo
enddo

if (nopera-1.lt.10) then
armodel = '0'//char(nopera-1+48)
else
armodel = char(int(float(nopera-1)/10.)+48)//
char(mod(nopera-1,10)+48)
+ endif

```

```

        title='AR model of the order: '//armodel
        xtitle='Number'
        ytitle='x'
        open(16,file='I'//armodel//sname,status='unknown')
        write(16,*) title
        write(16,*) xtitle
        write(16,*) ytitle

        do i=nopera,ndim
        c0(i) = (rand(kk)-0.5)*cno
c       write (*,*)'rand ',c0(i)
            do j=2,nopera
                c0(i) = c0(i)+ wie(j)*c0(i-(j-1))
            enddo
c       write(*,*) (wie(k),k=2,nopera)
        write (16,*) c0(i)
c       write (16,*) float(i)*dt,c0(i)
if(i.ge.ndim-8) write(*,*) i,'c0(i)',c0(i)
        x mean = x mean + c0(i)
        x ma   = x ma + abs(c0(i))
        x m3   = x m3 + c0(i)*c0(i)*c0(i)
        x_m4   = x_m4 + c0(i)*c0(i)*c0(i)*c0(i)
        x var  = x_var + c0(i)*c0(i)
        enddo
        close(16)
        x mean = x mean/float(ndim)
        x ma   = x ma/float(ndim-1)
        x m2   = x var/float(ndim-1)
        x_m3   = x_m3/float(ndim-1)
        x m4   = x m4/float(ndim-1)
        write (*,*) 'AR realization '
        write (*,*) 'Average      ',x mean
        write (*,*) 'Variance    ',x m2
        write (*,*) 'Skewness    ',x m3
        write (*,*) 'Peakedness  ',x m4
        write (*,*) 'Peakedness ',x ma,' > .79788 = m_abs(Gaussian) '
        x var = sqrt( x_var/float(ndim) - x_mean*x mean )
        write(*,*) 'mean+- sigma ',x mean,' ',x_var
        d mean = d mean/float(ndim-nopera+1)
        d var  = sqrt( d_var/float(ndim-nopera+1) - d mean*d mean )
        write(*,*) 'Pred_error mean +- sigma ',d_mean,' ',d_var

        write (*,*) '##### end <-----#'
        print *,' '
        print *,'Output: AR model on file ',mname
        print *,' '
        print *,'          ACF of the AR model on file ',aname
        print *,' '
stop
end

c
REAL FUNCTION RAND(K)
C       RANDOM NUMBER GENERATOR
C       MULTIPLIKATIVE KONGRUENZMETHODE
C       PIKE/HILL
C       COMM. ACM 8(1965) 10,605
C       DATA IX1,IX2,IA1,IA2,M,N /1895,8663,0,3125,32768,2048/
C
C       RAND FUNCTION
C
        save
        KL = IA2 * IX2
        IY1 = KL / M
        IY2 = KL - IY1 * M
        IY1 = IY1 + MOD(IA1*IX2,M) + MOD(IA2*IX1,M)
        IY1 = MOD(IY1,M)
        IX1 = MOD(IY1,N)
        IX2 = IY2
        RAND = (FLOAT(IX1) + FLOAT(IX2) / FLOAT(M)) / FLOAT(N)
        RETURN
        END
c

```

# 12. REDUCE-Texte

## Einführungen

Ueberberg, Johannes 1992 Einführung in die Computeralgebra mit REDUCE. Für Mathematiker, Informatiker und Physiker, BI Wissenschaftsverlag, Mannheim

Hehl, F.W., V. Winkelmann & H. Meyer 1993<sup>2</sup> REDUCE. Ein Kompaktkurs über die Anwendung von Computer-Algebra, Springer, ISBN 3-540-56705-4

Stauffer, D., F.W. Hehl, V. Winkelmann & J.G Zabolitzky 1993<sup>3</sup> Computer Simulation and Computer Algebra. Lectures for Beginners, Springer, ISBN 3-540-56530-2





```

operator y;
operator yy;
depend y,x;
write "enter taylor(n), n = number of equations";
write "enter right hand part of differential equation 1 .. N";
write " y^(i) = fi(x,y(1),...,y(n) = GL(i) ";
write "for instance GL(1) := y(1)*x";
write ".. and then call taylor1(L), L = TAYLOR expansion order";

```

```

algebraic procedure taylor(nn);
begin
dimension := nn;
array gl(nn);
end;

```

end;

Das ist die REDUCE-Ausgabe:

```

::::::::::::::::::
tl2.f
::::::::::::::::::
      SUBROUTINE TAYLOR(A,B,N,M,YY,Y0)
        common /const/ omega, l0, g
        DIMENSION YY(N,M),Y0(N)
C      REDUCE generated FORTRAN SUB ROUTINE
      H = (B-A) / FLOAT(M-1)
      X = A + H
      DO 10 I = 1,N
10     YY(I,1) = Y0(I)
      DO 40 J = 2,M
      S = 0.0
      Z=(YY(2.,J-1.)*H*OMEGA*(-SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)+L0)/SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)
      S = S + Z
      Z=(H**2*OMEGA*(-SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)*
      . YY(4.,J-1.)**2*YY(1.,J-1.)-SQRT(YY(4.,J-1.)**2+YY(
      . 2.,J-1.)**2)*YY(2.,J-1.)**2*YY(1.,J-1.)+YY(4.,J-1.)
      . **2*YY(1.,J-1.)*L0-YY(4.,J-1.)*YY(3.,J-1.)*YY(2.,J-
      . 1.)*L0)/(2.*SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)*(YY
      . (4.,J-1.)**2+YY(2.,J-1.)**2))
      S = S + Z
      YY(1.,J) = YY(1.,J-1) + S
      S = 0.0
      Z=YY(1.,J-1.)*H
      S = S + Z
      Z=(YY(2.,J-1.)*H**2*OMEGA*(SQRT(YY(4.,J-1.)**2+YY(2.
      . ,J-1.)**2)*L0-YY(4.,J-1.)**2-YY(2.,J-1.)**2))/(2.*(
      . YY(4.,J-1.)**2+YY(2.,J-1.)**2))
      S = S + Z
      YY(2.,J) = YY(2.,J-1) + S
      S = 0.0
      Z=(YY(4.,J-1.)*H*OMEGA*(SQRT(YY(4.,J-1.)**2+YY(2.,J-
      . 1.)**2)*G-SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)+L0)/
      . SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)
      S = S + Z
      Z=(H**2*OMEGA*(SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)*
      . YY(4.,J-1.)**2*YY(3.,J-1.)*G-SQRT(YY(4.,J-1.)**2+YY
      . (2.,J-1.)**2)*YY(4.,J-1.)**2*YY(3.,J-1.)+SQRT(YY(4.
      . ,J-1.)**2+YY(2.,J-1.)**2)*YY(3.,J-1.)*YY(2.,J-1.)**
      . 2*G-SQRT(YY(4.,J-1.)**2+YY(2.,J-1.)**2)*YY(3.,J-1.)
      . *YY(2.,J-1.)**2-YY(4.,J-1.)*YY(2.,J-1.)*YY(1.,J-1.)
      . *L0+YY(3.,J-1.)*YY(2.,J-1.)**2*L0)/(2.*SQRT(YY(4.,
      . J-1.)**2+YY(2.,J-1.)**2)*(YY(4.,J-1.)**2+YY(2.,J-1.
      . )**2))
      S = S + Z
      YY(3.,J) = YY(3.,J-1) + S
      S = 0.0
      Z=YY(3.,J-1.)*H

```



# Danksagung

Für die Überlassung experimenteller Daten danke ich Prof. Chuck Sonett (University of Arizona:  $\Delta^{14}\text{C}$ -Reihe), Dr. Matthias Förster (Berlin: Langmuir-Sonden-Daten), Dr. Seppo Urpo (Helsinki) und Dr. Albrecht Krüger (Astrophysikalisches Institut Potsdam: Tracking-Daten).

Für die Überlassung von Quelltexten danke ich Manfred Schultz, Stefan Schmidt, Michael Braun und Bernd-Michael Ritzmann Astrophysikalisches Institut Potsdam, Annette Witt, Max-Planck-Arbeitsgruppe Nichtlineare Dynamik an der Universität Potsdam.

Danken möchte ich allen Mitarbeitern der Max-Planck-Arbeitsgruppe Nichtlineare Dynamik an der Universität Potsdam, die durch eine angenehme Zusammenarbeit wichtige Voraussetzungen zum Entstehen dieser Arbeit mit geschaffen haben.

Den Mitgliedern des Astrophysikalischen Instituts Potsdam, insbesondere den Herren Prof. Rädler, André Saar und Dr. Andreas Priebe, danke ich für die großzügige Gastfreundschaft und Zurverfügungstellung von Raum, Material, Rechenleistung und vor allem für die Nutzung sehr guten Internetanbindung des Instituts.

Diese Arbeit ist in der Arbeitsgruppe Nichtlineare Dynamik der Max-Planck-Gesellschaft an der Universität Potsdam entstanden.