

Computational Physics

Jan Kierfeld

Version 9. April 2022

Vorwort

Das Skript orientiert sich an den Vorlesungen *Computational Physics* aus den Sommersemestern 2009, 2011, 2013, 2015, 2016 an der TU Dortmund. Es kann und wird Fehler enthalten.

E-mail jan.kierfeld@tu-dortmund.de

Homepage <https://cmt.physik.tu-dortmund.de/kierfeld-group/>

Jan Kierfeld

Ich bedanke mich für tatkräftige Mithilfe (Kapitel 7) bei Sebastian Knoche.

Inhaltsverzeichnis

1 Einleitung	8
1.1 Stellung in der Physik	8
1.2 Geschichte	9
1.3 Anwendungen	12
1.4 Nutzen	13
1.5 Inhalt	13
1.6 Literatur	15
1.7 Literaturverzeichnis Kapitel 1	15
2 Zahlen und Fehler	17
2.1 Zahldarstellungen	17
2.2 Benfordsches Gesetz	19
2.3 Fehler	22
2.3.1 Rundungsfehler	22
2.3.2 Abbruchfehler	23
2.3.3 Stabilität	23
2.4 Literaturverzeichnis Kapitel 2	25
2.5 Übungen Kapitel 2	26
3 Differentiation und Integration	27
3.1 Numerische Differentiation	27
3.1.1 Erste Ableitung	27
3.1.2 Zweite Ableitung	29
3.2 Numerische Integration	29
3.2.1 Trapezregel	30
3.2.2 Mittelpunktsregel	31
3.2.3 Simpsonregel	31
3.2.4 Euler-McLaurin Fehlerabschätzung	33
3.2.5 Romberg-Integration	34
3.2.6 Iterierte Trapezregel	35
3.2.7 Weitere Verfahren, mehrdimensionale Integrale	35
3.3 Uneigentliche Integrale	36
3.3.1 Unendliches Integrationsintervall	36
3.3.2 Singuläre Integranden	37
3.3.3 Hauptwertintegrale	38
3.3.4 Kramers-Kronig Relationen	38
3.4 Literaturverzeichnis Kapitel 3	44
3.5 Übungen Kapitel 3	45
4 Gewöhnliche Differentialgleichungen	48
4.1 Reduktion auf DGL erster Ordnung	48
4.2 Euler-Verfahren, Prädiktor-Korrektor	49
4.3 Runge-Kutta Verfahren	51
4.3.1 Runge-Kutta 2. Ordnung	51
4.3.2 Runge-Kutta 4. Ordnung	52

4.4	Schrittweitenanpassung	54
4.5	Integration Newtonscher Bewegungsgleichungen	56
4.5.1	Verlet-Algorithmen	56
4.5.2	Leapfrog-Algorithmus	57
4.6	Implizite Verfahren und steife DGL-Systeme	58
4.6.1	Implizite Verfahren	58
4.6.2	Steife DGL-Systeme	59
4.7	Weitere Verfahren	60
4.7.1	Prädiktor-Korrektor Verfahren höherer Ordnung	60
4.7.2	Bulirsch-Stoer Verfahren	61
4.7.3	Programmpakete/Solver	61
4.8	Literaturverzeichnis Kapitel 4	62
4.9	Übungen Kapitel 4	63
5	Molekulardynamik (MD) Simulation	66
5.1	Grundsätzliches	66
5.2	Kräfte, Randbedingungen, Initialisierung	68
5.2.1	Kräfte	68
5.2.2	Randbedingungen	73
5.2.3	Initialisierung	75
5.3	Integration	75
5.4	Messung von Observablen	80
5.4.1	Zeitmittel und Äquilibrierung	80
5.4.2	Energie, Temperatur	81
5.4.3	Druck	82
5.4.4	Paarverteilung	84
5.4.5	Paarverteilung und Virialentwicklung	86
5.4.6	Nachweis von Phasenübergängen	88
5.5	Kanonische MD Simulation	90
5.5.1	Isokinetischer Thermostat	90
5.5.2	Berendsen-Thermostat	90
5.5.3	Nose-Hoover Thermostat	91
5.6	Literaturverzeichnis Kapitel 5	93
5.7	Übungen Kapitel 5	94
6	Partielle Differentialgleichungen	95
6.1	Poisson-Gleichung	96
6.1.1	1D Poisson-Gleichung	96
6.1.2	2D Poisson-Gleichung	100
6.2	Wellengleichung	102
6.3	Diffusionsgleichung	104
6.4	Schrödingergleichung	106
6.5	Literaturverzeichnis Kapitel 6	108
6.6	Übungen Kapitel 6	109
7	Iterationsverfahren	111
7.1	Iterationen, Banachscher Fixpunktsatz	111
7.2	Nullstellen, Nichtlineare Gleichungen	114
7.2.1	Intervallhalbierung	114
7.2.2	Regula Falsi	115
7.2.3	Newton-Raphson-Methode	115
7.2.4	Nullstellen in höheren Dimensionen	116

7.3	Mean-Field Theorien und selbstkonsistente Gleichungen	117
7.3.1	Hartree-Fock-Näherung	118
7.3.2	Mean-Field-Theorie des Ising-Modells	119
7.4	Iterationen, Bifurkationen und Chaos	122
7.4.1	Iteration der Logistischen Abbildung	122
7.4.2	Fixpunkte, Bifurkationen und Chaos	124
7.4.3	Selbstähnlichkeit und Universalität	127
7.4.4	Renormierungsgruppe	129
7.5	Poincaré-Schnitte in chaotischen Systemen	131
7.5.1	Integrable Systeme	131
7.5.2	Poincaré-Schnitt	134
7.5.3	Weg ins Chaos: KAM-Theorem, Poincaré-Birkhoff-Theorem	136
7.6	Literaturverzeichnis Kapitel 7	139
7.7	Übungen Kapitel 7	140
8	Matrixdiagonalisierung, Eigenwertprobleme	141
8.1	Jacobi-Rotation	142
8.2	Householder und QR-Iteration	144
8.2.1	Householder-Algorithmus	144
8.2.2	Eigenwerte und Eigenvektoren tridiagonaler Matrizen	146
8.3	Potenzmethode, Transfermatrix	149
8.3.1	Potenzmethode	149
8.3.2	Transfermatrix des 1D Ising-Modells	150
8.3.3	Google PageRank	154
8.4	Matrixdiagonalisierung in der Quantenmechanik	156
8.5	Literaturverzeichnis Kapitel 8	157
8.6	Übungen Kapitel 8	158
9	Minimierung	161
9.1	Intervallhalbierung, Goldener Schnitt	163
9.2	Funktionen mehrerer Variablen	165
9.2.1	Konjugierte Richtungen	165
9.2.2	Powell-Verfahren	166
9.2.3	Steepest Descent	167
9.2.4	Konjugierte Gradienten	167
9.3	Literaturverzeichnis Kapitel 9	169
9.4	Übungen Kapitel 9	169
10	Zufallszahlen	170
10.1	Zufallszahlengeneratoren	170
10.1.1	Echter Zufall	170
10.1.2	Pseudo-Zufallszahlengeneratoren	171
10.1.3	Linear kongruente Generatoren	172
10.1.4	Xorshift und Kombinationen	173
10.2	Erzeugung verschiedener Verteilungen	174
10.2.1	Transformations- oder Inversionsmethode	175
10.2.2	Gaußverteilungen	176
10.2.3	Rückweisungsmethode	177
10.3	Literaturverzeichnis Kapitel 10	178
10.4	Übungen Kapitel 10	179

11 Monte-Carlo (MC) Simulation	180
11.1 Monte-Carlo Integration	181
11.1.1 Zwei Beispiele	181
11.1.2 Einfaches Sampling	183
11.1.3 Importance-Sampling	185
11.2 Markov-Sampling, Metropolis-Algorithmus	186
11.2.1 Markov-Prozesse, Master-Gleichung	187
11.2.2 Detailed Balance	191
11.2.3 Markov-Sampling, Metropolis-Algorithmus	192
11.3 MC Simulation (Beispiel Ising-Modell)	193
11.3.1 Ising-Modell	194
11.3.2 Metropolis-Algorithmus und Ising-Modell	196
11.3.3 Aufbau einer MC-Simulation	200
11.4 MC-Simulation kontinuierlicher Systeme	201
11.5 Skalengesetze, Finite-Size-Effekte	202
11.5.1 Korrelationslänge und Skalengesetze	203
11.5.2 Finite-Size-Scaling	204
11.6 Cluster-Algorithmen	207
11.7 Literaturverzeichnis Kapitel 11	209
11.8 Übungen Kapitel 11	211
12 Perkolation	214
12.1 Site- und Bond-Perkolation	214
12.1.1 Site-Perkolation	214
12.1.2 Bond-Perkolation	215
12.1.3 Geschichte und Anwendungen	215
12.2 Perkolation als Phasenübergang	216
12.2.1 Perkolationsschwelle	216
12.2.2 Cluster-Observablen und kritische Exponenten	217
12.3 Perkolation in $D=1$	218
12.3.1 Clusterzahlen	218
12.3.2 Perkolation in $D=1$	219
12.4 Potts-Modell und Perkolation	221
12.4.1 Q-Zustands Potts-Modell	221
12.4.2 Abbildung auf Perkolation im Limes $Q \rightarrow 1$	222
12.4.3 Mean-Field Theorie des Potts-Modells	225
12.5 Simulationsmethoden	227
12.5.1 Finite-Size-Scaling	227
12.5.2 Hoshen-Kopelman Algorithmus	229
12.6 Literaturverzeichnis Kapitel 12	232
12.7 Übungen Kapitel 12	233
13 Simulation stochastischer Bewegungsgleichungen	234
13.1 Brownsche Bewegung, Langevin-Gleichung	234
13.1.1 Ein Teilchen	234
13.1.2 N Teilchen	239
13.2 Langevin- und Brownsche Dynamik Simulation	240
13.2.1 Langevin-Dynamik Simulation	241
13.2.2 Brownsche Dynamik Simulation	242
13.3 Fokker-Planck-Gleichungen	243
13.3.1 Fokker-Planck-Gleichung (Rayleigh-Gleichung)	244
13.3.2 Klein-Kramers-Gleichung	246
13.3.3 Smoluchowski-Gleichung	247

13.3.4 Numerische Lösung von Fokker-Planck-Gleichungen	247
13.4 Literaturverzeichnis Kapitel 13	248

Literaturverzeichnis**249**

1 Einleitung

1.1 Stellung in der Physik

Die Unterteilung der Physik in “Theorie” und “Experiment” existiert erst seit ca. 1920, als sich mit Einstein und der zunehmenden mathematischen Komplexität physikalischer Theorien (allgemeine Relativitätstheorie) die theoretische Physik als eigenständiger Zweig der Physik etablierte. Mit der Quantenmechanik, Quantenfeldtheorien, der statistischen Physik von komplexen Phänomenen wie Phasenübergängen oder von komplexen Vielteilchensystemen in der Festkörperphysik manifestierte sich diese Trennung in Theorie und Experiment weiter.

Seit ca. 1950 sind zu diesen beiden Teildisziplinen die **Computerphysik oder Computersimulationen** als neues Teilgebiet hinzugekommen und nehmen einen stetig wachsenden Raum ein. “Computerexperimente” werden traditionell der Theorie zugeordnet und treten zunehmend an die Stelle analytischer Rechnungen, wo diese auf Grund der zunehmenden Komplexität der interessierenden Systeme nicht mehr möglich sind.

Ein physikalisches System, das experimentell untersucht wird, muss zunächst durch mathematisch formulierte Modelle oder Gleichungen theoretisch beschrieben werden, wobei verschiedene Level der Beschreibung möglich sind: Beispielsweise kann eine Flüssigkeit entweder klassisch hydrodynamisch auf dem Level von Geschwindigkeits- und Dichtefeldern durch partielle Differentialgleichungen wie die Navier-Stokes Gleichung beschrieben werden oder weit mikroskopischer durch einzelne wechselwirkende Flüssigkeitsteilchen. Einige theoretische Beschreibungen sind dann einer analytischen Lösung zugänglich, oft tritt dann aber auch eine Computersimulation bzw. numerische Methoden an die Stelle der analytischen Lösung. Das Arbeitsschema ist also heutzutage oft erweitert durch eine numerische oder Simulationskomponente:

$$\begin{aligned} \text{Experiment} &\Leftrightarrow \text{theoretisches Modell} \\ &\Leftrightarrow \text{analytische Rechnung und/oder Simulation/Numerik} \end{aligned}$$

Computermethoden kommen tatsächlich an verschiedenen Stellen und auf verschiedenen Ebenen dieses Schemas zum Einsatz:

1. **Simulationen oder “Computerexperimente”** bilden das gesamte System in mehr oder weniger *mikroskopischem* Detail ab. Dazu verwendete Verfahren wie Monte-Carlo oder Molekulardynamik (MD) Simulationen werden typisch bei Vielteilchensystemen eingesetzt.
2. **Numerische Analyse:** Ein physikalisches Problem wird zunächst theoretisch auf wenige Gleichungen oder Differentialgleichungen reduziert mit Hilfe theoretischer Standardmethoden. Die verbleibenden Gleichungen müssen dann aber oft numerisch gelöst oder analysiert werden. Beispiele hierfür sind selbstkonsistent zu lösende Mean-Field Gleichungen oder Hartree-Fock-Gleichungen in der statistischen Physik und der Festkörperphysik oder Formgleichungen für Tropfen, Vesikel oder Kapseln in der weichen Materie. Für solche numerischen Aufgaben bieten sich auch oft Programmmpakete wie Mathematica, Matlab oder Maple an.
3. **Symbolisches Rechnen:** Auch analytische Rechnungen und Umformungen selbst (Integrieren, Differenzieren, Lösen von DGLs, usw.) können mit Computerhilfe durchgeführt werden. Programmmpakete für solche symbolisch Rechnungen sind Mathematica, Maple oder Reduce.
4. **Datenanalyse/Visualisierung:** Numerische Methoden sind auch ganz allgemein wichtig bei

der Analyse und (graphischen) Aufbereitung von Daten, wobei sich dies sowohl auf experimentelle als auch auf Simulationsdaten beziehen kann. Hierfür gibt es Programme wie gnuplot, Origin, Maple, Mathematica oder Matlab, sowie Statistikprogramme.

Im Rahmen dieser Vorlesung werden wir uns hauptsächlich mit den Punkten 1 und 2 auseinandersetzen. Methoden, um Computersimulationen und numerische Analyseverfahren selbst zu programmieren, sollen hier dargestellt werden. Punkt 3 (symbolisches Rechnen) ist i.Allg. schwer selbst zu programmieren und man ist ohnehin stärker auf Programm pakete angewiesen. Auch für Punkt 4 ist es oft sinnvoller, Auswertungs- und Visualisierungsprogramme zu verwenden.

1.2 Geschichte

Die Geschichte der Computermethoden in der Physik ist natürlich auch eine Geschichte der zur Verfügung stehenden Hardware, die Computersimulationen in Größe und Schnelligkeit (bis heute) limitiert. Seit dem 2. Weltkrieg wurden leistungsfähige *elektronische* Computer entwickelt und auch sofort für physikalische Probleme und Simulationen eingesetzt. Ein wichtiger Ausgangspunkt war das Manhattan-Projekt, im Rahmen dessen auch viele numerische Berechnungen zur Atombombe durchgeführt wurden.

Die Geschichte der Rechenmaschinen reicht aber noch weiter vor das elektronische Zeitalter zurück. Bereits im 19. Jhd. hat Charles Babbage (1791-1871) erste *mechanische* Rechenmaschinen entworfen, die “difference engine” und die “analytical engine”. Die Differenzmaschine 1 (1832) wurde auch als Prototyp gebaut (Abb. 1.1); sie dient der Berechnung von Polynomen.

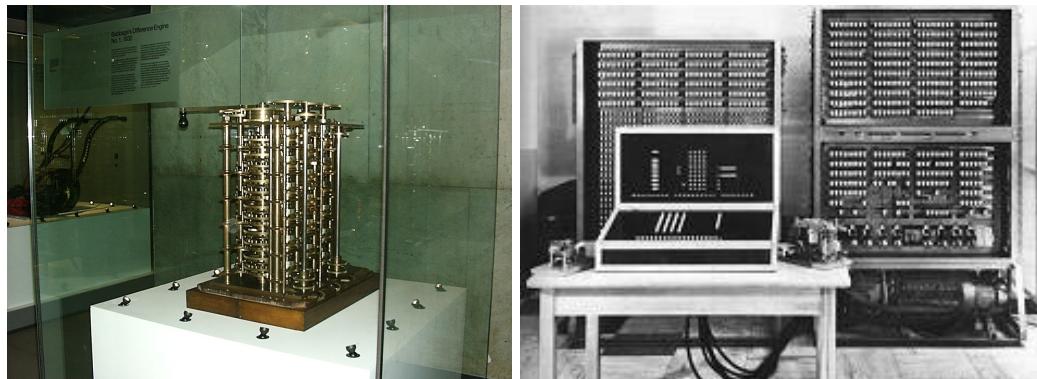


Abbildung 1.1: Links: Prototyp der Differenzmaschine 1 von Charles Babbage (1832) im London Science Museum. Rechts: Nachbau des Relaisrechner Z3 von Konrad Zuse im Deutschen Museum. (Quelle: Wikipedia).

Auch wichtige Konzepte des Computertheorie wurden bereits vor der Realisierung elektronischer Computer entwickelt. Beispielsweise wurde das Konzept des *Universalcomputers* (*Turingmaschine*) von Alan Turing 1936 formuliert [1]. Turing war maßgeblich an der Entschlüsselung des mit der Enigma kodierten deutschen Funkverkehrs im 2. Weltkrieg beteiligt.

Die erste fast elektronische Rechenmaschine (sie enthält noch elektrische Relais) ist die “Zuse Z3” von Konrad Zuse aus dem Jahr 1941 (Abb. 1.1). Die Z3 ist ein Binärcomputer mit Gleitkommaarithmetik. Sie ist programmierbar und beherrscht bedingte Sprünge, allerdings keine Schleifen. Auftraggeber war die Deutsche Versuchsanstalt für Luftfahrt zum Zwecke von aerodynamischen Berechnungen. Später entstand auch noch die Z4, die bereits mit einer Programmiersprache (“Plankalkül”) arbeitete.

Der erste voll elektronische Rechner war 1942 der Atanasoff-Berry-Computer. Er arbeitete bereits

im Binärsystem. Im Gegensatz zum Z3 oder ENIAC war er allerdings nicht frei programmierbar, sondern auf das Lösen linearer Gleichungssysteme beschränkt.



Abbildung 1.2: Links: ENIAC (1946): 17000 Elektronenröhren, 7200 Dioden, 27 Tonnen Gewicht, 167 m^2 Standfläche, Energieverbrauch 150 Kilowatt. (Quelle: Wikipedia). Rechts: MANIAC (im Vordergrund Nicholas Metropolis). Standfläche 1.85 m^2 , Energiebedarf 35 Kilowatt, 2400 Elektronenröhren und 500 Dioden. Lochstreifen oder Magnetband. Speicherung elektrostatisch (Kapazität: 1024 Wörter) oder auf Magnettrommel (Kapazität: 10000 Wörter). Eine Addition dauerte 80 Millisekunden, Multiplikation, Division eine Sekunde. Ausgabe erfolgte über Anelex-Drucker, Fernschreiber, auf Papierband oder Magnetstreifen.

Der erste programmierbare elektronische Rechner ENIAC (Electronic Numerical Integrator and Computer) war bereits ein Universalcomputer im Turingschen Sinne. Er wurde 1946 von der US Army (Ballistic Research Laboratory) für ballistische Rechnungen konstruiert. John von Neumann benutzte ENIAC dann aber als erstes für Berechnungen zur Atombombe im Manhattan Project. ENIAC wog 27 Tonnen und arbeitete noch im Dezimalsystem (siehe Abb. 1.2).

1945 formulierte John von Neumann auch die grundlegende Von-Neumann-Architektur von Computern, die aus Steuereinheit, arithmetischer Einheit und Speichereinheit besteht und so noch heute gültig ist. Daten und Programme werden gemeinsam im Arbeitsspeicher abgelegt. Insbesondere das Prinzip des im Computer gespeicherten Programms wurde in den darauffolgenden Jahren sehr erfolgreich umgesetzt.

Von 1948-1952 wurde unter der Leitung von Nicholas Metropolis in Los Alamos MANIAC I konstruiert, der das Prinzip des gespeicherten Programms verfolgte und u.a. zu Berechnungen für Nuklearwaffen verwendet wurde (Zustandsgleichungen für Materie unter extremen Bedingungen, Berechnungen zur Neutronendiffusion), aber auch zu zahlreichen anderen physikalischen Problemen aus Hochenergiephysik, nicht-linearer Dynamik, Hydrodynamik usw. Eine Liste der wichtigsten wissenschaftlichen Arbeiten mit MANIAC ist in [2] gegeben und in Abb. 1.3 abgedruckt. Vor allem Enrico Fermi und Edward Teller sahen den Nutzen von MANIAC und setzten ihn zur Lösung wichtiger physikalischer Probleme der Zeit ein [2]. Auf MANIAC I wurde von Metropolis *et al.* dann auch die erste 1953 veröffentlichte Monte-Carlo Simulation durchgeführt, in der die Zustandsgleichung eines Gases aus harten zweidimensionalen Scheiben bestimmt wurde [3]. Die Monte-Carlo Methode, also die Berechnung thermodynamischer Mittelwerte durch Sampling der Boltzmannverteilung mit Hilfe von Zufallszahlen, wurde 1947 von Nicholas Metropolis (in Zusammenarbeit mit Stanislaw Ulam und John von Neumann) erfunden und weiterentwickelt [4].

Ein weiterer Meilenstein war dann die Simulation des Fermi-Pasta-Ulam Modells gekoppelter anhar-

Pion-proton phase-shift analysis (Fermi, Metropolis; 1952)
 Phase-shift analysis (Bethe, deHoffman, Metropolis; 1954)
 Nonlinear coupled oscillators (Fermi, Pasta, Ulam; 1953)
 Genetic code (Gamow, Metropolis; 1954)
 Equation of state: Importance sampling (Metropolis, Teller; 1953)
 Two-dimensional hydrodynamics (Metropolis, von Neumann; 1954)
 Universalities of iterative functions (Metropolis, Stein, Stein; 1973)
 Nuclear cascades using Monte Carlo (Metropolis, Turkevich; 1954)
 Anti-clerical chess (Wells; 1956)
 The lucky numbers (Metropolis, Ulam; 1956)

Fig. 6. Scientific triumphs achieved with the MANIAC. Nick Metropolis was a co-author of the publications resulting from these studies except for the ones on nonlinear coupled oscillators and anti-clerical chess.

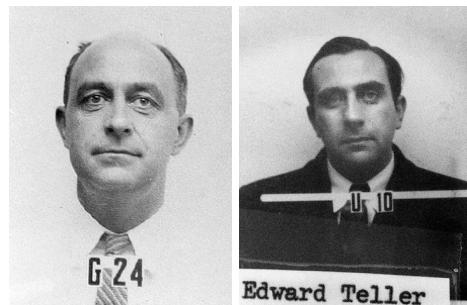


Abbildung 1.3: Links: Liste der wichtigsten wissenschaftlichen Arbeiten basierend auf MANIAC nach H.L. Anderson [2]. Mitte: Enrico Fermi (1901-1954), italienischer Physiker (Nobelpreis 1938). Rechts: Edward Teller (1908-2003) ungarisch-amerikanischer Physiker. Beide Abbildungen von ihren Los Alamos Badges. (Quelle: Wikipedia).

monischer Oszillatoren [5, 6]. Dies ist ein nichtlineares chaotisches System, von dem nicht klar war, ob es in einer mehr oder weniger zufällige Bewegung “thermalisiert” und ergodisch ist oder doch noch periodisches Verhalten zeigt, wie integrbare Modelle. Hier konnte mit Hilfe des Computers tatsächlich ein komplexes quasi-periodisches Verhalten gezeigt werden.

Das erste Ergebnis einer Molekulardynamik (MD) Simulation wurde 1957 von Alder und Wainwright veröffentlicht [7]. In den 50er Jahren war es nicht klar, ob ein Gas aus harten Kugeln kristallisieren kann. Dies war eine kontrovers diskutierte Frage: Uhlenbeck ließ wiederholt abstimmen auf Konferenzen mit durchaus namhaften Teilnehmern (Nobelpreisträgern), das Ergebnis dabei wahr mehrmals eine 50:50 geteilte Meinung [8, 9]. Diese Frage wurde dann von Alder und Wainwright mit Hilfe der ersten MD-Simulation auf dem Computer (einer UNIVAC bzw. IBM-704) entschieden: Sie konnten für ein zwei-dimensionales System aus harten Kugeln (also harte Scheiben) eine Kristallisation oberhalb einer kritischen Dichte numerisch nachweisen [7]. Harte Kugeln sind auch heute noch ein wichtiges Beispielsystem in der Computersimulation, das ungelöste Fragen bereithält. So ist gerade in zwei Dimensionen (also genau das Alder/Wainwright-System) die Natur des Phasenübergangs immer noch nicht ganz zweifelsfrei geklärt (ein kontinuierlicher Übergang vom Kosterlitz-Thouless Typ oder doch ein diskontinuierlicher Übergang).

Hier noch einmal einige Meilensteine:

- **Charles Babbage** (1791-1871): Entwurf erster mechanischer Rechenmaschinen (difference engine, analytical engine)
- **Alan Turing** (1912-1954): Konzept des Universalcomputers (Turingmaschine, 1936). Turing arbeitete im 2. Weltkrieg in Bletchley Park an der Entschlüsselung des deutschen Enigma-Codes.
- **Konrad Zuse** (1910-1995): Rechenmaschinen Z1 (mechanisch, 1938), Z3 (elektromechanisch, auf Relaisbasis, 1941), Z4 (Programmiersprache “Plankalkül”)
- **Atanasoff-Berry-Computer** (1942): erster voll elektronischer Rechner.
- **Colossus** (1943): In Bletchley Park gebaute Rechenmaschine auf Röhrenbasis zur Entzifferung des deutschen Lorenz-Schlüssels.
- **John von Neumann** (1903-1957): Er formulierte 1945 die Von-Neumann-Architektur von

Computern: Steuereinheit, arithmetischer Einheit, Speichereinheit. Daten und Programme werden gemeinsam im Arbeitsspeicher abgelegt.

- **ENIAC** (Electric Numerical Integrator and Computer, 1946): Erster programmierbarer Rechner auf Röhrenbasis, wurde in Los Alamos für militärische Rechnungen verwendet. Er wog 27 Tonnen und arbeitete noch im Dezimalsystem.
- **MANIAC I** (Mathematical Analyzer Numerical Integrator And Computer Model I, 1952): Der unter der Leitung von Nicholas Metropolis (1915-1999) konstruierte und in Los Alamos betriebene MANIAC I wurde zu Berechnungen für Nuklearwaffen verwendet. Er war aber auch der erste Rechner, der von Fermi und Teller für physikalische Simulationen eingesetzt wurde.
- **Monte Carlo Simulation** (1953): Ab 1947 entwickelt Nicholas Metropolis die Monte-Carlo Simulationsmethode. Die erste veröffentlichte physikalische Simulation wird von Nicholas Metropolis und anderen an MANIAC I durchgeführt und ist eine Monte Carlo Simulation zur Bestimmung der Zustandsgleichung eines Gases harter Kugeln (in 2 Raumdimensionen).
- **Fermi-Pasta-Ulam Problem** (1955): Ebenfalls am MANIAC I durchgeführt wurde die Simulation linearer gekoppelter anharmonischer Oszillatoren, wobei erstmals quasiperiodisches Verhalten gefunden wurde.
- **Molekulardynamik-Simulation** (1956): Die erste Molekulardynamik-Simulation ausgehend von den Bewegungsgleichungen harter Kugeln wurde von Alder und Wainwright durchgeführt.

1.3 Anwendungen

Wir wollen exemplarisch einige Anwendungen von Computerphysik anschneiden. Diese berühren vor allem Fragen, die sich einer direkten analytischen Lösung entweder prinzipiell (nichtlineare Probleme) oder auf Grund der nicht-Idealität realistischer Systeme (realistische Geometrien, Randbedingungen) entziehen.

In der klassischen Mechanik sind dies chaotische Systeme der nichtlinearen Dynamik. So ist beispielsweise jedes 3-Körper-Problem in der Mechanik nur noch numerisch lösbar, sofern die Wechselwirkungs Kräfte nicht gerade linear sind.

In der Elektrodynamik ist man in technischen Anwendungen oft mit partiellen Differentialgleichungen, z.B. Potentialprobleme in komplizierten Geometrien konfrontiert, die sich nicht mehr mit den Standardmethoden für idealisierte Geometrien wie Quader, Kugel oder Zylinder lösen lassen.

In der Hydrodynamik sind die relevanten partiellen Differentialgleichungen wie die Navier-Stokes Gleichung nicht-linear und daher auch nicht mehr analytisch lösbar, insbesondere wenn Turbulenz auftritt.

Auch in der Quantenmechanik werden analytische Lösungen selbst von 1-Teilchen Schrödingergleichungen schnell unmöglich, wenn die Geometrie und damit die Randbedingungen komplizierter werden oder die Potentiale nicht mehr in die einfachen Klassen stückweiser konstanter, harmonischer oder $1/r$ -Potentiale fallen. Ein einfaches Problem, das man schon numerisch lösen muss ist der eindimensionale anharmonische Oszillator. Die Probleme sind gravierender in der Quantenmechanik wechselwirkender Vielteilchensysteme, wie sie die Grundlage von Molekül- und Festkörperphysik bilden. Probleme, die quantitativ nur noch numerisch gelöst werden können, sind hier Molekülspektren und Bandstrukturen.

Auch in der statistischen Physik sind die interessanten Systeme mit vielen wechselwirkenden Teilchen oder Freiheitsgraden nicht mehr exakt analytisch lösbar, selbst wenn sie rein klassischer Natur sind: Ein wechselwirkendes reales Gas kann nur noch näherungsweise analytisch z.B. in einer Virialentwicklung behandelt werden. Insbesondere in der Theorie der Phasenübergänge und kritischen Punkten sind Simulationen wichtig, selbst wenn mächtige analytische Werkzeuge wie die Renormie-

rungsgruppe zur Verfügung stehen. Auch große biologische Systeme wie Proteine und deren Faltung lassen sich quantitativ nur mit Computermethoden analysieren.

Auch stochastische Prozesse, die durch stochastische Differentialgleichungen (Langevin-Gleichung) oder deterministische partielle Differentialgleichungen für Wahrscheinlichkeitsverteilungen (Fokker-Planck-Gleichung) beschrieben werden, lassen sich in den seltensten Fällen geschlossen analytisch lösen, so dass numerische Methoden wichtig sind. Exotischere Anwendungen umfassen dann auch stochastische Prozesse in der Finanzphysik, Modellierung von Prozessen auf Netzwerken und Graphen wie dem Internet oder auch soziales Verhalten in der Spieltheorie oder der Evolution.

1.4 Nutzen

Vor diesem Hintergrund kann man sich nun den generellen Nutzen der Computerphysik klarmachen. Die theoretische Lösung eines physikalischen Problems beginnt mit der Formulierung eines geeigneten Modells, das aber nur in seltenen Fällen unmittelbar analytisch gelöst werden kann, wie wir gerade gesehen haben.

Das Computerexperiment und andere Methoden können dazu dienen, vorhandene analytische Lösungen theoretischer Modelle nachzuprüfen. Dies dient auf der anderen Seite auch einer Überprüfung der Simulationsmethoden.

Dann können aber auch all die Fälle mit Computermethoden betrachtet werden, die nicht analytisch gelöst werden konnten. Ein einfaches Beispiel wäre mechanische Mehrteilchendynamik unter dem Einfluss der Gravitation. Das 2-Teilchenproblem ist einfach analytisch lösbar, das 3- (und mehr) Teilchenproblem bereits analytisch prinzipiell unlösbar, kann allerdings in Computersimulationen einfach und sehr genau numerisch gelöst werden (wichtige Anwendung: Mondlandung).

Mikroskopische Simulationen von Vielteilchensystemen gleichen “Computerexperimenten”, da wir es durchaus mit ähnlich großen Teilchenzahlen wie im realen Experiment zu tun haben. Solche Computerexperimente bilden nicht nur das reale System mikroskopisch nach sondern haben auch prinzipielle Vorteile gegenüber realen Experimenten. So erlauben mikroskopische Vielteilchensimulationen wie z.B. die Molekulardynamik einzelne Teilchen zu adressieren, was experimentell unmöglich ist. Dies erlaubt ganz neue Einblicke in das System, z.B. auf Nanometerlängenskalen, die mit experimentellen mikroskopischen Methoden noch gar nicht untersucht werden können. Im Idealfall könnten solche Computerexperimente irgendwann reale Experimente ersetzen. Beim biologischen Beispiel der Proteinfaltung wird allerdings klar, dass die zur Verfügung stehenden Computerressourcen bei weitem nicht ausreichen, um einen realen Faltungsvorgang auf der Zeitskala von s nachzusimulieren. Hier ist man im Computerexperiment auf Nanosekunden beschränkt.

Auf der anderen Seite stellt sich dann bei solchen großen Computerexperimenten natürlich auch die Frage, wo der Erkenntnisgewinn liegt, wenn die Natur in allen Einzelheiten “nachsimuliert” wird. Letztlich braucht es für echten Erkenntnisgewinn über die Mechanismen der Physik dann auch immer verifizierbare Theorien und Hypothesen. Nur so erhält man eine gewisse allgemeine Vorhersagekraft der Theorie.

1.5 Inhalt

Wir schließen mit einem kurzen Überblick über den weiteren Inhalt der Vorlesung:

Kap. 2 Zahlen und Fehler. Grundsätzliches zur diskreten Repräsentation von Zahlen und ihren Eigenheiten.

Inhalt: Zahldarstellungen, Rundungsfehler, Abbruchfehler, numerische Stabilität, Benford-sches Gesetz.

Kap. 3 Differentiation und Integration. Grundlage der Physik ist das Differentialkalkül; entsprechende numerische Methoden werden hier kurz eingeführt.

Inhalt: Ableitung, zweite Ableitung, Trapezregel, Mittelpunktsregel, Simpsonregel, Euler-McLaurin, Romberg, uneigentliche Integrale, Hauptwertintegrale, Anwendung: Kramers-Kronig-Relationen.

Kap. 4 Gewöhnliche Differentialgleichungen. Viele dynamische physikalische Probleme reduzieren sich auf das Lösen von gewöhnlichen Differentialgleichungen. Die wichtigsten Methoden und Prinzipien werden hier vorgestellt.

Inhalt: DGLn 1. und 2. Ordnung, Lösungsverfahren: Euler, Prädiktor-Korrektor, Runge-Kutta, Newtonsche Bewegungsgleichungen, Verlet, Leapfrog, adaptive Schrittweite.

Kap. 5 Molekulardynamik-Simulationen. Die wichtigste Anwendung gewöhnlicher Differentialgleichungen in der Computerphysik ist die MD-Simulation eines Vielteilchensystems. Hier wird zuerst das mikrokanonische Ensemble simuliert, indem die Newtonschen Bewegungsgleichungen gelöst werden für viele Teilchen. Abschließend werden Thermostaten für kanonische Ensembles besprochen.

Inhalt: Kräfte, Initialisierung, periodische Randbedingungen, Messungen, Verlet, Liouville-Operator, Paarverteilung $g(r)$, Anwendung: Lennard-Jones Fluid, Thermostaten.

Kap. 6 Partielle Differentialgleichungen.

Inhalt: Randbedingungen, Diskretisierung, Stabilität, Poisson-Gleichung, Wellengleichung, Diffusionsgleichung, Schrödingergleichung.

Kap. 7 Iterationsverfahren. Viele Probleme (nicht-lineare Gleichungen, Nullstellen) lassen sich als Fixpunktprobleme formulieren und können mit Iterationsverfahren gelöst werden. Sie tauchen in der Physik oft in Mean-Field Approximationen auf. In der Physik chaotischer Systeme ist die Iterationsdynamik auch von großem eigenständigem Interesse.

Inhalt: Mean-Field Theorien und Selbstkonsistenz, Intervallhalbierung, Regula Falsi, Newton-Raphson. Iterationen, Fixpunkte, Nullstellen, Nicht-lineare Gleichungen, Banachscher Fixpunktsatz, Fixpunkt-Bifurkationen und Chaos (Feigenbaum), Poincaré-Schnitte.

Kap. 8 Matrixdiagonalisierung, Eigenwertprobleme. Matrixdiagonalisierung und das Auffinden von Eigenwerten und Eigenvektoren ist in der numerischen Quantenmechanik essentiell. Die wichtigsten Verfahren werden kurz vorgestellt. Neben der Quantenmechanik gibt es aber auch andere Anwendungen, ein interessantes Beispiel stellt der PageRank-Algorithmus dar.

Inhalt: Jacobi-Rotation, Householder, QR-Zerlegung, Potenzmethode, Transfermatrix, Google PageRank, Diagonalisierung in der Quantenmechanik.

Kap. 9 Minimierung.

Inhalt: Schachtelung, Gradientenmethoden, konjugierte Gradienten.

Kap. 10 Erzeugung von Zufallszahlen, Zufallszahlengeneratoren. Für stochastische Verfahren wie Monte-Carlo Methoden oder auch für Zufallskräfte in stochastischen Bewegungsgleichungen werden Zufallszahlen benötigt.

Inhalt: Pseudo-Zufallszahlengeneratoren, linear kongruente Generatoren, Xorshift, Transformation von Verteilungen, Gaußverteilte Zufallszahlen.

Kap. 11 Monte-Carlo Integration und Monte-Carlo Simulationen. Monte-Carlo Methoden sind stochastischer Natur und unterscheiden sich damit grundlegend von deterministischen Methoden. Wir beginnen mit der Monte-Carlo Integration, wobei wir Importance-Sampling und Markov-Sampling einführen. Dann wird die Monte-Carlo Simulation mit dem Metropolis-Algorithmus am Beispiel des Ising-Modells vorgestellt. Abschließend werden Cluster-Algorithmen diskutiert.

Inhalt: Monte-Carlo Integration, Importance-Sampling, Markov-Sampling, Monte-Carlo Si-

mulation des Ising-Modells mit Metropolis-Algorithmus, Cluster-Algorithmen.

Kap. 12 Perkolation.

Kap. 13 Simulation stochastischer Bewegungsgleichungen. Systeme der statistischen Physik können dynamisch auch durch stochastische Bewegungsgleichungen beschrieben werden. Wir diskutieren die Brownsche Bewegung und die Langevin-Gleichung sowie die entsprechenden Fokker-Planck-Gleichungen für die Wahrscheinlichkeitsverteilungen.

Inhalt: Brownsche Bewegung, Langevin-Gleichung, Brownsche Dynamik und Langevin Simulation, Fokker-Planck-Gleichungen, Smoluchowski-Gleichung.

Was nicht behandelt wird, sind die Lösung linearer Gleichungssysteme und Interpolation, da diese Themen in Dortmund Teil der numerischen Mathematik sind bzw. bei Bedarf in fast allen Lehrbüchern (Numerical Recipes seien empfohlen) nachgearbeitet werden können.

1.6 Literatur

Die Vorlesung richtet sich nach keinem bestimmten Buch. Folgende Literatur ist hilfreich und teilweise Grundlage einzelner Kapitel:

1. Press *et al.*, *Numerical Recipes*, Cambridge University Press (free online editions of older versions available) [10, 11]
2. S.E. Koonin and D.C. Meredith, *Computational Physics*, Addison-Wesley [12]
3. W. Kinzel and G. Reents, *Physics by Computer*, Springer [13]
4. D. Frenkel and B. Smit, *Understanding Molecular Simulation*, Academic Press [14]
5. H. Gould, J. Tobochnik, W. Christian, *An Introduction to Computer Simulation Methods: Applications to Physical Systems*, Addison Wesley [15]
6. J.M. Thijssen, *Computational Physics*, Cambridge University Press [16]
7. D.P. Landau and K. Binder, *Monte Carlo Simulations in Statistical Physics*, Cambridge University Press [17]
8. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer [18]
9. R.W. Hamming, *Numerical Methods for Scientists and Engineers*, Dover [19]
10. S.H. Strogatz, *Nonlinear Dynamics and Chaos*, Westview Press [20]
11. G.H. Golub, C.F. van Loan, *Matrix Computation*, Johns Hopkins University Press [21]
12. M. Hjorth-Jensen, *Lecture Notes in Computational Physics*, University of Oslo, 2008 (Vorsicht: relativ viele Typos) [22]
13. R. Fitzpatrick, *Computational Physics*, University of Texas at Austin [23]
14. W. Krauth, *Statistical Mechanics: Algorithms and Computations*, Oxford University Press [24]

1.7 Literaturverzeichnis Kapitel 1

- [1] C. Moore. *A complex legacy*. Nature Phys. **7** (2011), 828–830.
- [2] H. L. Anderson. *Metropolis, Monte Carlo, and the MANIAC*. Los Alamos Science (1986), 96–108.
- [3] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller und E. Teller. *Equation of State Calculations by Fast Computing Machines*. J. Chem. Phys. **21** (1953), 1087–1092.

- [4] N. Metropolis. *The beginning of the Monte Carlo method*. Los Alamos Science **15** (1987), 125–130.
- [5] E. Fermi, J. Pasta und S. Ulam. *Studies of nonlinear problems*. LASL Report LA-1940 (1955).
- [6] T. Dauxois, M. Peyrard und S. Ruffo. *The Fermi–Pasta–Ulam ‘numerical experiment’: history and pedagogical perspectives*. Eur. J. Phys. **26** (2005), S3–S11.
- [7] B. Alder und T. Wainwright. *Phase Transition for a Hard Sphere System*. J. Chem. Phys. **27** (1957), 1208–1211.
- [8] C. Dellago und H. A. Posch. *Realizing Boltzmann’s dream: computer simulations in modern statistical mechanics*. In: *Boltzmann’s Legacy*. Hrsg. von G. Gallavotti, W. Reiter und J. Yngvason. Zuerich, Switzerland: European Mathematical Society Publishing House, Okt. 2008, 171–202.
- [9] G. Uhlenbeck. *Round Table on Statistical Mechanics*. In: *The many-body problem*. Hrsg. von J. Percus. London: Interscience Publishers/John Wiley, 1963. Kap. XXVIII, 493–509.
- [10] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [12] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.
- [13] W. Kinzel und G. Reents. *Physics by Computer*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
- [14] D. Frenkel und B. Smit. *Understanding Molecular Simulation*. 2nd. Orlando, FL, USA: Academic Press, Inc., 2001.
- [15] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [16] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [17] D. P. Landau und K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2005.
- [18] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis*. 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.
- [19] R. W. Hamming. *Numerical Methods for Scientists and Engineers*. 2nd. New York, NY, USA: Dover Publications, Inc., 1986.
- [20] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Studies in nonlinearity. Westview Press, 2008.
- [21] G. H. Golub und C. F. Van Loan. *Matrix Computations*. 3rd. Johns Hopkins Studies in the Mathematical Sciences. Baltimore, Maryland, USA: Johns Hopkins University Press, 1996.
- [22] M. Hjorth-Jensen. *Computational Physics (Skript)*. Oslo: University of Oslo, 2012.
- [23] R. Fitzpatrick. *Computational Physics (Skript)*. Austin, Texas: The University of Texas at Austin, 2012.
- [24] W. Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Statistical, Computational, and Theoretical Physics. Oxford University Press, 2006.

2 Zahlen und Fehler

Literatur zu diesem Teil:

Zu Zahlen und Fehlern: Numerical Recipes [1, 2], Stoer [3], Hamming [4].

Zum Benfordschen Gesetz: Hamming [4], Kapitel 2.8, und [5, 6, 7, 8].

2.1 Zahldarstellungen

In diesem Abschnitt werden die verschiedenen (diskreten) Zahldarstellungen im Computer diskutiert, insbesondere die floating point Darstellung.

Im Gegensatz zum kontinuierlichen Zahlraum des Körpers der reellen Zahlen ist der Zahlenraum im Computer notwendigerweise *diskret*. Es gibt verschiedene **Darstellungen**:

(i) ganze Zahlen (**integer**)

Im binären System, z.B. mit 16 Bits, stellt 1 Bit das Vorzeichen dar und die verbleibenden 15 Bits ($2^{15} - 1 = 32767$) den Betrag, so dass man alle integers $i = -32767, \dots, -1, 0, 1, \dots, 32767$ darstellen kann.

(ii) Festkomma (**fixed point**)

Hier hat man eine feste Zahl von Nachkommastellen und damit bei gegebenem Speicherplatz auch eine feste Zahl von Vorkommastellen: 12.2300 und 1234.4567 sind beides erlaubte Festkommazahlen in einem Format mit jeweils 4 Vor- und Nachkommastellen. Festkommazahlen sind im Wesentlichen integers, die mit einem **Skalenfaktor** b^E mit *festem Exponenten* E ($E < 0$) malgenommen werden. Dabei ist die **Basis** $b = 10$ im Dezimal- oder $b = 2$ im Binärsystem. Da Ergebnisse von Rechenoperationen sowohl die Zahl der Vorkomma- als auch der Nachkommastellen schnell überschreiten, kommt es zu Overflow- bzw. Rundungsfehlern.

(iii) Gleitkomma (**floating point**)

Overflow- und Rundungsfehler werden in diesem Format minimiert. Eine Zahl wird dargestellt als

$$\underbrace{-}_{\text{Vorzeichen } S} \quad \underbrace{0.1234567}_{\text{Mantisse } M} \quad \cdot \quad \underbrace{10^2}_{\text{Skalenfaktor } b^E}. \quad (2.1)$$

Der Wert der Zahl ist also $S \cdot M \cdot b^E$. Die **Mantisse** ist dabei eine Festkommazahl mit einer Vorkommastelle und einer festen Zahl von Nachkommastellen (die vom Speicherplatz abhängt und auch **Mantissenlänge** genannt wird). Die floating point Darstellung ist die bei Weitem bevorzugte, da sowohl sehr große als auch sehr kleine Zahlen mit gleicher relativer Genauigkeit dargestellt werden können. Die Basis im Skalenfaktor b^E hängt wieder vom **Zahlsystem** ab: $b = 2$ im Binärsystem, $b = 10$ im Dezimalsystem oder $b = 16$ im Hexadezimalsystem. Im Computer ist die interne Darstellung wegen der elektronischen Bauelemente auf das Binärsystem ausgelegt, während in höheren Sprachen die Darstellung dann normalerweise dezimal ist. Man schreibt das System z.B. auch als Subskript, also :

$$18.5_{10} = 10010.1_2.$$

Einige wichtige Eigenschaften der **floating point** Darstellung sind:

1) Diskrete Zahlen sind **nicht äquidistant**:

Der Abstand zwischen "benachbarten" Zahlen wächst mit dem Exponenten E ,
Beispiel: Für $b = 10$ und eine Mantisse $0.xxx$ der Länge 3 sind

$$\begin{aligned}x_1 &= 1.234 \cdot 10^E, \\x_2 &= 1.235 \cdot 10^E\end{aligned}$$

benachbart mit einem $\Delta x = 0.001 \cdot 10^E$, das von E abhängt. Also sind nur Zahlen innerhalb jeder Dekade (mit gleichem E) gleichverteilt.

2) Bei Rechnungen treten **Rundungsfehler** auf, z.B. bei Mantissen der Länge 3 und Basis 10,

$$\begin{aligned}x &= 0.101 \cdot 0.101 = 0.010\cancel{2}01, \\rd(x) &= 0.102 \cdot 10^{-1},\end{aligned}$$

wobei $rd(x)$ die gerundete Zahl bezeichnet. Die letzte Ziffer der Mantisse wird gerundet. Damit entsteht für eine Zahl $x \sim 10^E$ typischerweise ein Fehler

$$|rd(x) - x| \sim 10^{E-\text{Länge } M}.$$

Bei floating point Zahlen ist daher nur die Angabe des **relativen Fehlers** ε sinnvoll (weil unabhängig von E):

$$\varepsilon = \frac{|rd(x) - x|}{|x|} \sim 10^{-\text{Länge } M} \quad (2.2)$$

Entsprechend unterscheidet man bei jeder numerischen Rechnung $f = f(x)$ zwischen

$$\begin{aligned}\text{relativem Fehler} &= \frac{|rd(f) - f|}{|f|}, \\ \text{absolutem Fehler} &= |rd(f) - f|\end{aligned} \quad (2.3)$$

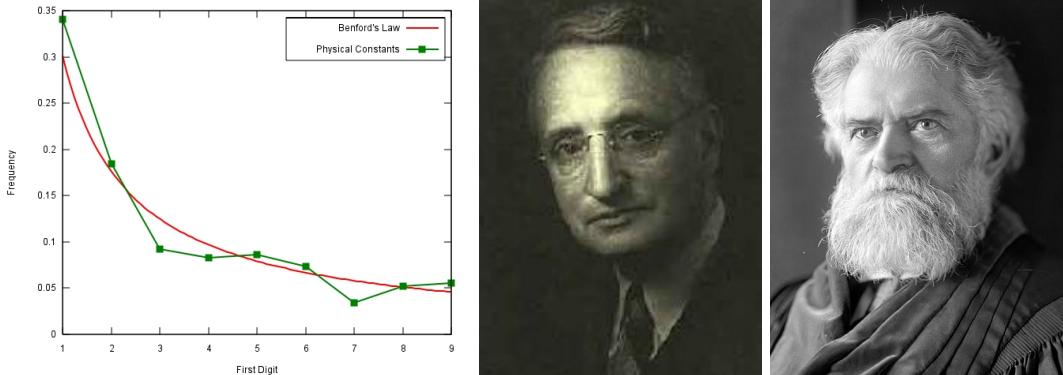


Abbildung 2.1: Links: Verteilung der führenden signifikanten Ziffern von physikalischen Konstanten (grün) im Vergleich zum Benfordschen Gesetz Gl. (2.4) (rot) (Quelle: Wikipedia). Mitte: Frank Benford (1883-1948), Physiker bei General Electric. Rechts: Simon Newcomb (1835-1909), Astronom und Mathematiker.

3) Daneben gibt es noch mehr oder weniger "kuriose" Eigenschaften:

- a) Da diskrete Zahlen x einen minimalen Abstand Δx haben, gibt es in möglichen Funktionswerten $f(x)$ "Lücken" Δf :

$$\Delta f \approx f'(x)\Delta x.$$

Hier ist Vorsicht geboten, wenn $|f'(x)|$ groß ist, dann sind auch die Lücken in $f(x)$ groß.

b) Noch kurioser ist folgender Befund: Theoretisch sind alle Gleitkomma-Mantissen M gleichverteilt, praktisch beobachtet man jedoch häufig *keine* Gleichverteilung!?
Beispiele sind:

- Betrachtet man die führenden signifikanten Ziffern von Naturkonstanten, treten die Ziffern 1,2 oder 3 viel häufiger auf, nämlich in 60% der Fälle! Siehe Abb. 2.1
- Simon Newcomb (1835-1909) hat 1881 festgestellt, dass Logarithmentafeln bei Seiten mit 1, 2 oder 3 viel häufiger abgegriffen sind. Das heißt, Logarithmen mit diesen führenden Ziffern werden öfter nachgeschaut, solche Zahlen kommen offenbar häufiger vor?! (Moderne Version von Thomas Jech: “When the 1 key on my old computer gave out I was not surprised”)

Diese überraschenden empirischen Befunde wurde von Frank Benford (1883-1948) im Jahr 1938 im sogenannten **Benfordschen Gesetz** quantifiziert [5].

2.2 Benfordsches Gesetz

Das Benfordsche Gesetz für die Verteilung führender Ziffern von Zahlen in Datensätzen (und zwei äquivalente Sätze) wird hergeleitet und Anwendungen diskutiert.

Das Benfordsche Gesetz [5] macht eine interessante Aussage über die Wahrscheinlichkeit, bei Zahlen in empirischen Datensätzen eine bestimmte signifikante führende Ziffer d in einer floating point Darstellung in der Basis b zu finden. Es ist nicht von zentraler Wichtigkeit für die Computerphysik oder Numerik, aber zeigt doch, dass auch scheinbar “trockene” Themen wie die Verteilung von Zahlen in Datensätzen manchmal Interessantes und Überraschendes zu bieten haben.

Benford selbst hat das Gesetz an Hand einer ganzen Reihe empirische Datensätze gefunden: Entwässerungsgebiete von 335 Flüssen (A), Einwohnerzahlen (B), physikalische Konstanten (C und siehe Abb. 2.1), Zahlen von den Titelseiten von Zeitungen (D), Zahlen aus Readers Digest Artikeln (M) oder American Football League Resultaten (P), siehe Abb. 2.2

Das Benfordsche Gesetz besagt, dass die Wahrscheinlichkeit $p_{\text{fZ}}(d)$ für eine führende Ziffer d durch die **Benford-Verteilung** gegeben ist:

$$p_{\text{fZ}}(d) = \ln \left(\frac{d+1}{d} \right) \frac{1}{\ln b} = \log_b \left(\frac{d+1}{d} \right) \quad (2.4)$$

Siehe auch Abb. 2.2 für $b = 10$. Wir wollen zunächst zwei äquivalente Aussagen ableiten.

Das Benfordsche Gesetz ist zum einen äquivalent dazu, dass in floating point Darstellung die Mantissen M der sogenannten **reziproken Verteilung** folgen:

$$p_{\text{Man}}(M) = \frac{1}{M \ln b} \quad \left(\frac{1}{b} \leq M < 1 \right). \quad (2.5)$$

Beweis:

Die Wahrscheinlichkeit, bei einer Mantissenverteilung $p_{\text{Man}}(M)$, die führende Ziffer d zu finden, ist genau die Wahrscheinlichkeit, dass M zwischen d/b und $(d+1)/b$ liegt, also

$$p_{\text{fZ}}(d) = \int_{d/b}^{(d+1)/b} dM p_{\text{Man}}(M) = \ln \left(\frac{d+1}{d} \right) \frac{1}{\ln b} \quad (2.6)$$

und damit wieder (2.4).

TABLE I
PERCENTAGE OF TIMES THE NATURAL NUMBERS 1 TO 9 ARE USED AS FIRST DIGITS IN NUMBERS, AS DETERMINED BY 20,229 OBSERVATIONS

Group	Title	First Digit									Count
		1	2	3	4	5	6	7	8	9	
A	Rivers, Area	31.0	16.4	10.7	11.3	7.2	8.6	5.5	4.2	5.1	335
B	Population	33.9	20.4	14.2	8.1	7.2	6.2	4.1	3.7	2.2	3259
C	Constants	41.3	14.4	4.8	8.6	10.6	5.8	1.0	2.9	10.6	104
D	Newspapers	30.0	18.0	12.0	10.0	8.0	6.0	6.0	5.0	5.0	100
E	Spec. Heat	24.0	18.4	16.2	14.6	10.6	4.1	3.2	4.8	4.1	1389
F	Pressure	29.6	18.3	12.8	9.8	8.3	6.4	5.7	4.4	4.7	703
G	H.P. Lost	30.0	18.4	11.9	10.8	8.1	7.0	5.1	5.1	3.6	690
H	Mol. Wgt.	26.7	25.2	15.4	10.8	6.7	5.1	4.1	2.8	3.2	1800
I	Drainage	27.1	23.9	13.8	12.6	8.2	5.0	5.0	2.5	1.9	159
J	Atomic Wgt.	47.2	18.7	5.5	4.4	6.6	4.4	3.3	4.4	5.5	91
K	n^{-1}, \sqrt{n}, \dots	25.7	20.3	9.7	6.8	6.6	6.8	7.2	8.0	8.9	5000
L	Design	26.8	14.8	14.3	7.5	8.3	8.4	7.0	7.3	5.6	560
M	Digest	33.4	18.5	12.4	7.5	7.1	6.5	5.5	4.9	4.2	308
N	Cost Data	32.4	18.8	10.1	10.1	9.8	5.5	4.7	5.5	3.1	741
O	X-Ray Volts	27.9	17.5	14.4	9.0	8.1	7.4	5.1	5.8	4.8	707
P	Am. League	32.7	17.6	12.6	9.8	7.4	6.4	4.9	5.6	3.0	1458
Q	Black Body	31.0	17.3	14.1	8.7	6.6	7.0	5.2	4.7	5.4	1165
R	Addresses	28.9	19.2	12.6	8.8	8.5	6.4	5.6	5.0	5.0	342
S	$n^1, n^2, \dots, n!$	25.3	16.0	12.0	10.0	8.5	8.8	6.8	7.1	5.5	900
T	Death Rate	27.0	18.6	15.7	9.4	6.7	6.5	7.2	4.8	4.1	418
Average		30.6	18.5	12.4	9.4	8.0	6.4	5.1	4.9	4.7	1011
Probable Error		± 0.8	± 0.4	± 0.4	± 0.3	± 0.2	± 0.2	± 0.2	± 0.2	± 0.3	—

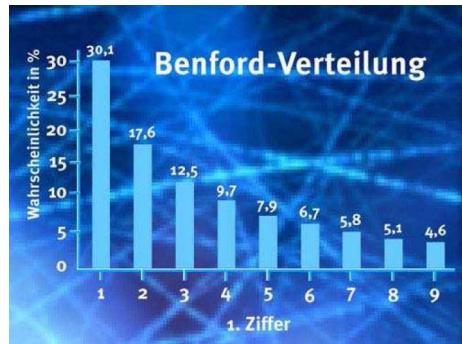


Abbildung 2.2: Links: Benfords Daten aus [5]. Die Daten folgen in guter Approximation der Benford-Verteilung (2.4) für $b = 10$ (Rechts).

Zum anderen ist das Benfordsche Gesetz zur Aussage äquivalent, dass die Logarithmen $\ln M$ der Mantissen *gleichverteilt* sind.

Beweis:

Mit $x \equiv \ln M$ und $\frac{dx}{dM} = \frac{1}{M}$ folgt aus (2.5) für die Verteilung der Logarithmen x :

$$p_{\ln M}(x) = p_{\text{Man}}(M) \frac{dM}{dx} = p_{\text{Man}}(M) M = \frac{1}{\ln b} = \text{const.}$$

In ähnlicher Form hatte Newcomb 1881 des Benfordsche Gesetz formuliert. Aus der verstärkten Abnutzung der Logarithmentafeln bei den führenden Ziffern stellte er fest, dass nicht etwa die Mantissen selbst gleichverteilt sind, sondern deren Logarithmen.

Wie kann man sich nun eine dieser äquivalenten Formen des Benfordschen Gesetzes erklären? Dazu werden wir mehrere Beweise betrachten. Die Benfordsche Verteilung gilt natürlich nicht für jede beliebige Verteilung von Daten. Daher ist es hier entscheidend, wie plausibel und allgemein sich die Annahmen über die Entstehung der Datensätze fassen lassen, unter denen sich ein Beweis führen lässt.

1. Beweis: Skalenargument

Das Benfordsche Gesetz gilt für einheitenbehaftete Daten. Wenn ein solches Gesetz gilt, sollte es in allen Einheitensystemen gelten. Dann muss die Verteilung $p(x)$ der Daten x *invariant unter Umskalierung* sein:

$$p(kx) = f(k)p(x). \quad (2.7)$$

Aus der Normierung folgt:

$$\int dx p(x) = 1 \Rightarrow \int dx p(kx) = 1/k \Rightarrow f(k) = 1/k. \quad (2.8)$$

Differenzieren von (2.7) nach k bei $k = 1$ liefert dann eine Differentialgleichung für $p(x)$, die leicht

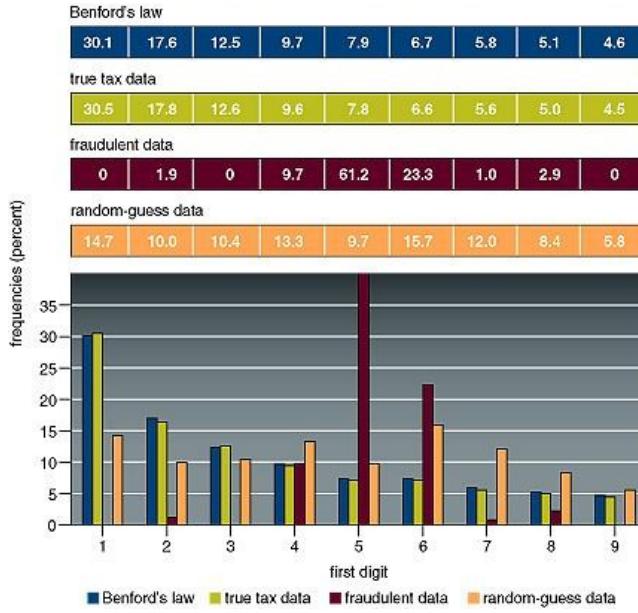


Abbildung 2.3: Links: Benfordsches Gesetz und Steuerdaten, aus [7].

zu lösen ist,

$$\partial_k|_{k=1} p(kx) = xp'(x) = -p(x) \Rightarrow p(x) = \text{const } \frac{1}{x}, \quad (2.9)$$

und auf eine reziproke Verteilung der Daten führt. Nach diesem Beweis sollte das Benfordsche Gesetz also nur für reziprok verteilte Daten x gelten; tatsächlich gilt es sehr viel allgemeiner, was die anderen Beweise zeigen.

Aus $p(x)$ kann man nun die Verteilung $p_{\text{Man}}(M)$ der Mantissen $M(x)$ von $x = M(x)b^{E(x)}$ in einer floating point Darstellung in Basis b gewinnen:

$$p_{\text{Man}}(M) = \sum_{x \text{ mit } M(x)=M} p(x) = \sum_E p(Mb^E) \stackrel{(2.7), (2.8)}{=} \sum_E b^{-E} p(M) = \text{const } p(M) \stackrel{(2.9)}{=} \text{const } \frac{1}{M}.$$

Die Mantissen folgen also (bis auf die Normierung) der gleichen reziproken Verteilung $p_{\text{Man}}(M) \propto 1/M$. Mit der Normierung der Mantissenverteilung auf dem Intervall $1/b \leq M < 1$ ergibt sich genau die reziproke Verteilung (2.5) und damit das Benfordsche Gesetz.

2. Beweis: Multiplikationsargument

Diesen Beweis werden wir nicht im Detail ausführen, sondern nur die Beweisidee skizzieren. Diese beruht auf zwei wichtigen Eigenschaften der reziproken Verteilung.

Die erste Eigenschaft wird im Hamming-Buch [4], Kapitel 2.8, gezeigt:

1) Wenn zwei Zufallsmantissen multipliziert werden, von denen *eine* der reziproken Verteilung folgt, ist die Mantisse des Produktes auch reziprok verteilt.

Dies zeigt eine gewisse ‘‘Persistenz’’ der reziproken Mantissenverteilung: Wenn Daten/Zahlen durch *wiederholte Multiplikation* gewonnen werden, sind deren Mantissen reziprok verteilt, wenn nur *eine* Mantisse reziprok verteilt war.

Die zweite Eigenschaft wird auch im Hamming-Buch [4], Kapitel 2.8, spezieller gezeigt und ist in allgemeiner Form in [6] zu finden:

2) Mantissen großer Produkte von Zahlen aus identischen und unabhängigen aber sonst beliebigen

Verteilungen nähern sich wieder der reziproken Verteilung an.

Diese zweite Eigenschaft zeigt, wie eine reziproke Mantissenverteilung leicht zustande kommen kann durch *wiederholte Multiplikation* von Zahlen, die aus der gleichen Verteilung gezogen werden.

Beides zusammengenommen macht plausibel, warum man sehr oft bei einer Benford-Verteilung landet, wenn Daten durch Multiplikation (oder Division) entstehen.

3. Beweis: Kombinationsargument

Eine modernere Beweisidee des Benfordschen Gesetzes geht auf Hill zurück [7] und beruht auf der Idee, dass das Benfordsche Gesetz dann realisiert ist, wenn Daten aus *vielen unabhängigen Verteilungen kombiniert* werden: Wenn viele Verteilungen zufällig und unabhängig ausgewählt werden und aus jeder Verteilung zufällig Stichproben gezogen werden, dann konvergiert die Verteilung der führenden Ziffern der kombinierten Stichprobe gegen die Benford-Verteilung, auch wenn die einzelnen Verteilungen dem Benfordschen Gesetz nicht genau folgen.

Die **Anwendungen** des Benfordschen Gesetzes liegen oft darin, dass Abweichungen auf eine gewisse “Nicht-Zufälligkeit” der Datensätze hindeuten. So wird das Benfordsche Gesetz zum Beispiel von Steuerbehörden benutzt, um Hinweise auf betrügerische Steuerdatensätze zu bekommen [7], siehe Abb. 2.3. Auch bei der iranischen Wahl 2009 gab es einige interessante Abweichungen vom Benfordschen Gesetz bei der Stimmenauszählung [8].

2.3 Fehler

In der Numerik spielen a) Rundungsfehler und b) Abbruchfehler von Algorithmen eine zentrale Rolle. Beide Fehlerarten werden hier eingeführt und Stabilität diskutiert.

2.3.1 Rundungsfehler

Wir hatten bereits in Kapitel 2.1 gesehen, dass der diskrete Zahlenraum im Computer zwangsläufig zu **Rundungsfehlern** führt.

Der Zusammenhang zwischen dem **relativen Fehler** ε und der gerundeten Zahl $rd(x)$ ist (siehe 2.2)

$$rd(x) = x(1 + \varepsilon).$$

Für floating point Zahlen tritt der Rundungsfehler in der letzten Stelle der Mantisse M auf und ist damit gleichverteilt im Intervall

$$|\varepsilon| \leq \frac{b}{2} b^{-\text{Länge } M-1} = \frac{1}{2} b^{-\text{Länge } M} \quad (2.10)$$

Als Beispiel betrachten wir eine Zahl $0.xxxx_{10}$ in Dezimaldarstellung mit $b = 10$ und Mantissenlänge 4. Rundung in der letzten Stelle erzeugt einen relativen Fehler $|\varepsilon| \leq 5 \cdot 10^{-5}$.

Diese permanenten kleinen Rundungsfehler ergeben in vielen numerischen Rechnungen und Simulationen in der Computerphysik ein **numerisches Rauschen**, das normalerweise klein ist, aber dessen man sich trotzdem bewusst sein sollte.

Rundungsfehler pflanzen sich bei Anwendung einer Funktion $f(x)$ auf eine fehlerbehaftete Zahl x fort. Taylorentwicklung ergibt die Standardformeln zur **Fehlerfortpflanzung**:

$$\begin{aligned} f(x(1 + \varepsilon)) &\approx f(x) + f'(x)x\varepsilon \\ \varepsilon_f &= \left| \frac{xf'(x)}{f(x)} \right| \varepsilon_x \end{aligned} \quad (2.11)$$

Besonders schlimm ist der Fall, wo $f(x) \approx 0$, die sogenannte **Auslöschung**.

Ein wichtiges **Beispiel** ist die Subtraktion $f(x, y) = x - y$ mit

$$\varepsilon_{x-y} = \left| \frac{x}{x-y} \right| \varepsilon_x + \left| \frac{y}{x-y} \right| \varepsilon_y,$$

wo der relative Fehler der Differenz unbegrenzt wächst für $x \approx y$.

Beim numerischen Rechnen gilt daher ganz allgemein

Rundungsfehler (besonders bei Auslöschung) möglichst vermeiden

Dies kann oft durch geschicktes analytisches Umformen erreicht werden.

Ein **Beispiel** dazu ist das Lösen der quadratischen Gleichung $x^2 + px + q = 0$. Mittels pq-Formel sind die Lösungen

$$x_{\pm} = -\frac{p}{2} \pm \left(\frac{p^2}{4} - q \right)^{1/2} \quad (2.12)$$

Für $|4q| \ll p^2$ gibt es unter der Wurzel eine Auslöschung bei naiver Berechnung mit (2.12), und zwar für x_+ wenn $p > 0$ bzw. für x_- wenn $p < 0$. Um diese Auslöschung zu umgehen kann man die Identität

$$x_+ x_- = q \quad (2.13)$$

benutzen, um eine Rechenmethode *ohne* Auslöschung anzugeben (für $p > 0$, der andere Fall analog): Berechne zunächst $x_- \approx -p$ mit (2.12), wobei keine Auslöschung auftritt, und berechne dann $x_+ = q/x_- \approx -q/p$ mittels (2.13), um die Auslöschung zu vermeiden.

2.3.2 Abbruchfehler

Viele numerische Verfahren beruhen auf Algorithmen, in denen eine N -malige Iteration/Rekursion auftritt oder in denen ein Kontinuum diskretisiert wird in kleine Intervalle Δx . In solchen Algorithmen treten **Abbruchfehler** auf, weil $N < \infty$ bzw. $\Delta x > 0$. Diese Abbruchfehler sind die zweite wichtige Fehlerklasse, die bei numerischen Rechnungen auftritt.

Als **Beispiel** betrachten wir die Berechnung von e^x mittels der Potenzreihe $e^x = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$. Daraus ergibt sich das naheliegende Verfahren, e^x zu berechnen, indem die Potenzreihe nach N Termen abgebrochen wird:

$$e^x \approx \sum_{n=0}^{N-1} \frac{1}{n!} x^n$$

Das führt zu einem Abbruchfehler $\sum_{n=N}^{\infty} \frac{1}{n!} x^n \leq \frac{1}{N!} x^N e^x$, den man hier nach oben abschätzen kann. Mit einer solchen Abschätzung kann man nun sofort ein N angeben, bis zu dem summiert werden muss, damit eine *vorgegebene* Genauigkeit kontrolliert erreicht wird.

Eine zweite wichtige Forderung in der numerischen Mathematik ist daher

Abbruchfehler analytisch abschätzen, um sie zu kontrollieren

2.3.3 Stabilität

Allerdings ist ein gewisses “numerisches Rauschen” durch Rundungs-/Abbruchfehler immer unvermeidlich in einer Rechnung oder Simulation. Daher ist darüberhinaus noch folgende dritte Forderung wichtig:

Numerische Verfahren müssen **stabil** sein gegenüber kleinem Rauschen

Als **Beispiel** betrachten wir dazu die Lösung der Differentialgleichung $y' = -y$ mit $y(0) = 1$. Die analytische Lösung ist natürlich $y(x) = e^{-x}$. Wir lernen noch mehrere numerische Lösungsverfahren genauer kennen, z.B. auch das **Euler-Verfahren**, das auf der Diskretisierung der x -Koordinate $y_n \equiv y(nh)$ ($x = nh$) mit einem kleinen $h = \Delta x \ll 1$ beruht. Wir werden zwei naheliegende Euler-artige Verfahren in Bezug auf Stabilität vergleichen.

1) Original Euler-Verfahren:

Aus

$$y_{n+1} - y_n \approx y'_n h = -y_n h$$

(Fehler rechts $\mathcal{O}(h^2)$ nach Taylor) kann man eine Rekursion

$$y_{n+1} = y_n(1 - h) \quad (2.14)$$

gewinnen. Dies ist das Euler-Verfahren. Mit dieser Rekursion und beginnend bei der Anfangsbedingung $y_0 = 1$ berechnen wir dann $y_n = (1 - h)^n$ als Lösung unserer DGL im Computer. Offensichtlich konvergiert diese numerische Lösung gegen 0 solange $h < 2$. Dann konvergiert sie auch tatsächlich gegen die analytische Lösung wegen

$$y_n = (1 - h)^n = \left(1 - \frac{x}{n}\right)^n \xrightarrow{n \rightarrow \infty} e^{-x}.$$

D.h. für $h < 2$ ist die numerische Lösung stabil.

2) Verbesserungsvorschlag (?):

Wir könnten auch ein “symmetrisches Euler-Verfahren” benutzen mit

$$y_{n+1} - y_{n-1} \approx 2y'_n h = -2y_n h,$$

was rechts einen kleineren Fehler $\mathcal{O}(h^3)$ hat als das Original Euler-Verfahren. Dies führt auf eine Rekursion

$$y_{n+1} + 2hy_n - y_{n-1} = 0. \quad (2.15)$$

Was macht der Computer, wenn wir diese Rekursion iterieren mit Startwerten $y_0 = 1$ und einem weiteren Startwert $y_1 = y_0 e^{-h} \approx y_0(1 - h)$? Dazu versuchen wir die Rekursion (2.15) exakt zu lösen, was bei dieser *Differenzengleichung* etwas schwieriger ist als oben bei (2.14). Für solche linearen Differenzengleichungen macht man allgemein einen Lösungsansatz $y_n = \alpha^n$, woraus sich für α aus (2.15) eine quadratische Gleichung mit zwei Lösungen ergibt:

$$\alpha^2 + 2h\alpha - 1 = 0 \Rightarrow \alpha_{\pm} = -h \pm (h^2 + 1)^{1/2} \approx \pm 1 - h$$

Die allgemeine Lösung der linearen Differenzengleichung (2.15) ist dann die Superposition beider Lösungen

$$y_n = C_+ \alpha_+^n + C_- \alpha_-^n. \quad (2.16)$$

Offensichtlich konvergiert aber nur der Lösungsteil $\alpha_+^n \approx (1 - h)^n \xrightarrow{n \rightarrow \infty} e^{-x}$ gegen die analytische Lösung. Dies ist auch die Lösung, wenn wir *genau* mit $y_0 = 1$ und $y_1 = \alpha_+ \approx 1 - h$ starten. Sobald wir die numerische Rekursion nun aber so starten, dass am Anfang $y_0 = 1$ und $y_1 = \alpha_+(1 + \varepsilon) \neq \alpha_+$ (z.B. $y_1 = 1 - h$) mit einer kleinen Abweichung von α_+ und damit $C_- \neq 0$ gilt, bekommen wir ein Stabilitätsproblem. Dann hat die numerische Lösung auch einen Anteil $C_- \alpha_-^n \approx (-1)^n (1 + h)^n$, der a) oszilliert und b) wegen $|\alpha_-| > 1$ (auch für beliebig kleines h) immer weiter anwächst und irgendwann den eigentlich erwünschten Anteil $C_+ \alpha_+^n$ mit $|\alpha_+| < 1$ dominiert. Das heißt aber, die Rekursion (2.15) ist numerisch instabil und damit keine wirkliche Verbesserung.

2.4 Literaturverzeichnis Kapitel 2

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing.* 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis.* 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.
- [4] R. W. Hamming. *Numerical Methods for Scientists and Engineers.* 2nd. New York, NY, USA: Dover Publications, Inc., 1986.
- [5] F. Benford. *The Law of Anomalous Numbers.* Proceedings of the American Philosophical Society **78** (1938), 551–572.
- [6] N. Hüngerbühler. *Benfords Gesetz über führende Ziffern : Wie die Mathematik Steuersündern das Fürchten lehrt.* 2007.
- [7] T. Hill. *The First Digit Phenomenon.* American Scientist **86** (1998), 358.
- [8] B. F. Roukema. *A first-digit anomaly in the 2009 Iranian presidential election.* J. Appl. Stat. **41** (Jan. 2014), 164–199.

2.5 Übungen Kapitel 2

1. Rundungsfehler:

Berechnen Sie die folgenden Ausdrücke numerisch zunächst direkt nach Formel. Suchen Sie dann nach einem numerischen Rechenweg, der Auslöschung vermeidet. Vergleichen Sie die relativen Fehler.

a) $\frac{1}{\sqrt{x}} - \frac{1}{\sqrt{x+1}}$ für große $x \gg 1$

b) $\frac{1 - \cos x}{\sin x}$ für kleine $x \ll 1$

c) $\ln(a+x) - \ln x$ für große $x \gg a$

2. Stabilität:

Implementieren Sie die obigen Rekursionen (2.14) (Original Euler) und (2.15) (modifizierter Euler). Starten Sie mit $y_0 = 0$ und beim modifizierten Euler zusätzlich mit $y_1 = 1 - h$. Vergleichen Sie mit der analytischen Lösung, indem Sie den relativen Fehler berechnen. Zeigen Sie, dass beim modifizierten Euler der relative Fehler der numerischen Lösung für große x irgendwann anwächst.

3. Benfordsches Gesetz:

Testen Sie die Benford-Verteilung für große Produkte aus Zufallszahlen. Finden Sie dazu zunächst einen Zufallszahlengenerator, der zufällige gleichverteilte Zahlen im Intervall $]0, 1[$ erzeugt (siehe auch Kapitel 10.1). Multiplizieren Sie dann z.B. 1000 dieser Zufallszahlen, $r = \prod_{i=1}^{1000} x_i$. Erstellen Sie ein Histogramm der führenden Ziffer der Mantisse für 1000 auf diese Art generierte r und vergleichen Sie mit der Benfordschen Verteilung.

3 Differentiation und Integration

Literatur zu diesem Teil:

Es gibt zahlreiche Literatur in der numerischen Mathematik, z.B. Numerical Recipes [1, 2], Stoer [3], aber auch in vielen Computerphysiktexten wie Koonin/Meredith [4] oder das Hjorth-Jensen Skript [5].

3.1 Numerische Differentiation

Numerische Versionen von erster und zweiter Ableitung werden in Form von Differenzenquotienten aus Taylorentwicklungen hergeleitet.

3.1.1 Erste Ableitung

Die 1. Ableitung einer Funktion $f(x)$ wird durch den kontinuierlichen Grenzprozess

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

definiert, der im Computer nicht ausgeführt werden kann, da h sich notwendigerweise nur in diskreten Schritten 0 nähern kann. Naiv wird man daher den **Differenzenquotienten** (siehe Abb. 3.2)

$$f'(x) \approx \frac{f(x + h) - f(x)}{h} + \mathcal{O}(h) \quad (3.1)$$

mit einem "genügend" kleinem h in der Numerik verwenden.

Der Abbruchfehler $R(f')$ ist durch Taylorentwicklung genauer quantifizierbar:

$$\begin{aligned} f(x + h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \dots \\ \frac{f(x + h) - f(x)}{h} &= f'(x) + \frac{1}{2}f''(x)h + \dots \end{aligned}$$

also

$$R(f') = \left| \frac{1}{2}f''(x)h \right| = \mathcal{O}(h)$$

Ein grundsätzliches Problem bei der numerischen Differentiation ist, dass bei großem h der Abbruchfehler groß ist, aber gleichzeitig bei zu kleinem h Auslöschung im Zähler von (3.1) zu großen Rundungsfehlern führt, siehe Abb. 3.1. In der Praxis ist der Bereich zwischen diesen beiden Extremen aber groß genug, so dass numerische Differentiation im Allgemeinen unproblematisch ist.

Etwas besser als die naive Methode (3.1) ist ein **Differenzenquotient mit 2 symmetrischen Punkten** $x \pm h$ (siehe Abb. 3.2):

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h} + \mathcal{O}(h^2) \quad (3.2)$$

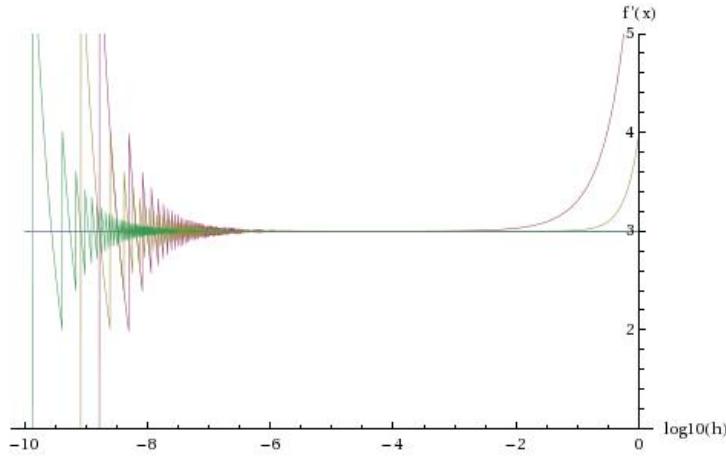


Abbildung 3.1: Typisches Fehlerverhalten bei der numerischen Differentiation. Die Kurven zeigen das Ergebnis der Berechnung von $f'(1) = 3$ der Funktion $f(x) = x^3$ mittels der einfachen Formel (3.1) (rot), der symmetrischen Formel (3.2) (gelb) und der 4-Punkt Formel (3.3) (grün) als Funktion der Diskretisierung h (logarithmisch). Bei der Berechnung wurde der Zähler des Differenzenquotienten jeweils auf 10^{-8} genau gerundet. Rechts bei großen h sieht man den Abbruchfehler, links bei sehr kleinen h den Fehler durch Auslöschung beim Runden. Der Abbruchfehler ist sehr viel kleiner bei der 4-Punkt Formel, Auslösungsprobleme bleiben vergleichbar.

Der Abbruchfehler $R(f')$ ist hier eine Größenordnung kleiner, da sich *alle* geraden Taylorglieder im Zähler wegheben müssen:

$$f(x \pm h) = f(x) \pm f'(x)h + \frac{h^2}{2}f''(x) \pm \frac{h^3}{6}f'''(x) \dots$$

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6}f'''(x) + \dots$$

also

$$R(f') = \left| \frac{1}{6}f'''(x)h^2 \right| = \mathcal{O}(h^2)$$

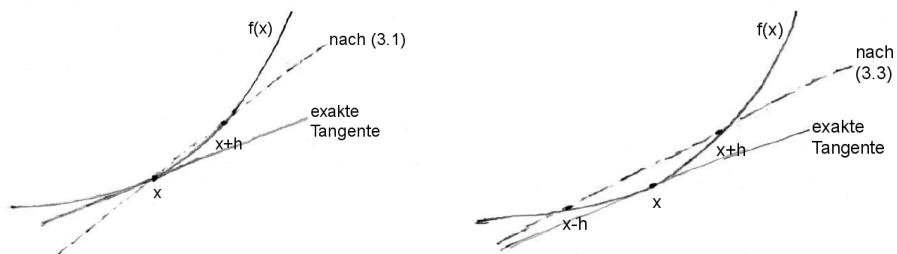


Abbildung 3.2: Links: Näherung der Tangente an den Graphen durch den Differenzenquotienten (3.1). Rechts: Näherung der Tangente an den Graphen durch den symmetrischen Differenzenquotienten (3.2).

Nach dem gleichen Schema kann man noch aufwendigere Formeln mit noch kleinerem Fehler kon-

struieren, z.B. die **symmetrische 4-Punkt Formel**

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} + \mathcal{O}(h^4) \quad (3.3)$$

Taylorentwicklung des Zählers zeigt hier, dass sich auch die Taylorglieder 3. Ordnung und damit alle Glieder bis einschließlich 4. Ordnung in h wegheben, so dass Zähler $= 12h f'(x) + \mathcal{O}(h^5)$ und damit tatsächlich $R(f') = \mathcal{O}(h^4)$.

Die Nachteile der aufwendigeren Formel (3.3) sind:

- $f(x)$ muss an 4 Punkten ausgewertet werden. Dies kann viel Rechenzeit kosten bei numerisch aufwendigem $f(x)$, wie man es in der Physik öfter antrifft, z.B. wenn $f(x)$ eine potentielle Energie und $-f'(x)$ die zugehörige Kraft in einem Vielteilchensystem sind. Dann erfordert jede $f(x)$ -Berechnung eine Summe über alle Wechselwirkungen im Vielteilchensystem.
- Formel (3.3) ist anfälliger für Auslöschung bei kleinem h

In der Praxis benutzt man daher besser die symmetrische 2-Punkt Variante (3.2).

3.1.2 Zweite Ableitung

Die 2. Ableitung erhält man als **doppelten Differenzenquotienten** ebenfalls aus einer Taylorentwicklung:

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \mathcal{O}(h^4)$$

Dies ergibt

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2) \quad (3.4)$$

mit einem Abbruchfehler $R(f'') = \mathcal{O}(h^2)$.

Man kann leicht zeigen, dass sich die numerische 2. Ableitung (3.4) tatsächlich auch aus zweimaliger Anwendung der numerischen 1. Ableitung in der symmetrischen Form (3.2) ergibt:

$$f''(x) \approx \frac{f'(x+h) - f'(x-h)}{2h} \approx \frac{f(x+2h) - f(x) - f(x) + f(x-2h)}{4h^2}$$

was genau (3.4) mit $2h$ statt h entspricht.

In der Praxis benutzen wir die symmetrische 2-Punkt Formel (3.2) für erste und die Formel (3.4) für zweite Ableitungen.

3.2 Numerische Integration

Trapezregel, Mittelpunktsregel, Simpsonregel werden aus einer Zerlegung in Integrationen über kleine Teilintervalle und Taylorentwicklung des Integranden hergeleitet. In der Romberg-Integration wird die Trapezregel mit Interpolation kombiniert, und es wird eine iterierte Trapezregel als praktisches Verfahren vorgestellt. Abschließend werden weitere Verfahren und mehrdimensionale Integration angeschnitten.

Auch die Integration ist über einen Grenzprozess von Riemann-Summen definiert,

$$\int_a^b f(x) dx = \lim_{h \rightarrow 0} \sum_{k=1}^{N(h)} h f(x_k) \quad \text{mit } x_k = a + kh \text{ und } N(h) = \frac{b-a}{h},$$

der im Computer so nicht ausgeführt werden kann.

Die Strategie zur Herleitung numerischer Näherungen wird folgende sein:

- (i) Zerlege das Intervall $[a, b]$ in N Teilintervalle der Länge $h = (b - a)/N$,

$$\int_a^b f(x)dx = \int_a^{a+h} f(x)dx + \int_{a+h}^{a+2h} f(x)dx + \dots + \int_{b-h}^b f(x)dx. \quad (3.5)$$

- (ii) Berechne $\int_{x_k}^{x_k+h} f(x)dx$ durch Taylorentwicklung des Integranden. Dabei verwenden wir für Ableitungen wieder unsere numerischen Formeln aus Kapitel 3.1.

Diese Strategie führt auf mehrere sogenannte **Newton-Cotes-Formeln** basierend auf **äquidistanten** Stützstellen: Je nach Ordnung und Position der Stützstellen der Taylorentwicklung werden wir so zunächst **Trapezregel**, **Mittelpunktsregel** und **Simpsonregel** erhalten. Diese Verfahren können dann durch *Interpolation (Romberg-Methode)* oder *Iteration* verbessert werden, was wir am Beispiel der Trapezregel diskutieren werden.

3.2.1 Trapezregel

Bei der Trapezregel entwickeln wir die Integranden in (3.5) bis zur 1. Ordnung um den *Intervalrand*:

$$\begin{aligned} f(x) &= f(x_k) + f'(x_k)(x - x_k) + \mathcal{O}((x - x_k)^2) \\ &\stackrel{(3.1)}{=} f(x_k) + \frac{f(x_k + h) - f(x_k)}{h}(x - x_k) + \mathcal{O}((x - x_k)^2, (x - x_k)h) \end{aligned}$$

Für das Integral über ein Teilintervall ergibt sich damit

$$\begin{aligned} \int_{x_k}^{x_k+h} f(x)dx &= h f(x_k) + \frac{h}{2} (f(x_k + h) - f(x_k)) + \mathcal{O}(h^3) \\ &= \frac{h}{2} (f(x_k + h) + f(x_k)) + \mathcal{O}(h^3) \end{aligned}$$

Dies motiviert auch den Namen Trapezregel: In jedem Teilintervall benutzen wir eine *lineare* Approximation des Integranden durch eine Gerade durch die beiden Funktionswerte an den Intervallrändern. Daher ist der Wert des Integrals über jedes Teilintervall genau die Fläche eines Trapezes zwischen der x -Achse und der approximierenden Geraden, siehe Abb. 3.3.

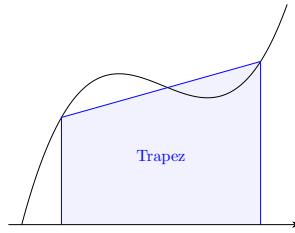


Abbildung 3.3: Lineare Approximation und Trapezregel.

Für das Gesamtintegral erhalten wir dann in der Summe die **Trapezregel**:

$$\begin{aligned} \int_a^b f(x)dx &= h \left[\frac{1}{2}f(a) + f(a + h) + f(a + 2h) + \dots + f(b - h) + \frac{1}{2}f(b) \right] + \mathcal{O}(Nh^3) \\ &= h \sum_{k=1}^{N-1} f(x_k) + \frac{h}{2} (f(x_0) + f(x_N)) + \mathcal{O}(N^{-2}) \end{aligned} \quad (3.6)$$

Die Trapezregel zeichnet sich durch ‘‘Gewichte’’ $1/2$ an den Intervallrändern und 1 für alle inneren Punkte aus. Der Abbruchfehler $\mathcal{O}(Nh^3)$ ist wegen $N = (b - a)/h$ von der Größenordnung $\mathcal{O}(N^{-2})$, wenn er nur als Funktion der Zahl der Stützstellen geschrieben wird.

Wir merken außerdem an, dass die Trapezregel per Konstruktion für *lineare* Funktionen exakt wird, siehe Abb. 3.3.

3.2.2 Mittelpunktsregel

Wenn wir den Integranden in (3.5) wieder bis zur *1. Ordnung*, aber um die *Intervallmitten* herum Taylor entwickeln, ergibt sich die Mittelpunktsregel:

$$\int_{x_k}^{x_k+h} f(x)dx = hf\left(x_k + \frac{h}{2}\right) + \mathcal{O}(h^3)$$

Alle ungeraden Terme $(x - x_k - h/2)^{(2n+1)}$ im Integranden – und damit insbesondere auch der Term erster Ordnung – ergeben bei $\int_{x_k}^{x_k+h} \dots = 0$, was die Größenordnung des Fehlers erklärt. Für das Gesamtintegral erhalten wir dann in der Summe die **Mittelpunktsregel**:

$$\int_a^b f(x)dx = h \left[f\left(a + \frac{h}{2}\right) + f\left(a + \frac{3}{2}h\right) + \dots + f\left(b - \frac{3}{2}h\right) + f\left(b - \frac{h}{2}\right) \right] + \mathcal{O}(Nh^3)$$

(3.7)

Die Fehlergrößenordnung ist identisch zur Trapezregel. Im Unterschied zur Trapezformel werden hier aber nur ‘‘innere’’ Punkte benötigt (die alle mit gleichem ‘‘Gewicht’’ 1 eingehen), d.h. die Funktion muss nicht direkt an den Intervallrändern ausgewertet werden. Das kann vorteilhaft sein, wenn der Integrand in irgendeiner Form bei $x = a$ oder $x = b$ ‘‘problematisch’’ ist. Die Trapezregel (3.6) ist damit das einfachste Beispiel für eine **abgeschlossene Newton-Cotes Formel**, während die Mittelpunktsregel (3.7) das einfachste Beispiel für eine **offene Newton-Cotes Formel** darstellt.

3.2.3 Simpsonregel



Abbildung 3.4: Links: Thomas Simpson (1710-1761), englischer Mathematiker. Mitte: Leonhard Euler (1707-1783), Mathematiker und Physiker. Rechts: Colin MacLaurin (1698-1746), schottischer Mathematiker. (Quelle: Wikipedia).

Im Gegensatz zu Trapez- und Mittelpunktsregel wird bei der Simpsonregel der Integrand bis zur *2. Ordnung* entwickelt, also durch eine *quadratische* Funktion approximiert, was einen kleineren Fehler zur Folge haben wird. Dabei wird um die *Intervallmitte* entwickelt. In leichter Abwandlung

von (3.5) starten wir mit $N/2$ (also N gerade) Intervallen der Länge $2h$:

$$\int_a^b f(x)dx = \int_a^{a+2h} f(x)dx + \dots + \int_{b-2h}^b f(x)dx.$$

Taylorentwicklung um die Intervallmitten x_k bis zur zweiten Ordnung ergibt:

$$\begin{aligned} f(x) &= f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2 + \mathcal{O}((x - x_k)^3) \\ &\stackrel{(3.2),(3.4)}{=} f(x_k) + \frac{f(x_k + h) - f(x_k - h)}{2h}(x - x_k) + \\ &\quad + \frac{1}{2} \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}(x - x_k)^2 + \mathcal{O}\left((x - x_k)^3, (x - x_k)h^2, \underline{(x - x_k)^2h^2}\right) \end{aligned}$$

Bei Integration $\int_{x_k-h}^{x_k+h} \dots$ über ein Teilintervall ergeben alle ungeraden Terme $(x - x_k)^{(2n+1)}$ im Integranden Null, insbesondere auch die Terme $\sim (x - x_k)^3, (x - x_k)h^2$. Daher bestimmt der unterstrichene Term letztlich den Fehler. Für das Integral über ein Teilintervall erhalten wir dann

$$\begin{aligned} \int_{x_k-h}^{x_k+h} f(x)dx &= 2hf(x_k) + \frac{h}{3} (f(x_k + h) - 2f(x_k) + f(x_k - h)) + \mathcal{O}(h^5) \\ &= h \left(\frac{1}{3}f(x_k + h) + \frac{4}{3}f(x_k) + \frac{1}{3}f(x_k - h) \right) + \mathcal{O}(h^5) \end{aligned}$$

Für das Gesamtintegral erhalten wir dann in der Summe die **Simpsonregel** (manchmal auch **Kepfersche Fassregel** genannt):

$$\int_a^b f(x)dx = h \left[\frac{1}{3}f(a) + \frac{4}{3}f(a + h) + \frac{2}{3}f(a + 2h) + \frac{4}{3}f(a + 3h) + \dots + \frac{4}{3}f(b - h) + \frac{1}{3}f(b) \right] + \mathcal{O}(N^{-4}) \quad (3.8)$$

Die Simpsonregel zeichnet sich durch ‘‘Gewichte’’ $1/3$ an den Intervallrändern und den charakteristischen Wechsel $4/3, 2/3$ für alle inneren Punkte aus. Da N gerade war, macht man sich leicht klar, dass der letzte innere Punkt auch wieder Gewicht $4/3$ haben muss. Der Abbruchfehler ergibt sich aus $\mathcal{O}(Nh^5) = \mathcal{O}(N^{-4})$ wegen $N = (b - a)/h$. Eine Taylorentwicklung um einen anderen Punkt als die Intervallmitte hätte andere Koeffizienten geliefert und einen größeren Fehler $\mathcal{O}(N^{-3})$ als in (3.8) ergeben.

Wir merken an, dass die Simpsonregel exakt wird für *quadratische* Funktionen.

3.2.4 Euler-McLaurin Fehlerabschätzung

Wir können die Fehlerabschätzung der Integrationsformeln systematischer vornehmen mit Hilfe der **Euler-McLaurin-Formel**, die die Approximation einer Summe durch ein Integral beschreibt:¹

$$\begin{aligned} \sum_{k=1}^{N-1} g(k) &= \int_0^N g(k) dk - \frac{1}{2} [g(0) + g(N)] + \frac{1}{12} [g'(N) - g'(0)] \\ &\quad - \frac{1}{720} [g'''(N) - g'''(0)] + \dots \\ &= \int_0^N g(k) dk - \frac{1}{2} [g(0) + g(N)] + \sum_{n \geq 1} \frac{B_{2n}}{(2n)!} [g^{(2n-1)}(N) - g^{(2n-1)}(0)] \end{aligned} \quad (3.10)$$

wobei die **Bernoullizahlen** sind, die definiert sind über die Potenzreihe der Funktion

$$\begin{aligned} \tau(t) &\equiv \frac{t}{1-e^{-t}} = \sum_{n=0}^{\infty} B_n \frac{t^n}{n!} \\ B_2 &= \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \dots \end{aligned} \quad (3.11)$$

Aus der Euler-MacLaurin Formel (3.10) folgt mit $g(k) = f(x_k)$ und $dx = hdk$

$$\int_{x_0}^{x_N} f(x) dx = h \sum_{k=1}^{N-1} f(x_k) + \frac{h}{2} [f(x_0) + f(x_N)] - \frac{h^2}{12} [f'(x_N) - f'(x_0)] + \dots \quad (3.12)$$

was wieder die Trapezregel (3.6) ist, wobei der letzte Summand nun den Fehler $\mathcal{O}(h^2) = \mathcal{O}(N^{-2})$ genauer spezifiziert.

Folgende Aussagen erlauben es, den Fehler bei der Trapezregel genauer zu beschreiben:

- Für Euler-MacLaurin (3.10) und für (3.12) gilt: Der Fehler beträgt betragsmäßig höchstens das Doppelte des ersten vernachlässigten Terms

¹ Beweis der Euler-MacLaurin Formel (3.10)
Dazu starten wir mit

$$\sum_{x=1}^{N-1} e^{tx} = e^t \frac{1 - e^{Ntx}}{1 - e^t} \stackrel{(*)}{=} \tau(t) \frac{1}{t} (e^{Ntx} - 1) = \tau(t) \int_0^N e^{tx} dx \quad (3.9)$$

wobei bei (*) die Definition (3.11) der Bernoulli-Zahlen B_n ins Spiel kommt.
Trick: Nun ersetzen wir x durch Operator d/dh , für den

$$e^{x \frac{d}{dh}} g(h) = g(x+h)$$

gilt, wie man durch Taylorentwicklung zeigt (auch bekannt aus der QM: "Impulsoperator" d/dh ist Erzeugender von Translationen), und lässt beide Seiten von (3.9) auf die Funktion $g(h)$ wirken:

$$\begin{aligned} \sum_{x=1}^{N-1} g(x+h) &\stackrel{(3.9)}{=} \tau\left(\frac{d}{dh}\right) \int_0^N g(x+h) dx \\ &= \tau\left(\frac{d}{dh}\right) \int_h^{N+h} g(x) dx \end{aligned}$$

Wird in der rechten Seite die Reihenentwicklung (3.11) eingesetzt und die Formel bei $h = 0$ ausgewertet, führt dies auf die Euler-MacLaurin Formel:

$$\sum_{k=1}^{N-1} g(k) = B_0 \int_0^N g(x) dx + B_1(g(N) - g(0)) + \frac{1}{2} B_2(g'(N) - g'(0)) + \dots$$

mit $B_0 = 1$, $B_1 = 1/2$, $B_{2n+1} = 0$ ($n \geq 1$).

- Alle Fehlerterme in (3.12) sind gerade (wegen $B_{2n+1} = 0$ für $n \geq 1$).

Diese Aussagen über die Fehler bei der Trapezregel führen auf die Romberg-Integration.

3.2.5 Romberg-Integration

Aus den letzten Aussagen über den Fehler bei der Trapezregel folgt: Wenn T_N das Trapezregel-Ergebnis für N Intervalle mit einem Fehler Δ_N ist, dann gilt $\Delta_N \sim N^{-2}$ in führender Ordnung und daher bei Intervallhalbierung

$$\Delta_N = \frac{1}{4} \Delta_{N/2} + \mathcal{O}(N^{-4})$$

Dann hat aber die Kombination

$$S_N = \frac{4}{3} T_N - \frac{1}{3} T_{N/2} \quad (3.13)$$

einen Fehler der Ordnung $\mathcal{O}(N^{-4})$, da sich die Fehler $\mathcal{O}(N^{-2})$ genau wegheben. S_N ist aber gerade die Simpsonregel (3.8) für N Intervalle, die ja auch einen kleineren Fehler hat.

Die Konstruktion (3.13) ist das einfachste Beispiel der **Romberg-Methode**:

Die Euler-MacLaurin Formel (3.12) stellt das Trapezregel-Ergebnis $T(h)$ als Polynom in h^2 dar, da nur gerade Potenzen bei den Fehlertermen vorkommen:

$$T(h) = \underbrace{\int_a^b f(x)dx}_{\equiv I} + \delta_1 h^2 + \delta_2 h^4 + \dots \quad (3.14)$$

mit dem gesuchten Integral I als Wert bei $h = 0$ und festen δ_m (durch $f^{(2m-1)}(a)$ und $f^{(2m-1)}(b)$ bestimmt nach (3.12)).

Die Idee der **Romberg-Integration** ist folgende:

- Berechne $T(h)$ nach Trapezregel für $n + 1$ verschiedene h .
- Finde interpolierendes Polynom n -ten Grades in h^2 , siehe (3.14).
- Der Wert des Polynoms bei $h = 0$ approximiert das gesuchte Integral I .

Wir zeigen nun, dass die Formel (3.13) tatsächlich der einfachsten Romberg-Methode mit einer *linearen* Interpolation ($n = 1$) entspricht. Dazu betrachten wir (3.14) bis zur Ordnung h^2 und machen folgenden Ansatz:

$$T(h) = I + \delta_1 h^2 \quad (3.15)$$

und arbeiten obige Schritte (i)-(iii) ab:

- Es gilt $T(h) = T_N$ und $T(2h) = T_{N/2}$ nach Definition von T_N und $T_{N/2}$.
- Also gilt

$$\begin{aligned} T(h) &= I + \delta_1 h^2 = T_N \\ T(2h) &= I + 4\delta_1 h^2 = T_{N/2} \end{aligned}$$

woraus $3\delta_1 h^2 = T_{N/2} - T_N$ folgt. Damit lässt sich δ_1 und damit das interpolierende Polynom (3.15) bestimmen.

- Der Wert I des Polynoms bei $h = 0$ ist

$$I = T_N - \delta_1 h^2 = \frac{4}{3} T_N - \frac{1}{3} T_{N/2}$$

wie in (3.13)

Das heißt, die Simpsonregel kann als niedrigste Ordnung der Romberg-Interpolation interpretiert werden. Allgemein gilt, dass die Romberg-Integration mit $n > 1$ eine sehr effektive Methode für glatte Integranden darstellt.

3.2.6 Iterierte Trapezregel

Eine für die Praxis wichtige Eigenschaft der Trapezregel ist ihre **Iterierbarkeit** bei Intervallhalbierung:

- **1. Iteration:** Diskretisierung mit gesamter Intervalllänge $h_1 = b - a$:

$$\int_a^b f(x)dx \approx (b - a) \left[\frac{1}{2}f(a) + \frac{1}{2}f(b) \right] \equiv T_1$$

- **2. Iteration:** Eine zusätzliche Stützstelle in der Intervallmitte halbiert das Diskretisierungsintervall auf $h_2 = (b - a)/2$:

$$\begin{aligned} \int_a^b f(x)dx &\approx \frac{b - a}{2} \left[\frac{1}{2}f(a) + f\left(\frac{a+b}{2}\right) + \frac{1}{2}f(b) \right] \equiv T_2 \quad \text{oder} \\ T_2 &= \frac{1}{2} \left(T_1 + \underbrace{(b - a)f\left(\frac{a+b}{2}\right)}_{\text{neu berechnen}} \right) \end{aligned}$$

wobei T_1 schon in 1. Iteration berechnet wurde und nur die Beiträge von der neuen Stützstelle neu berechnet werden müssen.

- **(n+1)-te Iteration:** Wir führen 2^{n-1} zusätzliche Stützstellen in der Mitte jedes Diskretisierungsintervalls ein, so dass $h_{n+1} = h_n/2 = (b - a)/2^n$ und

$$T_{n+1} = \frac{1}{2} \left(T_n + h_n \underbrace{\sum_{k=0}^{2^{n-1}-1} f(a + h_{n+1} + kh_n)}_{\text{neu berechnen}} \right)$$

wobei T_n schon in n -ter Iteration berechnet wurde und nur die Beiträge von den neuen Stützstellen neu berechnet werden.

Der Vorteil der Iterierbarkeit liegt offensichtlich darin, dass bereits berechnete Beiträge immer weiter verwendet werden können und in jeder Iteration die Funktion $f(x)$ nur an den neuen Stützstellen neu ausgewertet werden muss. Die Iterierbarkeit erlaubt auch eine effektive Fehler- bzw. Konvergenzkontrolle, indem so lange iteriert wird, bis aufeinanderfolgende Iterationen sich nur noch um ein vorgegebenes Genauigkeitsziel unterscheiden. Ein solches Verfahren inklusive Fehlerkontrolle ist z.B. in den *Numerical Recipes* [2] in der Funktion `qtrap` mit Hilfe der Struktur `Trapzd` realisiert.

Es ist auch möglich, die Iteration noch mit einer **Romberg-Interpolation** zu verbinden: Die Werte T_1, \dots, T_n mit Diskretisierungslängen h_1, \dots, h_n können nach $h = 0$ interpoliert werden mit einem Polynom $(n - 1)$ -ten Grades in h^2 . Dies ist beispielsweise in der Funktion `qromb` in den *Numerical Recipes* [2] realisiert (dazu benötigt man dann natürlich auch eine Polynom-Interpolationsroutine, die hier nicht besprochen werden wird).

3.2.7 Weitere Verfahren, mehrdimensionale Integrale

Neben den bisher besprochenen Verfahren, die alle *äquidistante* Stützstellen benutzen, gibt es noch eine große Klasse von Verfahren, die auf **nicht-äquidistanten Stützstellen** beruhen, nämlich

die **Gauß-Quadratur**. Dort wird versucht, die Stützstellen gerade dem Integranden angepasst zu wählen, um Fehler klein zu halten.

In der Physik sind natürlich **mehrdimensionale Integrale** von besonderem Interesse. Hier gibt es zwei Verfahrensweisen:

- (i) Wir reduzieren ein n -dimensionalen Integral auf n 1-dimensionale Integrationen, die wir mit den bereits diskutierten Methoden ausführen, also z.B.

$$\int_{[a,b]^n} f(x_1, \dots, x_n) d^n \vec{x} = \int_a^b dx_1 \dots \int_a^b dx_n f(x_1, \dots, x_n).$$

(Äquivalent kann man auch eine Zerlegung des Integrationsgebietes in kleine Würfel vornehmen.) Dies wird natürlich zunehmend schwierig für komplizierter geformte Integrationsgebiete. Außerdem akkumulieren sich Fehler, besonders bei hochdimensionalen Integralen.

- (ii) Daher verwendet man bei hochdimensionalen Integralen oft die **Monte-Carlo Integration**, die wir in Kapitel 11 noch als stochastische Methode kennenlernen werden. Bei der Monte-Carlo Integration verwendet man *zufällig* im Integrationsgebiet verteilte Stützstellen. Damit kann man in der Praxis recht einfach auch über komplizierter geformte Gebiete integrieren. Außerdem stellt sich heraus, dass der Fehler bei hochdimensionalen Integralen bei stochastischer Monte-Carlo Integration kleiner sein wird als bei den bisher besprochenen deterministischen Methoden.

Bei hochdimensionalen Integralen sieht man dann auch, wie das Integrationsproblem zu einem zentralen Problem der statistischen Physik wird, wo beispielsweise die Zustandssumme eines klassischen N -Teilchen Systems mit Hamiltonfunktion $\mathcal{H}(\{\vec{r}_i\}, \{\vec{p}_i\})$ als $6N$ -dimensionales Integral über Orte und Impulse geschrieben wird:

$$Z = \int d^3 \vec{r}_1 \dots \int d^3 \vec{r}_N \int d^3 \vec{p}_1 \dots \int d^3 \vec{p}_N e^{-\mathcal{H}(\{\vec{r}_i\}, \{\vec{p}_i\})/k_B T}.$$

Hier wird aus der Monte-Carlo Integration dann die **Monte-Carlo Simulation**, in der man typischerweise nicht die Zustandssumme Z selbst, sondern Mittelwerte von Observablen berechnet.

3.3 Uneigentliche Integrale

Es wird diskutiert, wie “problematische” Integrale (unendliches Integrationsintervall, singuläre Integranden, Hauptwertintegrale) numerisch integriert werden können. Eine physikalische Anwendung sind Kramers-Kronig Relationen.

Auch bei der numerischen Integration sollte man uneigentliche Integrale nochmal gesondert diskutieren. Probleme bereiten dabei **unendliche Integrationsintervalle**, also $a = -\infty$ und/oder $b = \infty$, **singuläre Integranden** oder **Hauptwertintegrale**. Hauptwertintegrale sind gerade in der Physik von gewisser Wichtigkeit, da sie in der linearen Antworttheorie in den **Kramers-Kronig Relationen** auftreten.

3.3.1 Unendliches Integrationsintervall

- a) Man kann versuchen, die Integration durch Substitution auf ein *endliches* Integrationsintervall abzubilden, z.B. mit $u = 1/x$ wird

$$\int_a^\infty dx f(x) = \int_0^{1/a} du \frac{1}{u^2} f\left(\frac{1}{u}\right). \quad (3.16)$$

Dann muss man eventuell den Preis zahlen, dass eine singuläre Jacobideterminante (wie $1/u^2$ in (3.16)) auftritt und man damit einen singulären Integranden erhält.

b) Einfacher ist es oft, die Integration bei einem großen, aber endlichen x_{\max} abzubrechen,

$$\int_a^{\infty} f(x)dx = \int_a^{x_{\max}} f(x)dx + \int_{x_{\max}}^{\infty} f(x)dx \quad (3.17)$$

und das verbleibende Integral von x_{\max} bis ∞ als ‘‘Fehler’’ abzuschätzen oder bei bekannter Asymptotik des Integranden sogar analytisch zu lösen.

3.3.2 Singuläre Integranden

Das Integral kann zunächst immer so aufgeteilt werden, dass die Singularität am Rand eines Integrationsintervalls auftritt. Wir unterscheiden 3 Fälle:

Fall 1: Der Integrand ist problematisch, ohne singulär zu sein.

a) Ein Beispiel ist $\int_0^1 dx(\sin x)/x$, wo $f(0)$ nicht ‘‘einfach’’ berechenbar ist. Hier kann man einfach die Mittelpunktsregel (3.7) benutzen, die ja nur ‘‘innere’’ Stützstellen verwendet.

b) Ein anderes Beispiel ist $\int_0^b dx\sqrt{x}$, wo $f'(0) = \infty$. Damit divergiert der Fehler der Trapezregel nach der Euler-MacLaurin Formel, siehe (3.12). Hier kann man den problematischen Teil abspalten (mit kleinem h):

$$\int_0^b f(x)dx = \int_0^h f(x)dx + \int_h^b f(x)dx.$$

Der zweite Teil kann ganz normal nach der Trapezregel berechnet werden. Der erste Teil kann analytisch behandelt werden bei bekannter Asymptotik $f(x) \approx cx^\alpha$ ($0 < \alpha < 1$) für $x \leq h \ll 1$:

$$\int_0^h dx cx^\alpha = \frac{c}{\alpha+1} h^{\alpha+1} = \frac{h}{\alpha+1} f(h)$$

Er kann dann mit dem zweiten Teil zu einer ‘‘modifizierten Trapezregel’’ kombiniert werden:

$$\int_0^b f(x)dx = h \left[\left(\frac{1}{\alpha+1} + \frac{1}{2} \right) f(h) + f(2h) + \dots + \frac{1}{2} f(b) \right]$$

Fall 2: Der Integrand ist singulär, aber integrierbar.

Wir betrachten Integrale vom Typ $\int_0^b f(x)dx$ mit $f(x) \approx cx^{-\alpha}$ mit $0 < \alpha < 1$.

a) Man kann versuchen den singulären Faktor abzuspalten,

$$f(x) = \frac{g(x)}{x^\alpha} = \frac{g(x) - g(0)}{x^\alpha} + \frac{g(0)}{x^\alpha},$$

so dass der erste Summand nicht mehr singulär ist und numerisch integriert werden kann, während der zweite Summand als einfache Potenz analytisch gelöst werden kann.

b) Eine andere Möglichkeit besteht in einer Substitution, $u = x^{1-\alpha}$, so dass $du = (1-\alpha)x^{-\alpha}dx$:

$$\int_0^b f(x)dx = \int_0^b \frac{g(x)}{x^\alpha} dx = \frac{1}{1-\alpha} \int_0^{b^{1/(1-\alpha)}} du g(u^{1/(1-\alpha)})$$

Die Methoden für Fall 2 sind im Übrigen auch auf Fall 1 anwendbar.

Fall 3: Der Integrand ist nicht integrierbar.

Dann sollte man auch definitiv nicht versuchen, dass Integral numerisch zu berechnen! Die einzige Ausnahme sind Hauptwertintegrale.

3.3.3 Hauptwertintegrale

Kann man einem offensichtlich divergentem Integral

$$\int_a^b dx \frac{f(x)}{x-z} = ???$$

einen sinnvoll definierten Wert zuweisen?

Dies ist durch folgenden Grenzprozess tatsächlich möglich:

$$\boxed{\mathcal{P} \int_a^b dx \frac{f(x)}{x-z} \equiv \lim_{\epsilon \rightarrow 0} \left[\int_a^{z-\epsilon} dx \frac{f(x)}{x-z} + \int_{z+\epsilon}^b dx \frac{f(x)}{x-z} \right]} \quad (3.18)$$

Dieser Grenzprozess definiert das **Hauptwertintegral**, das mit dem Symbol $\mathcal{P} \int \dots$ (\mathcal{P} für “principal value”) bezeichnet wird.

Wir betrachten ein **Beispiel**:

$$\mathcal{P} \int_{-1}^1 dx \frac{1}{x} = \lim_{\epsilon \rightarrow 0} \left[\int_{-1}^{-\epsilon} \frac{dx}{x} + \int_{\epsilon}^1 \frac{dx}{x} \right] = \lim_{\epsilon \rightarrow 0} \left[\ln \epsilon + \ln \frac{1}{\epsilon} \right] = 0 \quad (3.19)$$

Numerisch berechnet man Hauptwertintegrale, indem man die Umgebung um die Singularität bei $x = z$ zunächst isoliert:

$$\mathcal{P} \int_a^b dx \frac{f(x)}{x-z} = \underbrace{\int_a^{z-\Delta} dx \frac{f(x)}{x-z}}_{\text{numerisch}} + \underbrace{\int_{z+\Delta}^b dx \frac{f(x)}{x-z}}_{\text{numerisch}} + \underbrace{\mathcal{P} \int_{z-\Delta}^{z+\Delta} dx \frac{f(x)}{x-z}}_{\equiv I_\Delta(z)}$$

Um $I_\Delta(z)$ zu berechnen, substituieren wir $s = (x - z)/\Delta$,

$$I_\Delta(z) = \mathcal{P} \int_{-1}^1 ds \frac{f(s\Delta + z)}{s},$$

und ziehen dann $\mathcal{P} \int_{-1}^1 ds f(z)/s = 0$ ab (siehe (3.19)), so dass

$$I_\Delta(z) = \int_{-1}^1 ds \frac{f(s\Delta + z) - f(z)}{s},$$

was dann kein Hauptwertintegral mehr darstellt, weil der Integrand nicht mehr singulär ist, und damit auch numerisch berechnet werden kann.

3.3.4 Kramers-Kronig Relationen

Hauptwertintegrale spielen in der Physik eine wichtige Rolle, und zwar in den Kramers-Kronig Relationen (Kramers 1927, Kronig 1926) in der **linearen Antworttheorie**. In der Regel antworten physikalische Systeme *linear* auf kleine zeitabhängige äußere Kräfte oder *Anregungen* $F(t)$. Der allgemeinste kausale, lineare Zusammenhang zwischen einer Messgröße oder *Antwort* $A(t)$ auf eine Anregung $F(t)$ ist

$$\boxed{A(t) = \int_{-\infty}^{\infty} R(t-t')F(t')dt',} \quad (3.20)$$

wobei $R(t)$ eine **Response-Funktion** ist. Nach Fouriertrafo

$$\tilde{A}(\omega) = \int_{-\infty}^{\infty} dt A(t) e^{i\omega t}, \quad A(t) = \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \tilde{A}(\omega) e^{-i\omega t}$$



Abbildung 3.5: Links: Hendrik Anthony Kramers (1894-1952), niederländischer Physiker (ca. 1928, Mitte mit Uhlenbeck (rechts) und Goudsmit (links)). Rechts: Ralph Kronig (1904-1995), deutsch-amerikanischer Physiker. (Quelle: Wikipedia).

ergibt sich daraus nach dem Faltungssatz

$$\tilde{A}(\omega) = \tilde{R}(\omega)\tilde{F}(\omega) \quad (3.21)$$

Solche linearen Antworten finden sich überall in der Physik, solange die Störung $F(t)$ klein ist. Wir geben einige **Beispiele**:

- 1) In der E-Dynamik gibt es zahlreiche lineare Antworten z.B. die Polarisierung P eines Dielektrikums als Antwort auf ein angelegtes elektrisches Feld E , die durch die Suszeptibilität χ als Antwort-Funktion gegeben ist: $\tilde{P}(\omega) = \epsilon_0\tilde{\chi}(\omega)\tilde{E}(\omega)$. Daraus ergibt sich dann auch ein linearer Zusammenhang $\tilde{D}(\omega) = \tilde{\epsilon}(\omega)\tilde{E}(\omega)$ zwischen dielektrischer Verschiebung D und elektrischem Feld E .
- 2) Beim getriebenen gedämpften Oszillator gibt es einen linearen Zusammenhang zwischen Auslenkung und äußerer Kraft,

$$\tilde{x}(\omega) = \tilde{R}(\omega)\tilde{F}(\omega).$$

Die Antwortfunktion R ergibt sich in diesem Fall direkt aus der Bewegungsgleichung durch Fouriertransformation:

$$\begin{aligned} \ddot{x} &= -\gamma\dot{x} - \omega_0^2x + \frac{F(t)}{m} \\ \tilde{x}(\omega) &= \underbrace{\frac{1}{m} \frac{1}{\omega_0^2 - \omega^2 - i\omega\gamma}}_{= \tilde{R}(\omega)} \tilde{F}(\omega). \end{aligned}$$

- 3) In visko-elastischen Materialien gibt es einen linearen Zusammenhang zwischen Spannung σ und Verzerrung ϵ , der durch ein Elastizitätsmodul oder eine Viskosität als Antwortfunktion gegeben ist, $\tilde{\sigma}(\omega) = \tilde{G}(\omega)\tilde{\epsilon}(\omega)$.

Für die Antwortfunktion $\tilde{R}(\omega)$ gilt: $\tilde{R}(\omega)$ hat i.Allg. einen Real *und* Imaginärteil.

Diese werden oft auch als $\tilde{R}(\omega) = \tilde{R}'(\omega) + i\tilde{R}''(\omega)$ geschrieben. Der **Imaginärteil** $\text{Im } \tilde{R}(\omega)$ ist der um $\pi/2$ phasenverschobene **dissipative Response** des Systems, der groß wird, wenn Resonanzen oder Absorption bei der Frequenz ω vorliegen. Der **Realteil** $\text{Im } \tilde{R}(\omega)$ ist der **nicht-dissipative** (auch dispersive) **Response** und enthält u.a. auch den statischen Response auf zeitunabhängige Störungen bei $\omega = 0$.

Dies kann man sich gut am obigen Beispiel 2) des *Oszillators* veranschaulichen:

- Für den statischen Response bei $\omega = 0$ gilt: $\operatorname{Re} \tilde{R}(0) = \frac{1}{m\omega_0^2} \sim \frac{1}{\text{Federkonstante}}$ und $\operatorname{Im} \tilde{R}(0) = 0$. Er hat keinen Imaginärteil.
- Resonanzen und Energieabsorption finden an den Resonanzfrequenzen (Eigenfrequenzen) ω_0 des Oszillators statt. Dort gilt $\operatorname{Re} \tilde{R}(\omega_0) = 0$, während der dissipative Anteil $\operatorname{Im} \tilde{R}(\omega)$ bei ω_0 maximal wird.
- Der dissipative Anteil $\operatorname{Im} \tilde{R}(\omega) \sim \gamma$ ist immer proportional zur Reibungskonstanten und verschwindet für $\gamma = 0$.

Auch aus dem Beispiel 1) des Dielektrikums in der E-Dynamik wissen wir, dass $\operatorname{Im} \chi$ Dämpfung und Absorption elektrischer Felder im Medium beschreibt (Eindringtiefe), während $\operatorname{Im} \chi$ die nicht-dissipative Polarisierbarkeit angibt. Es gibt auch einen engen Zusammenhang zum Oszillatorteispiel, da die einfachste Modellvorstellung eines Dielektrikums auf gebundenen Ladungen (Elektronen) beruht, die als vom Feld E angeregte Oszillatoren angesehen werden. Dies wird später noch diskutiert werden.

Die Kramers-Kronig Relationen stellen einen Zusammenhang zwischen dem Real- und Imaginärteil von $\tilde{R}(\omega)$ her. Dieser Zusammenhang ist eine Integralbeziehung, in der Hauptwertintegrale vorkommen. Die Kramers-Kronig Relationen beruhen auf drei sehr allgemeinen Eigenschaften der Response-Funktion $R(t)$, die fast immer gelten:

- $R(t)$ ist *reell*, woraus

$$\tilde{R}^*(\omega) = \tilde{R}(-\omega) \quad (3.22)$$

im Fourieraum folgt.

- Es besteht ein *kausaler* Zusammenhang zwischen Störung F und Antwort A ; daher sollte $R(t) = 0$ sein für $t > 0$. Für die Fouriertransformierte $\tilde{R}(\omega) = \int_0^\infty dt R(t) e^{i\omega t}$ folgt dann

$$\tilde{R}(\omega) \text{ ist analytisch in oberer Halbebene,} \quad (3.23)$$

weil dort $\operatorname{Re}(i\omega t) < 0$ und damit die e-Funktion in $\int_0^\infty dt R(t) e^{i\omega t}$ in der ganzen oberen ω -Halbebene Konvergenz bewirkt. Dazu muss zusätzlich noch gelten, dass $R(t)$ *beschränkt* ist oder aber zumindest schwächer als exponentiell wächst für große t (z.B. potenzartig $\sim t^n$). Dies ist in der Regel erfüllt.

- Es gilt

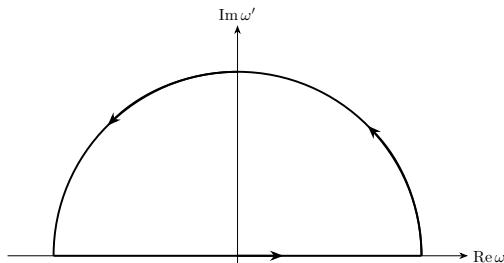
$$\lim_{|\omega| \rightarrow \infty} |\tilde{R}(\omega)| = 0, \quad (3.24)$$

d.h. der Response ist nicht beliebig "schnell" (kurze Zeitskalen entsprechen $\omega \rightarrow \infty$) bzw. $R(t)$ sollte für $t \rightarrow 0$ kleiner bleiben als $\sim t^{-1}$. *Beschränktheit* von $R(t)$ würde auch hier ausreichen.

Aus (3.23) folgt nach dem Cauchy-Integralsatz (oder Residuensatz) für ω mit $\operatorname{Im} \omega > 0$

$$\tilde{R}(\omega) = \oint_C \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega}, \quad (3.25)$$

wo C eine geschlossene Kontur um ω ist, die nur Punkte in der oberen Halbebene umschließt. Wegen der Analytizität von $\tilde{R}(\omega)$ kann C frei gewählt werden, z.B. bestehend aus der reellen Achse und einem Halbkreis $\omega' = r' e^{i\phi'}$ mit $\phi' \in [0, \pi]$, der unendlich groß gemacht wird, also $r' \rightarrow \infty$:



Wenn nun zusätzlich (3.24) gilt, fällt das Integral über den unendlich großen Halbkreis heraus,

$$\left| \int_{\text{Halbkreis}} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega} \right| = \left| \int_0^\pi \frac{d\phi'}{2\pi} r' e^{i\phi'} \frac{\tilde{R}(r' e^{i\phi'})}{r' e^{i\phi'} - \omega} \right| \xrightarrow{r' \rightarrow \infty} 0,$$

und nur das Integral entlang der reellen Achse trägt in (3.25) bei:

$$\tilde{R}(\omega) = \oint_C \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega} = \int_{-\infty}^{\infty} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega} \quad \text{für } \operatorname{Im} \omega > 0. \quad (3.26)$$

Für beliebiges $\omega \in \mathbb{R}$ ist $\operatorname{Im}(\omega + i\varepsilon) = \varepsilon > 0$ und es folgt

$$\boxed{\tilde{R}(\omega + i\varepsilon) = \int_{-\infty}^{\infty} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega - i\varepsilon} \quad \text{für } \operatorname{Im} \omega > 0} \quad (3.27)$$

Um eine Aussage über reelle Frequenzen ω zu gewinnen, muss der Limes $\varepsilon \rightarrow 0$ durchgeführt werden. Dies erfordert eine kleine Deformation des Integrationsweges, damit die Kontour in der komplexen Ebene auch weiterhin unterhalb von ω verläuft:



Wenn der Radius ε des kleinen Halbkreises in der Kontour gegen 0 geschickt wird, können wir nach Definition des Hauptwertintegrals schreiben

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega - i\varepsilon} &= \underbrace{\int_{-\infty}^{\omega} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega}}_{= \mathcal{P} \int_{-\infty}^{\omega} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega}} + \int_{\text{Halbkreis um } \omega} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')} {\omega' - \omega} \\ &= \mathcal{P} \int_{-\infty}^{\omega} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega} + \frac{1}{2} \tilde{R}(\omega). \end{aligned} \quad (3.28)$$

Das Integral über den kleinen Halbkreis in der komplexen Ebene entspricht einem ‘halben Residuum’:

$$\int_{\text{Halbkreis um } 0} \frac{d\omega'}{2\pi i} \frac{1}{\omega'} = \int_0^\pi \frac{d\phi'}{2\pi} r' e^{i\phi'} \frac{1}{r' e^{i\phi'}} = \frac{1}{2}.$$

Symbolisch kann man (3.28) auch als

$$\boxed{\frac{1}{x - i\varepsilon} = \mathcal{P} \frac{1}{x} + i\pi\delta(x)} \quad (3.29)$$

schreiben, was so zu verstehen ist, dass diese Gleichheit nur unter dem Integral und im Limes $\varepsilon \rightarrow 0$ gilt.

Damit wird aus (3.27) im Limes $\varepsilon \rightarrow 0$

$$\tilde{R}(\omega) = \mathcal{P} \int_{-\infty}^{\infty} \frac{d\omega'}{2\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega} + \frac{1}{2} \tilde{R}(\omega)$$

und schließlich

$$\boxed{\tilde{R}(\omega) = \mathcal{P} \int_{-\infty}^{\infty} \frac{d\omega'}{\pi i} \frac{\tilde{R}(\omega')}{\omega' - \omega}} \quad (3.30)$$

Daraus ergibt sich nun eine Integralbeziehung zwischen dem nicht-dissipativem (dispersivem) $\text{Re } \tilde{R}(\omega)$ und dem dissipativem $\text{Im } \tilde{R}(\omega)$, die **Kramers-Kronig Relationen**

$$\begin{aligned}\text{Re } \tilde{R}(\omega) &= \mathcal{P} \int_{-\infty}^{\infty} \frac{d\omega'}{\pi} \frac{\text{Im } \tilde{R}(\omega')}{\omega' - \omega} \\ \text{Im } \tilde{R}(\omega) &= -\mathcal{P} \int_{-\infty}^{\infty} \frac{d\omega'}{\pi} \frac{\text{Re } \tilde{R}(\omega')}{\omega' - \omega}\end{aligned}\quad (3.31)$$

Diese gelten für *jede* Funktion $\tilde{R}(\omega)$, die die Kausalität b) und die Eigenschaft c) einer nicht beliebig schnellen Antwort erfüllen. Wir können negative Frequenzen in den Kramers-Kronig Relationen (3.31) vermeiden, indem wir noch die Eigenschaft a) einer reellen Response-Funktion benutzen. Dann gilt $\text{Im } \tilde{R}(\omega') = -\text{Im } \tilde{R}(-\omega')$ und $\text{Re } \tilde{R}(\omega') = \text{Re } \tilde{R}(-\omega')$ und damit

$$\begin{aligned}\text{Re } \tilde{R}(\omega) &= \mathcal{P} \int_0^{\infty} \frac{d\omega'}{\pi} \frac{2\omega' \text{Im } \tilde{R}(\omega')}{\omega'^2 - \omega^2} \\ \text{Im } \tilde{R}(\omega) &= -\mathcal{P} \int_0^{\infty} \frac{d\omega'}{\pi} \frac{2\omega \text{Re } \tilde{R}(\omega')}{\omega'^2 - \omega^2}\end{aligned}\quad (3.32)$$

Die Kramers-Kronig Relationen verknüpfen dispersiven Realteil und dissipativen Imaginärteil einer Antwort:

- Oft lässt sich entweder der Realteil oder der Imaginärteil besser messen, dann kann der jeweils andere Teil mit (3.32) bestimmt werden.
- Kann beides gemessen werden, liefert (3.32) eine Konsistenzprüfung.

Diese Berechnungen müssen oft numerisch mit den Formeln (3.32) durchgeführt werden, wobei dann Messdaten die Funktionswerte $\text{Re } \tilde{R}(\omega_k)$ oder $\text{Im } \tilde{R}(\omega_k)$ an Stützstellen ω_k vorgeben.

Wir betrachten das **Beispiel** des Dielektrikums, in dem einfachen Modell, das die gebundenen Ladungen (Elektronen) als vom Feld E angeregte gedämpfte Oszillatoren betrachtet [6]. Ein Elektron ist klassisch durch eine Auslenkung $x(t)$ oder $\tilde{x}(\omega)$ beschrieben. Die Kraft auf ein Elektron im elektrischen Feld $E(t)$ ist $F(t) = -|e|E(t)$, die Polarisation des Dielektrikums ist pro Elektron $p(t) = -|e|x(t)$. Die Bewegungsgleichung im E-Feld ist dann

$$m [\ddot{x} + \gamma \dot{x} + \omega_0^2 x] = -|e|E,$$

wo m die Elektronenmasse, γ die (kleine) Dämpfung und ω_0 die Eigenfrequenz der gebundenen Elektronen ist. Fouriertransformation ergibt dann eine Beziehung zwischen Polarisation und E-Feld und damit die Suszeptibilität χ , die die Response-Funktion darstellt:

$$\begin{aligned}\tilde{p}(\omega) &= -|e|\tilde{x}(\omega) \\ &= \underbrace{\frac{e^2}{m} \frac{1}{\omega_0^2 - \omega^2 - i\omega\gamma}}_{\varepsilon_0 \tilde{\chi}(\omega) \text{ für ein Elektron}} \tilde{E}(\omega).\end{aligned}$$

Für N Moleküle pro Volumen (j = Elektronenindex im Molekül) gilt dann

$$\tilde{\chi}(\omega) = \frac{Ne^2}{\varepsilon_0 m} \sum_j \frac{1}{\omega_{0,j}^2 - \omega^2 - i\omega\gamma_j} \quad (3.33)$$

Wir überprüfen zunächst die beiden wichtigen Eigenschaften b) und c) von Response-Funktionen:

- Das elektrische Feld $E(t)$ ist die Ursache für die Verschiebung $x(t)$ und damit die Polarisation $P(t)$. Die Response-Funktion $\tilde{\chi}(\omega)$ sollte diesen kausalen Zusammenhang beschreiben. Wir sehen, dass die entsprechende Eigenschaft (3.23), tatsächlich erfüllt wird, da die Singularitäten von $\tilde{\chi}(\omega)$ bei $\omega = -i\gamma_j/2 \pm (\omega_{0,j}^2 - \gamma_j^2/4)^{1/2}$, also in der unteren komplexen Halbebene liegen.
- Für $|\omega| \rightarrow \infty$ verschwindet die Suszeptibilität (3.33) auch tatsächlich, letztendlich auf Grund der Trägheit (der \ddot{x} -Term). Damit ist (3.24) erfüllt.

Damit erfüllt $\tilde{\chi}(\omega)$ die Voraussetzungen für die Gültigkeit der Kramers-Kronig Relationen und es gibt einen Zusammenhang zwischen $\operatorname{Re} \tilde{\chi}(\omega)$ (Dispersion) und $\operatorname{Im} \tilde{\chi}(\omega)$ (Dissipation). In Abb. 3.6 sind als Beispiel $\operatorname{Re} \tilde{\chi}(\omega)$ und $\operatorname{Im} \tilde{\chi}(\omega)$ für eine Eigenfrequenz ($j = 1$) bei $\omega_{0,1} = 1$ gezeigt, die über Kramers-Kronig Relationen zusammenhängen.

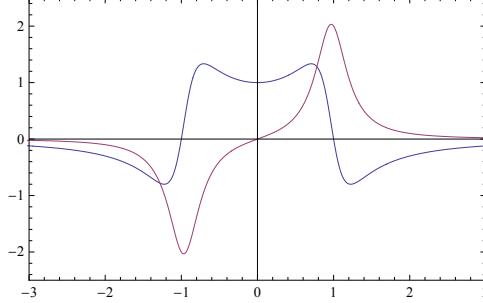


Abbildung 3.6: Beispiel für Real- (blau) und Imaginärteil (rot) von $\tilde{\chi}(\omega)$, die über Kramers-Kronig Relationen zusammenhängen. Das Beispiel ist $\tilde{\chi}(\omega)$ aus (3.33) (in Einheiten von $Ne^2/\epsilon_0 m$) für $j = 1$ mit $\omega_{0,1} = 1$ und $\gamma_1 = 0.5$ (in Einheiten einer beliebigen inversen Zeit). Absorptionsmaxima im Imaginärteil sind typischerweise mit einem Nulldurchgang im Realteil verknüpft.

In der Form (3.31) gelten die Kramers-Kronig Relationen jedoch *nicht* direkt für die Dielektrizitätskonstante $\tilde{\varepsilon}(\omega)$, obwohl auch $\tilde{\varepsilon}(\omega)$ eine kausale Response-Funktion (Antwort von D auf E) darstellt: Wegen

$$n^2(\omega) = \frac{\tilde{\varepsilon}(\omega)}{\varepsilon_0} = 1 + \tilde{\chi}(\omega) \quad (\mu = \mu_0)$$

ist hier die Bedingung (3.24) nur von $\tilde{\chi}(\omega) = \tilde{\varepsilon}(\omega)/\varepsilon_0 - 1$ erfüllt.

Abschließend machen wir uns noch die physikalische Bedeutung von $\operatorname{Re} \varepsilon$ und $\operatorname{Im} \varepsilon$ im Hinblick auf Dispersion und Dissipation klar, indem wir eine elektromagnetische Welle $E, B \sim e^{ikt-i\omega t}$ mit $k = (\omega/c)n = (\omega/c)\sqrt{\varepsilon/\varepsilon_0}$ betrachten. Im Falle einer Wellendämpfung bzw. Absorption durch das Dielektrikum wird $k \equiv \beta + i\alpha/2$ komplex. Die Intensität der Welle ist dann $\sim e^{-\alpha x}$ und fällt mit dem **Absorptionskoeffizienten** α ab, während β die **normale Dispersion** der Welle beschreibt. Für Real- und Imaginärteil von ε gilt:

$$\begin{aligned} \frac{\omega^2}{c^2} \operatorname{Re} \frac{\varepsilon}{\varepsilon_0} &= \operatorname{Re} k^2 = \beta^2 - \frac{\alpha^2}{4} \\ \frac{\omega^2}{c^2} \operatorname{Im} \frac{\varepsilon}{\varepsilon_0} &= \operatorname{Im} k^2 = \beta\alpha \end{aligned}$$

Wenn $\alpha \ll \beta$ ist, gilt $\beta \approx (\omega/c)\sqrt{\operatorname{Re}(\varepsilon/\varepsilon_0)}$, also beschreibt der Realteil von ε die normale Dispersion. Dagegen gilt $\alpha \approx (\operatorname{Im} \varepsilon / \operatorname{Re} \varepsilon) \beta$, also beschreibt der Imaginärteil von ε die Absorptionseigenschaften. Die Kramers-Kronig Relationen stellen also wichtige Beziehungen zwischen verschiedenen optischen Eigenschaften eines Festkörpers (Dispersion und Adsorption) her, weil beide Eigenschaften auf die gleiche Antwortfunktion der Eletronen zurückgehen.

3.4 Literaturverzeichnis Kapitel 3

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis*. 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.
- [4] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.
- [5] M. Hjorth-Jensen. *Computational Physics (Skript)*. Oslo: University of Oslo, 2012.
- [6] J. D. Jackson. *Classical electrodynamics*. 3rd ed. New York, NY: Wiley, 1999.

3.5 Übungen Kapitel 3

1. Harmonischer Oszillator:

Die Wellenfunktionen des harmonischen Oszillators lauten

$$\psi_n(x) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} \exp\left(-\frac{1}{2}x^2\right) H_n(x).$$

- a) Schreiben Sie ein Programm zur Erzeugung der Hermite-Polynome $H_n(x)$ aus der Rekursionsbeziehung

$$\begin{aligned} H_{n+1}(x) &= 2xH_n(x) - 2nH_{n-1}(x), \\ H_0(x) &= 1, \\ H_1(x) &= 2x. \end{aligned}$$

Plotten Sie einige Hermite-Polynome H_n und die Wellenfunktionen ψ_n für $n = 1, 2, 5, 42$.

- b) Überprüfen Sie die Differentialgleichung

$$H_n''(x) - 2xH_n'(x) + 2nH_n(x) = 0$$

mittels numerischer Differentiation für $n = 1, 2, 5$.

2. Integrationsroutine:

Implementieren Sie jeweils eine Integrationsroutine für (i) Trapezregel, (ii) Simpsonregel, (iii) Mittelpunktsregel, an die folgende 4 Argumente übergeben werden sollen: 1) Integrand $f(x)$, 2) untere Grenze a , 3) obere Grenze b , 4) Integrationsintervallbreite h oder Zahl der Integrationsintervalle N (bei der Simpsonregel sollte N gerade sein).

3. Eindimensionale Integrale:

Berechnen Sie folgende Integrale numerisch jeweils mittels (i) Trapezregel, (ii) Simpsonregel. Halbieren Sie bei beiden Verfahren die Intervallbreite h bis die relative Änderung des Ergebnisses kleiner als 10^{-4} wird.

a)

$$I_1 = \int_1^{100} dx \frac{\exp(-x)}{x}$$

b)

$$I_2 = \int_0^1 dx x \sin\left(\frac{1}{x}\right)$$

(Kontrolle: $I_1 \simeq 0.21938$ und $I_2 \simeq 0.37853$.)

4. (Hauptwert-)Integrale:

- a) Berechnen Sie folgendes Hauptwertintegral numerisch:

$$I_1 = \mathcal{P} \int_{-1}^1 dt \frac{e^t}{t}$$

(Kontrolle: $I_1 \simeq 2.1145018$.)

b) Berechnen Sie folgendes Integral numerisch mit einem relativen Fehler $\varepsilon \leq 10^{-5}$:

$$I_2 = \int_0^\infty dt \frac{e^{-t}}{\sqrt{t}}$$

(Kontrolle: $I_2 \simeq 1.77245385$.) Berechnen Sie das Integral analytisch zum Vergleich.

5. Beugung:

Berechnen Sie numerisch die Intensität bei Fraunhoferbeugung an einem Ring $a < |\vec{r}| < 2a$:

$$I(\vec{q}) = \text{const} |u(\vec{q})|^2 \quad \text{mit}$$

$$u(\vec{q}) = \int_{\text{Ring}} d^2 \vec{r} e^{-i\vec{q} \cdot \vec{r}}$$

- a) Schreiben Sie das Integral in Polarkoordinaten. Warum hängt u nur von $q \equiv |\vec{q}|$ ab? Schreiben Sie $u(q)$ als reelles Integral und berechnen Sie $u(q)$ und $I(q)$ analytisch.
- b) Führen Sie dann das 2-dimensionale Integral numerisch aus in Polarkoordinaten (Winkel- und Radialintegration) für $q = 0.1 n/a$ mit $n = 0, 1, 2, \dots, 50$. Plotten Sie die entsprechenden Werte $I(q)$ ($\text{const} = 1$).
- c) Berechnen Sie die Intensität $I(\vec{q})$ auch für einen Viertelring $a < |\vec{r}| < 2a$ und $0 < \phi < \pi/2$ (in Polarkoordinaten), indem Sie das 2-dimensionale Integral wieder numerisch in Polarkoordinaten ausführen. Werten Sie dazu Realteil und Imaginärteil von $u(q_x, q_y)$ getrennt aus, z.B. für $q_x = 0.2 n/a$, $q_y = 0.2 m/a$ mit $n, m = -30, \dots, 0, \dots, 30$.

6. Drehmomente:

Wir betrachten ein zweidimensionales quadratisches System aus identischen magnetischen Dipolen mit magnetischen Momenten \vec{m}_{kl} der Stärke M an Gitterplätzen $\vec{R}_{kl} = k a \vec{e}_x + l a \vec{e}_y$ ($k, l = -N, \dots, -1, 0, 1, \dots, N$, also $(2N+1)^2$ Momente) mit einer Gitterkonstante a , siehe Abb. 3.7.

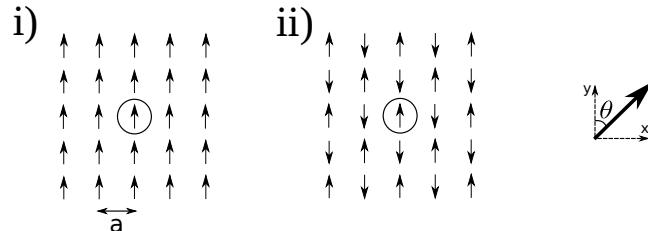


Abbildung 3.7: Quadratgitter mit magnetischen Dipolen.

Wir wollen das Drehmoment auf den magnetischen Dipol in der Mitte des Systems ($k = l = 0$) als Funktion seines Winkels θ_0 mit der y -Achse berechnen. Die übrigen Momente sollen dabei in den Konfigurationen (i) (ferromagnetisch) und (ii) (antiferromagnetisch) (siehe Abb. 3.7) fixiert sein.

- a) Schreiben Sie ein Programm zur Berechnung der Gesamtwechselwirkungsenergie $E(N, \theta_0)$ des Moments in der Mitte mit allen übrigen Momenten als Funktion des Winkels θ_0 für Konfigurationen (i) und (ii). Gehen Sie dabei von der Wechselwirkungsenergie

$$E = \frac{\mu_0}{4\pi} \frac{1}{|\vec{R}|^3} \left(-3(\hat{\vec{R}} \cdot \vec{m})(\hat{\vec{R}} \cdot \vec{n}) + (\vec{m} \cdot \vec{n}) \right)$$

zwischen zwei Dipolmomenten \vec{m} und \vec{n} mit Abstandsvektor \vec{R} ($\hat{\vec{R}} \equiv \vec{R}/|\vec{R}|$ ist der zugehörige Einheitsvektor) aus. Plotten Sie die Funktionen $E(N, \theta_0)$ für $N = 2, 5, 10$ jeweils für Konfiguration (i) und (ii).

b) Differenzieren Sie numerisch nach θ_0 , um den Betrag des Drehmoments $T(N, \theta_0) = |\partial E/\partial\theta_0|$ auf das Moment in der Mitte zu berechnen. In welche Richtung zeigt der Vektor \vec{T} des Drehmomentes? Plotten Sie $T(N, \theta_0)$ für $N = 2, 5, 10$ jeweils für Konfiguration (i) und (ii).

Kontrollieren Sie ihr Ergebnis, indem Sie den Drehmomentvektor direkt über $\vec{T} = \vec{m} \times \vec{B}(0)$ berechnen, wobei nun das Gesamtmagnetfeld $\vec{B}(0)$, das durch die anderen Momente in der Mitte bei $\vec{R} = 0$ erzeugt wird, numerisch zu berechnen ist.

7. Elektrostatik:

Wir berechnen das elektrostatische Potential für zwei Ladungsverteilungen in einem Würfel der Kantenlänge $2a$ numerisch durch direkte dreidimensionale Integration.

a) Zuerst betrachten wir eine homogene Ladungsverteilung

$$\rho(x, y, z) = \begin{cases} \rho_0 & |x| < a, |y| < a, |z| < a \\ 0 & \text{sonst} \end{cases}$$

und berechnen das elektrostatische Potential auf der x -Achse:

$$\phi(x) = \frac{1}{4\pi\epsilon_0} \int dx' \int dy' \int dz' \frac{\rho(x', y', z')}{[(x - x')^2 + y'^2 + z'^2]^{1/2}}. \quad (3.34)$$

Führen Sie eine geeignete Wahl der Einheiten für ϕ und x ein, um das numerisch zu berechnenden Integral einheitenslos zu machen (Computer kennen keine Einheiten...).

Führen Sie dann das 3-dimensionale Integral (3.34) numerisch aus, zunächst für x -Werte $x/a = 0.1n$ mit $n = 11, 12, \dots, 80$ außerhalb des Würfels. Plotten Sie die entsprechenden Werte $\phi(x)$. Welche Asymptotik erwarten Sie für große x (Stichwort Multipolentwicklung)? Überprüfen Sie Ihre Vermutung.

b) Versuchen Sie das Integral auch für $|x| < a$ innerhalb des Würfels (x -Werte $x/a = 0.1n$ mit $n = 0, 1, 2, \dots, 10$) numerisch auszuwerten. Könnte es Probleme geben?

c) Nun betrachten wir die Ladungsverteilung

$$\rho(x, y, z) = \begin{cases} \rho_0 x/a & |x| < a, |y| < a, |z| < a \\ 0 & \text{sonst} \end{cases}$$

Überlegen Sie sich wieder, wie Sie das numerisch zu berechnende Integral einheitenslos machen.

Führen Sie dann wieder das 3-dimensionale Integral (3.34) numerisch aus, außerhalb und innerhalb des Würfels, d.h. für x -Werte $x/a = 0.1n$ mit $n = 0, 1, 2, \dots, 80$. Plotten Sie die entsprechenden Werte $\phi(x)$. Welche Asymptotik erwarten Sie nun für große x ? Berechnen Sie das erste nicht-verschwindende Multipolmoment und überprüfen Sie das zu erwartende asymptotische Verhalten.

4 Gewöhnliche Differentialgleichungen

Literatur zu diesem Teil:

Dies ist ein zentrales Thema in der Computerphysik, auch im Hinblick auf dynamische Systeme und Chaos oder Molekulardynamiksimulationen. Es gibt umfangreiche Literatur, sowohl von Seiten der numerischen Mathematik, z.B. Numerical Recipes [1, 2], Stoer [3], Hamming [4], als auch von Seiten der Computerphysik, z.B. Koonin/Meredith [5], Gould/Tobochnik [6], Kinzel [7] und Frenkel [8].

4.1 Reduktion auf DGL erster Ordnung

Wir zeigen, wie sich jede gewöhnliche DGL auf eine DGL 1. Ordnung reduzieren lässt.

Die allgemeinste gewöhnliche Differentialgleichung (DGL) n-ter Ordnung für eine Funktion $y = y(x)$ ist von der Form

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)}) \quad (4.1)$$

Im Folgenden könnte die skalare Funktion $y(x)$ auch problemlos durch eine vektorwertige Funktion $\vec{y}(x)$ ersetzt werden:

$$\vec{y}^{(n)} = f(x, \vec{y}, \vec{y}', \vec{y}'', \dots, \vec{y}^{(n-1)}) \quad (4.2)$$

Bei einer **gewöhnlichen DGL** bleibt das Argument x allerdings immer ein **Skalar**. Erst bei **partiellen DGLn**, die später behandelt werden, ist auch das Argument \vec{x} ein Vektor und es treten partielle Ableitungen $\frac{\partial}{\partial x_1} y, \frac{\partial}{\partial x_2} y, \dots$, usw. auf.

Die allgemeinste DGL (4.1) für $y(x)$ kann immer in ein **System von n Gleichungen 1. Ordnung** umgewandelt werden. Dazu führen wir einen n-komponentigen Vektor $\vec{y}(x)$ ein, der aus den Ableitungen $y^{(0)}$ bis $y^{(n-1)}$ besteht,

$$\vec{y} \equiv \begin{pmatrix} y_1 & = & y \\ y_2 & = & y' \\ \vdots & & \\ y_n & = & y^{(n-1)} \end{pmatrix} \quad (4.3)$$

Damit lässt sich die DGL (4.1) als eine DGL 1. Ordnung für den Vektor $\vec{y}(x)$ schreiben:

$$\vec{y}' = \begin{pmatrix} y'_1 \\ y'_2 \\ \vdots \\ y'_{n-1} \\ y'_n \end{pmatrix} = \begin{pmatrix} y_2 \\ y_3 \\ \vdots \\ y_n \\ f(x, y_1, y_2, \dots, y_n) \end{pmatrix}$$

(4.4)

Damit haben wir uns klar gemacht, dass es völlig ausreichend ist, (vektorwertige) gewöhnliche DGLn 1. Ordnung der Form

$$\vec{y}' = \vec{f}(x, \vec{y})$$

(4.5)

numerisch lösen zu können.

Wir wollen uns einige besonders wichtige **Anwendungen** gewöhnlicher DGLn in der Physik klar machen:

- Jede **Bewegungsgleichung** in der Mechanik, sei sie als Newtonsche Bewegungsgleichung oder im Hamiltonformalismus im Phasenraum formuliert, ist eine gewöhnliche DGL bei der x der Zeit t entspricht. Wir betrachten N Teilchen mit jeweils f Freiheitsgraden, deren Bahnen durch einen Nf -dimensionalen Vektor $\vec{r}(t)$ beschrieben werden. Beim Übergang von der Formulierung der Bewegung in einem beliebigen Kraftfeld \vec{F} als DGL 2. Ordnung nach Newton,

$$\ddot{\vec{r}} = \vec{F}(\vec{r}, \dot{\vec{r}}, t),$$

zu einer DGL 1. Ordnung für den Vektor

$$\vec{y} = \begin{pmatrix} \vec{r} \\ \vec{p} \end{pmatrix}$$

im Phasenraum nach Hamilton,

$$\boxed{\begin{aligned} \dot{\vec{r}} &= \vec{p}/m \\ \dot{\vec{p}} &= \vec{F}(\vec{r}, \vec{p}, t), \end{aligned}} \quad (4.6)$$

wird im Prinzip der gleiche ‘‘Trick’’ benutzt wie beim obigen Übergang von (4.1) zu (4.4).

- Ist die Teilchenzahl N sehr groß, bekommen wir den Übergang zur **statistischen Mechanik**. Die im nächsten Kapitel diskutierte **Molekulardynamik(MD)-Simulation** macht im Prinzip nichts anderes als alle N Newtonsches Bewegungsgleichungen numerische zu lösen. In diesem Fall sieht man auch, dass die Berechnung der ‘‘rechten Seite’’ \vec{f} der DGL (4.5) in diesem Fall der Berechnung aller Kräfte \vec{F} in einem N -Teilchensystem entspricht. Bei Paarwechselwirkungen sind dies $N(N-1)/2 \sim N^2$ Kräfte, die bei *jeder* Berechnung von \vec{f} effektiv auszurechnen sind. Daher ist es bei numerischen Verfahren zur Lösung von (4.5) in der Physik durchaus ein Leistungskriterium wie oft \vec{f} in jedem Zeitschritt berechnet werden muss.
- Ist die Zahl der Freiheitsgrade Nf groß, können wir auch auch den Übergang zum **Kontinuum** vollziehen. Ein Beispiel ist die diskretisierte Darstellung einer schwingenden Saite durch N gekoppelte Massen. Daher führt dieser Limes auch auf die Lösung **partieller DGLn**, wie der Wellengleichung im Fall der schwingenden Saite.

4.2 Euler-Verfahren, Prädiktor-Korrektor

Es werden einfachste Verfahren für DGLn 1. Ordnung in der Zeit diskutiert. Das Euler-Verfahren beruht auf einer Diskretisierung der Zeitableitung, das Prädiktor-Korrektor auf einer Darstellung der Zeitentwicklung als Integral und Anwendung der Trapezregel.

Die typische **zeitabhängige DGL** 1. Ordnung in der Physik

$$\boxed{\dot{\vec{y}} = \vec{y}' = \vec{f}(t, \vec{y})} \quad (4.7)$$

mit **Anfangsbedingungen** $\vec{y}(0) = \vec{y}_0$ wird grundsätzlich in allen Verfahren durch **Diskretisierung** in der Zeit gelöst. Dazu diskretisiert man das Zeitintervall $t \in [0, T]$, in dem die Lösung gesucht ist, mit einer **Schrittweite** h in $N = T/h$ diskrete Zeitschritte $t_n = nh$ mit $n = 0, \dots, N$. Gesucht sind dann die Funktionswerte $\vec{y}_n = \vec{y}(t_n)$.

Das einfachste Lösungsverfahren ist das sogenannte Euler-Verfahren, das auf einer einfachen Diskretisierung der Zeitableitung auf der linken Seite der DGL (4.7) beruht wie bei der numerischen Differentiation (3.1):

$$\begin{aligned}\dot{\vec{y}} &= \frac{\vec{y}_{n+1} - \vec{y}_n}{h} + \mathcal{O}(h) \\ \frac{\vec{y}_{n+1} - \vec{y}_n}{h} &= \vec{f}(t_n, \vec{y}_n) + \mathcal{O}(h)\end{aligned}$$

Dies führt auf eine Rekursion für \vec{y}_n ,

$$\boxed{\vec{y}_{n+1} = \vec{y}_n + h\vec{f}(t_n, \vec{y}_n) + \mathcal{O}(h^2)} \quad (4.8)$$

die das **Euler-Verfahren** darstellt. Es besitzt folgende Eigenschaften:

- Der Fehler für *einen* Schritt beträgt $\mathcal{O}(h^2)$; der akkumulierte und fortgepflanzte Fehler bei $N = T/h$ Schritten für die Integration von $t_0 = 0$ bis $t_N = T$ ist dann von der Ordnung $\mathcal{O}(Nh^2) = \mathcal{O}(h)$. Ein solches Verfahren wird dann nach dem akkumulierten Fehler als **Verfahren 1. Ordnung** bezeichnet.
- Das Euler-Verfahren ist ein **Einschrittverfahren**: Um \vec{y}_{n+1} zu bestimmen, wird nur Information bei t_n und \vec{y}_n benötigt, daher ist das Verfahren auch “asymmetrisch” und es kam die numerische Rechtsableitung zum Einsatz. Es wird nur eine Funktionsberechnung von \vec{f} in jedem Schritt benötigt.
- Für die Praxis sollte man sich merken, dass das Euler-Verfahren einfach zu implementieren ist und \vec{f} in jedem Schritt nur 1-mal berechnet werden muss. Auf der anderen Seite werden wir noch weitaus genauere Verfahren kennenlernen, die aber aufwendiger sind.
- Die Genauigkeitsangabe beruht auf der Annahme, dass die Funktion \vec{f} zumindest stetig ist, ebenso wie der Genauigkeitsgewinn bei den aufwendigeren genaueren Verfahren, die wir noch kennenlernen werden. Es gibt physikalische Anwendungen bei *stochastischen Prozessen*, bei denen die rechte Seite tatsächlich unstetige Zufallskräfte enthält, die thermisches Rauschen beschreiben. Bei solchen **stochastischen Bewegungsgleichungen** kommen tatsächlich einfache Euler-Verfahren zum Einsatz, da die aufwendigeren Verfahren auch keinen Gewinn mehr garantieren können, siehe Kapitel (13.2).

Es ist einfach, Verbesserungen des Euler-Verfahrens zu finden, die einen geringeren Fehler aufweisen. Dazu ist es hilfreich, die Zeitentwicklung von \vec{y}_n nach \vec{y}_{n+1} zunächst exakt als Integration zu schreiben:

$$\vec{y}_{n+1} = \vec{y}_n + \int_{t_n}^{t_{n+1}} dt \vec{f}(t, \vec{y}(t)) \quad (4.9)$$

Dann können wir die numerischen Integrationsmethoden aus Kapitel (3.2) verwenden, um Lösungsverfahren zu gewinnen. Zunächst macht man sich klar, dass das Euler-Verfahren dann einer sehr schlechten Näherung

$$\int_{t_n}^{t_{n+1}} dt \vec{f}(t, \vec{y}(t)) = h \vec{f}(t_n, \vec{y}_n) + \mathcal{O}(h^2)$$

entspricht, die sich schon durch Anwendung der einfachen *Trapezregel* (3.6) um eine Größenordnung verbessern lässt:

$$\begin{aligned}\int_{t_n}^{t_{n+1}} df \vec{f}(t, \vec{y}(t)) &= \frac{h}{2} \left[\vec{f}(t_n, \vec{y}_n) + \vec{f}(t_{n+1}, \vec{y}_{n+1}) \right] + \mathcal{O}(h^3) \\ \vec{y}_{n+1} &= \vec{y}_n + \frac{h}{2} \left[\vec{f}(t_n, \vec{y}_n) + \vec{f}(t_{n+1}, \vec{y}_{n+1}) \right] + \mathcal{O}(h^3)\end{aligned} \quad (4.10)$$

Dabei tritt nun allerdings das Problem auf, dass dies keine Rekursion in n mehr darstellt, da wir auf der rechten Seite $\vec{f}(t_{n+1}, \vec{y}_{n+1})$, also insbesondere \vec{y}_{n+1} zur Zeit t_{n+1} bereits kennen müssen,

wir allerdings erst bei der Zeit t_n angekommen sind. Eine typische Lösung dieses Problems besteht darin, \vec{y}_{n+1} mit einem einfachen Euler-Verfahren “vorherzusagen” (einen **Prädiktor** zu generieren):

$$\boxed{\begin{aligned}\vec{k}_1 &= h\vec{f}(t_n, \vec{y}_n) \\ \vec{y}_{n+1,p} &= \vec{y}_n + \vec{k}_1\end{aligned}} \quad (4.11)$$

Dann kann man mit Hilfe von (4.10) den Prädiktor “korrigieren” (den **Korrektor** berechnen)

$$\boxed{\begin{aligned}\vec{k}_2 &= h\vec{f}(t_{n+1}, \vec{y}_{n+1,p}) \\ \vec{y}_{n+1} &= \vec{y}_n + \frac{1}{2}(\vec{k}_1 + \vec{k}_2) + \mathcal{O}(h^3)\end{aligned}} \quad (4.12)$$

Die Rekursionen (4.11) zusammen mit (4.12) stellen das **Heun-Verfahren 2. Ordnung** dar; es ist die einfachste Version eines **Prädiktor-Korrektor Verfahrens**, das folgende Eigenschaften besitzt:

- Der Fehler für *einen* Schritt ist $\mathcal{O}(h^3)$ und damit um eine Größenordnung kleiner als beim einfachen Euler-Verfahren. Der akkumulierte Fehler bei $N = T/h$ Schritten ist dann $\mathcal{O}(Nh^3) = \mathcal{O}(h^2)$ und damit ist dies ein **Verfahren 2. Ordnung**.
- Die Vorhersage in (4.11) muss nur auf $\mathcal{O}(h^2)$ genau sein (Euler), damit der korrigierte Wert in (4.12) einen kleineren Fehler $\mathcal{O}(h^3)$ hat.
- Typischerweise macht der Korrektor das Verfahren auch numerisch **stabiler** (siehe Hamming [4], Kapitel 22.5, 22.6).
- Das Prädiktor-Korrektor Verfahren ist genauer als das Euler-Verfahren, allerdings muss \vec{f} in jedem Schritt 2-mal berechnet werden.

Wir haben hier zunächst nur die einfachste Version eines Prädiktor-Korrektor Verfahrens kennengelernt. Aufwendigere Versionen werden in Kapitel 4.7 noch einmal kurz erläutert werden.

4.3 Runge-Kutta Verfahren

Runge-Kutta Verfahren verwenden Zwischenschritte. Wir diskutieren die Runge-Kutta Verfahren 2. Ordnung und 4. Ordnung. Das Runge-Kutta Verfahren 4. Ordnung stellt das genaueste im Rahmen dieser Vorlesung behandelte Verfahren dar.

Anwendung der Trapezregel in der Integraldarstellung (4.9) hatte uns auf das Prädiktor-Korrektor Verfahren geführt. Mit gleicher Genauigkeit kann man die **Mittelpunktsregel** (3.7) benutzen. Dies wird auf das **Runge-Kutta-Verfahren 2. Ordnung** führen. Mit höherer Genauigkeit können wir die **Simpsonregel** (3.8) verwenden, die dann auf das **Runge-Kutta-Verfahren 4. Ordnung** führt.

4.3.1 Runge-Kutta 2. Ordnung

Zur Herleitung des Runge-Kutta Verfahrens 2. Ordnung verwenden wir die Mittelpunktsregel (3.7) in der exakten Integraldarstellung (4.9) eines DGL-Integrationsschrittes,

$$\begin{aligned}\int_{t_n}^{t_{n+1}} dt \vec{f}(t, \vec{y}(t)) &= h\vec{f}(t_{n+1/2}, \vec{y}_{n+1/2}) + \mathcal{O}(h^3) \\ \vec{y}_{n+1} &= \vec{y}_n + h\vec{f}(t_{n+1/2}, \vec{y}_{n+1/2}) + \mathcal{O}(h^3)\end{aligned}$$

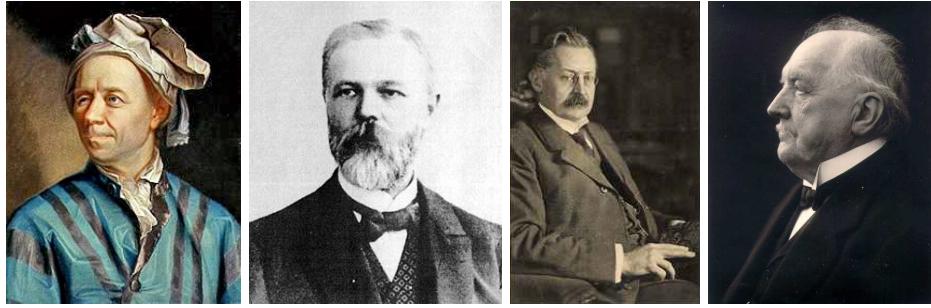


Abbildung 4.1: Von links nach rechts: Leonhard Euler (1707-1783), Mathematiker und Physiker. Karl Heun (1859-1929), deutscher Mathematiker. Carl David Tolm   Runge (1856-1927), Mathematiker und Physiker (auch bekannt vom Lenz-Runge-Vektor in der Mechanik). Martin Wilhelm Kutta (1867-1944), deutscher Mathematiker. (Quelle: Wikipedia).

wobei wir einen **Zwischenschritt** in der Intervallmitte bei $t = t_{n+1/2} = \frac{1}{2}(t_n + t_{n+1})$ eingef  hrt haben. Auch hier tritt das Problem auf, dass wir $\vec{y}_{n+1/2}$ in der Intervallmitte aber noch nicht kennen, wenn wir erst bis t_n integriert haben. Wir benutzen wieder das Euler-Verfahren, um $\vec{y}_{n+1/2} = \vec{y}_n + \frac{1}{2}h\vec{f}(t_n, \vec{y}_n) + \mathcal{O}(h^2)$ zu approximieren und erhalten damit das **Runge-Kutta Verfahren 2. Ordnung**

$$\boxed{\begin{aligned}\vec{k}_1 &= h\vec{f}(t_n, \vec{y}_n) \\ \vec{k}_2 &= h\vec{f}\left(t_{n+1/2}, \vec{y}_n + \frac{1}{2}\vec{k}_1\right) \\ \vec{y}_{n+1} &= \vec{y}_n + \vec{k}_2 + \mathcal{O}(h^3)\end{aligned}} \quad (4.13)$$

Die wichtigsten Eigenschaften des Runge-Kutta Verfahrens 2. Ordnung sind

- Der Wert \vec{y}_{n+1} wird   ber einen Zwischenschritt bei $t_{n+1/2}$ berechnet.
- Die Approximation am Zwischenschritt   hnelt dem Pr  diktor-Korrektor Verfahren, dort wird aber \vec{y}_{n+1} f  r den *ganzen* Schritt vorhergesagt.
- Wie der Name bereits sagt, ist dies ein **Verfahren 2. Ordnung** mit einem Fehler $\mathcal{O}(h^3)$ in jedem Schritt und einem akkumulierten Fehler $\mathcal{O}(h^2)$.
- Ein Fehler $\mathcal{O}(h^2)$ in \vec{k}_1 bei der Approximation von $\vec{y}_{n+1/2}$ (mit Euler) reicht aus, um einen Fehler $\mathcal{O}(h^3)$ f  r \vec{y}_{n+1} zu gew  hrleisten.
- Das Verfahren ist damit genauer als ein Euler-Verfahren, allerdings muss \vec{f} in jedem Schritt 2-mal berechnet werden.

4.3.2 Runge-Kutta 4. Ordnung

Zur Herleitung des Runge-Kutta Verfahrens 4. Ordnung verwenden wir die genauere Simpsonregel (3.8) in der exakten Integraldarstellung (4.9) eines DGL-Integrationsschrittes,

$$\begin{aligned}\int_{t_n}^{t_{n+1}} dt \vec{f}(t, \vec{y}(t)) &= \frac{h}{6} \left[\vec{f}(t_n, \vec{y}_n) + 4\vec{f}(t_{n+1/2}, \vec{y}_{n+1/2}) + \vec{f}(t_{n+1}, \vec{y}_{n+1}) \right] + \mathcal{O}(h^5) \\ \vec{y}_{n+1} &= \vec{y}_n + \frac{h}{6} \left[\vec{f}(t_n, \vec{y}_n) + 2\vec{f}(t_{n+1/2}, \vec{y}_{n+1/2}) + 2\vec{f}(t_{n+1/2}, \vec{y}_{n+1/2}) + \vec{f}(t_{n+1}, \vec{y}_{n+1}) \right] + \mathcal{O}(h^5)\end{aligned} \quad (4.14)$$

Runge-Kutta Methods

General form RK2 (2nd order) $\mathcal{O}(\Delta t^2)$

$$\begin{aligned}\dot{x} &= f(x_n, t_n) \\ \cdot K_1 &= \Delta t f(t_n, x_n) \\ \cdot K_2 &= \Delta t f(t_n + \alpha \Delta t, x_n + \beta K_1) \\ x_{n+1} &= x_n + a K_1 + b K_2\end{aligned}$$

 $a, b, \alpha, \beta = ? \rightarrow$ Constants subject to constraints.Taylor series in Δt

$$\begin{aligned}x_{n+1} &= x(t_n + \Delta t) = x_n(t_n) + \Delta t \dot{x}(t_n) + \frac{(\Delta t)^2}{2} \ddot{x}(t_n) + \dots \\ \textcircled{1} \quad x_{n+1} &= x_n + \Delta t f(t_n, x_n) + \frac{(\Delta t)^2}{2} [f_t(t_n, x_n) + f(t_n, x_n) f_x(t_n, x_n)] \\ &\quad \uparrow \text{time} \qquad \uparrow \text{space derivatives}\end{aligned}$$

Compare eq. 1 with 2:

$$\begin{aligned}a+b &= 1 \\ ab &= \frac{1}{2} \\ a\beta &= \frac{1}{2}\end{aligned} \left. \begin{array}{l} \text{constraints} \\ \text{constraints} \end{array} \right\}$$

Examples: Modified Euler's Method

$$\textcircled{2} \quad a=b=\frac{1}{2}, \alpha=\beta=1$$

$$K_1 = \Delta t f(t_n, x_n), K_2 = \Delta t f(t_n + \Delta t, x_n + K_1)$$

$$x_{n+1} = x_n + \frac{1}{2}(K_1 + K_2)$$

$$\textcircled{2} \quad \text{Mid point: } \alpha=\beta=\frac{1}{2}, b=1, a=0$$

$$K_1 = \Delta t f(t_n, x_n), K_2 = \Delta t f(t_n + \frac{1}{2} \Delta t, x_n + \frac{1}{2} K_1)$$

$$x_{n+1} = x_n + K_2$$

Higher Order RKs:

RK3:

$$\begin{aligned}\dot{x} &= f(x, t) \\ \text{3 stages} \quad \left[\begin{aligned}K_1 &= \Delta t f(t_n, x_n) \\ K_2 &= \Delta t f(t_n + \alpha \Delta t, x_n + \beta K_1) \\ K_3 &= \Delta t f(t_n + \gamma \Delta t, x_n + \delta K_1 + \epsilon K_2)\end{aligned} \right] \\ x_{n+1} &= x_n + a K_1 + b K_2 + c K_3\end{aligned}$$

$$\begin{array}{ll} \text{Orders:} & 2 \quad 3 \quad 4 \quad \textcircled{5} \quad 6 \quad \textcircled{7} \quad 8 \\ \text{Stages:} & 2 \quad 3 \quad 4 \quad \textcircled{6} \quad 7 \quad \textcircled{9} \quad 11 \end{array}$$

Central question: How to find a proper value for Δt ?

Dormand-Prince Method: (Matlab)

5th order RK / 4th order RK

 $\mathcal{O}(\Delta t^5) / \mathcal{O}(\Delta t^4) \rightarrow$ Calculate $x_n \rightarrow x_{n+1} / \hat{x}_{n+1}$ $\varepsilon \rightarrow$ tolerance = desired error

$$\frac{\varepsilon_n}{\varepsilon} = \frac{(\Delta t)^5}{(\Delta t)^4} = \Delta t = s \cdot \Delta t \left(\frac{\varepsilon}{\varepsilon_n} \right)^{1/5}$$

 \hookrightarrow time step 1 should have used.

$$e_n = |x_n - \hat{x}_{n+1}| \sim \mathcal{O}(\Delta t^5)$$

 \rightarrow two scenarios:

- ① $e_n > t \rightarrow$ reject
- ② $e_n < t \rightarrow$ accept & rescale

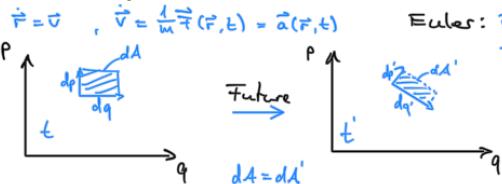
 \hookrightarrow safety factor $s = 0.9$

Stiff differential equation

$$y'(x) = \frac{y(x+h) - y(x-h)}{h}$$

Implicit methods are unconditionally stable.
 $\hookrightarrow h$ independent

Newton's equation of motion (Hamiltonian dynamics)

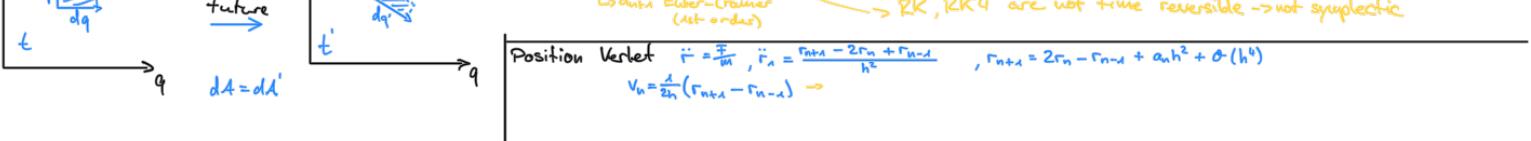
 $\dot{P} = \dot{V}, \dot{V} = \frac{1}{m} \vec{F}(P, t) = \vec{a}(P, t)$ Future \rightarrow $dA = dA'$

$$\begin{aligned}y' &= -50(y - \cos x) \\ &\quad \text{slow} \\ &\quad \text{fast}\end{aligned}$$

Adams-Moulton

$$y_{n+1} = y_n + \frac{1}{2}$$

Problem: Space and momentum defined at the same time

 \rightarrow inherently unstableQM: $E_q, p_J = i\hbar \rightarrow$ Classical: $E_q, p_J^2 = 1$ \hookrightarrow RK, RKF4 are not time reversible \rightarrow not symplectic

$$\begin{aligned}\ddot{r} &= \frac{\vec{r}_n}{m}, \ddot{r}_n = \frac{r_{n+1} - 2r_n + r_{n-1}}{h^2}, r_{n+1} = 2r_n - r_{n-1} + a_n h^2 + \mathcal{O}(h^4) \\ v_n &= \frac{1}{2h}(r_{n+1} - r_{n-1})\end{aligned}$$

wobei wir wieder einen **Zwischenschritt** in der Intervallmitte bei $t = t_{n+1/2} = \frac{1}{2}(t_n + t_{n+1})$ eingeführt haben. Beim Runge-Kutta Verfahren 4. Ordnung werden die Werte $\vec{y}_{n+1/2}$ und \vec{y}_{n+1} auf der rechten Seite, die bei $t = t_n$ noch nicht bekannt sind, nun in 4 Schritten approximiert:

$$\boxed{\begin{aligned} 1) \quad & \vec{k}_1 = h \vec{f}(t_n, \vec{y}_n) \\ 2) \quad & \vec{k}_2 = h \vec{f}\left(t_{n+1/2}, \vec{y}_n + \frac{1}{2} \vec{k}_1\right) \end{aligned}} \quad (4.15)$$

Diese beiden Schritte sind identisch zum Runge-Kutta Verfahren 2. Ordnung. Hier wird die Steigung¹ \vec{k}_1 am Intervallanfang $t = t_n$ benutzt, um $\vec{y}_{n+1/2}$ zunächst mittels eines einfachen Euler-Verfahrens zu approximieren (Schritt 1). Damit wird die Steigung \vec{k}_2 in der Intervallmitte approximativ berechnet (Schritt 2). Diese Approximation der Steigung in der Intervallmitte wird verbessert, indem (ähnlich wie beim Korrektor im Prädiktor-Korrektor Verfahren) $\vec{y}_{n+1/2}$ noch einmal mit dem Wert der Steigung in der Intervallmitte \vec{k}_2 berechnet wird (statt mit \vec{k}_1):

$$\boxed{3) \quad \vec{k}_3 = h \vec{f}\left(t_{n+1/2}, \vec{y}_n + \frac{1}{2} \vec{k}_2\right)} \quad (4.16)$$

Schließlich wird \vec{y}_{n+1} mit der verbesserten Steigung \vec{k}_3 in der Intervallmitte approximiert, um die Steigung \vec{k}_4 am Intervallende zu bekommen:

$$\boxed{4) \quad \vec{k}_4 = h \vec{f}\left(t_{n+1}, \vec{y}_n + \vec{k}_3\right)} \quad (4.17)$$

Die 4 Steigungen sind in Abb. 4.2 veranschaulicht. Dann werden die 4 Steigungen auf der rechten Seite in (4.14) eingesetzt, um den Integrationsschritt zu vervollständigen:

$$\boxed{\vec{y}_{n+1} = \vec{y}_n + \frac{1}{6} [\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4] + \mathcal{O}(h^5)} \quad (4.18)$$

Die Schritte (4.15), (4.16), (4.17) und (4.18) definieren einen Integrationsschritt im **Runge-Kutta Verfahren 4. Ordnung**.

Die wichtigsten Eigenschaften dieses Verfahrens sind

- Wie der Name bereits sagt, ist dies ein **Verfahren 4. Ordnung** mit einem Fehler $\mathcal{O}(h^5)$ in jedem Schritt und einem akkumulierten Fehler $\mathcal{O}(h^4)$.
- Die Fehler sind $\mathcal{O}(h^3)$ für die Beiträge von \vec{k}_2 , $\mathcal{O}(h^4)$ für die Beiträge von \vec{k}_3 und $\mathcal{O}(h^5)$ für die Beiträge von \vec{k}_4 ; im letzten Schritt (4.18) werden die Beiträge aber genau so kombiniert, dass nur noch ein Fehler $\mathcal{O}(h^5)$ überlebt. Den Beweis dafür haben wir hier nicht gegeben, dieser kann z.B. durch konsequente Taylorentwicklung erbracht werden.
- Das Verfahren ist noch genauer als Runge-Kutta 2. Ordnung, allerdings muss hier \vec{f} in jedem Schritt sogar 4-mal berechnet werden.

Das Runge-Kutta Verfahren 4. Ordnung ist für viele Anwendungen in der Physik das Verfahren der Wahl und gibt eine sehr gute Genauigkeit. Es kann für praktische Anwendungen noch angepasst und beschleunigt werden durch eine Schrittweitenanpassung. Es ist insbesondere das Verfahren der Wahl für Bewegungen $\vec{y}(t)$ mit wenigen Freiheitsgraden (= Dimension des Vektors \vec{y}), z.B. bei allen Anwendungen in der Mechanik, wenn es um die Dynamik einer überschaubaren Anzahl von Teilchen geht. Bei sehr vielen Freiheitsgraden wie bei einer typischen MD-Simulation für eine große Zahl N von Teilchen ($N \sim 10^5$ ist nicht ungewöhnlich) wird die 4-malige Berechnung der rechten Seite \vec{f} (die ja alle Wechselwirkungskräfte enthält) in jedem Zeitschritt zunehmend zum Problem. Dann verwendet man eher Verfahren niedrigerer Ordnung, bei MD-Simulationen speziell den Verlet-Algorithmus, der unten besprochen wird.

¹ Genaugenommen sind im Folgenden immer \vec{k}_i/h die Steigungen von $\vec{y}(t)$.

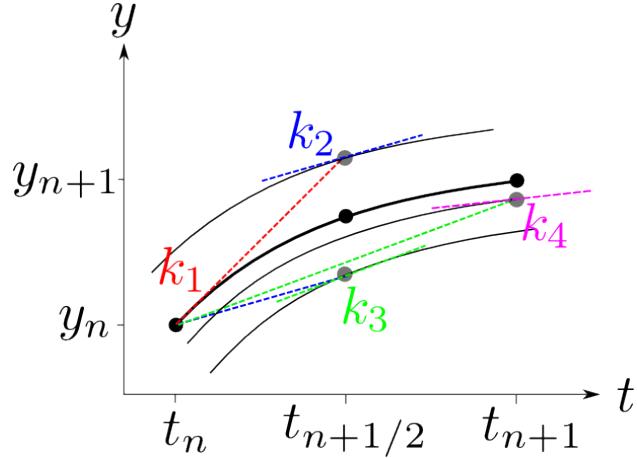


Abbildung 4.2: Grafische Veranschaulichung der 4 Schritte im Runge-Kutta Verfahren. Schwarze Linien zeigen Lösungschar $y' = f(y, t)$ zu verschiedenen Anfangsbedingungen. Im Schritt 1) (rot) wird die Steigung k_1 am Intervallanfang y_n ausgewertet. In Schritt 2) (blau) wird die Steigung k_2 in der Intervallmitte ermittelt, indem $y_{n+1/2}$ mit Hilfe der Steigung k_1 von y_n aus approximiert wird. In Schritt 3) (grün) wird die Steigung k_3 in der Intervallmitte ermittelt, indem $y_{n+1/2}$ mit Hilfe der Steigung k_2 von y_n aus approximiert wird. In Schritt 4) (violett) wird die Steigung k_4 am Intervallende berechnet, indem y_{n+1} mit Hilfe der Steigung k_3 von y_n aus approximiert wird.

4.4 Schrittweitenanpassung

Die DGL-Integration mit Euler- oder Runge-Kutta Verfahren kann in der Praxis stark beschleunigt werden durch dynamische Anpassung der Schrittweite h .

Ein wichtiger Vorteil der Euler- und Runge-Kutta-Verfahren besteht darin, dass jeder Integrations schritt *unabhängig* von vergangenen Schritten ist, d.h. um von t_n nach t_{n+1} zu integrieren, ist keine Information aus Zeiten $t < t_n$ notwendig. Dies ist beispielsweise anders bei aufwendigeren Prädiktor-Korrektor Verfahren, siehe Kapitel 4.7 unten. Aus dieser Eigenschaft der Euler- und Runge-Kutta-Verfahren folgt, dass die Schrittweite h von Schritt zu Schritt *während* der numerischen Lösung angepasst werden kann.

Dies kann die Lösung sehr beschleunigen, da es Bereiche gibt, wo die rechte Seite \vec{f} der DGL (4.7) nur schwach variiert und man daher große Schritte nehmen kann. Auf der anderen Seite kann man in Bereichen mit schnell variiierendem \vec{f} die Genauigkeit nur bei kleiner Schrittweite gewährleisten. Ein Beispiel, wäre die Lösung der Newtonschen Bewegungsgleichung für ein Teilchen in einem kompliziert geformten Potential, siehe Abb. 4.3.

Oft ist dabei nicht im Voraus klar, ob und wann die Schrittweite verkleinert werden muss oder vergrößert werden kann. Daher ist eine **adaptive Schrittweite** gefragt, die sich “automatisch” anpasst, so dass eine gegebenes Genauigkeitsziel erfüllt bleibt.

Die **Schrittweitenanpassung** beruht daher erst einmal darauf zu jedem Zeitpunkt die Genauigkeit abschätzen zu können. Dazu stellen wir zwei Verfahren vor:

- 1) Wir vergleichen die Ergebnisse von 2 Schritten mit h (genauer) und 1 Schritt mit $2h$ (unge nauer) für ein Verfahren $(n-1)$ -ter Ordnung (Euler $n = 2$; Runge-Kutta 2. Ordnung $n = 3, 4$. Ordnung $n = 5$). Dann gilt für die Abweichung $\Delta y \equiv |\vec{y}_{2h} - \vec{y}_{2x1h}|$

$$\Delta y \sim h^n \quad (4.19)$$

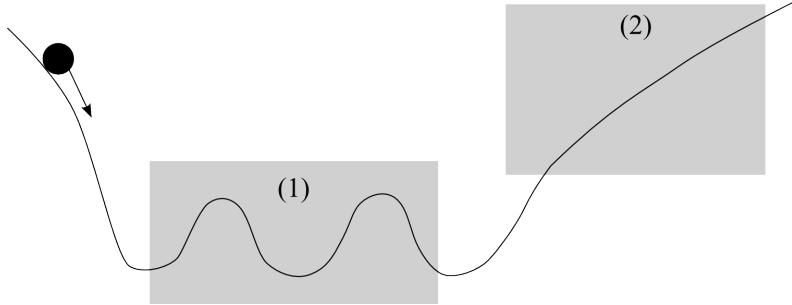


Abbildung 4.3: Teilchen in einem komplizierten Potential. Im Bereich (1) mit schnell variierendem Potential sollte die Schrittweite klein sein, um Genauigkeit zu gewährleisten, im Bereich (2) kann die Schrittweite groß werden um die numerische Lösung zu beschleunigen.

Δy ist eine lokale Fehlerabschätzung. Der Mehraufwand, um Δy zu berechnen bei einem Runge-Kutta Verfahren 4. Ordnung, kann folgendermaßen abgeschätzt werden: Bei 2 Schritten h muss \vec{f} 8mal berechnet werden, bei 1 Schritt $2h$ (um den Vergleich zu haben), muss \vec{f} zusätzlich 3mal berechnet werden. Daher haben wir einen Faktor 11/8 als “overhead”, wenn wir in jedem Schritt das Genauigkeitsmaß Δy mitberechnen.

- 2) In den *Numerical Recipes* [2] (Kapitel 17.2) ist folgendes trickreiches **Runge-Kutta-Fehlberg** Verfahren (auch **embedded Runge-Kutta**) angegeben:

Wir berechnen \vec{f} 6mal und generieren daraus

- a ein Runge-Kutta Verfahren 5. Ordnung und
- b ein anderes Runge-Kutta Verfahren 4. Ordnung,

indem wir jeweils andere Koeffizienten wählen. Daraus gewinnen wir dann eine Fehlerabschätzung $\Delta y \equiv |\vec{y}_{RK5} - \vec{y}_{RK4}|$ mit $\Delta y \sim h^5$, also gilt hier $n = 5$ in (4.19). Der overhead ist bei dieser Vorgehensweise fast optimal klein.

Mit der Fehlerabschätzung Δy aus 1) oder 2) können wir dann die Schrittweite anpassen nach folgender Regel:

Wenn $\frac{|\Delta y|}{|\vec{y}|} \left\{ \begin{array}{l} > \\ < \end{array} \right\}$ gegebener relativer Fehler ε , dann $\left\{ \begin{array}{l} \text{reduziere} \\ \text{vergrößere} \end{array} \right\}$ Schrittweite h zu

$$h' = h \left(\frac{\varepsilon |\vec{y}|}{|\Delta y|} \right)^{1/n} \left\{ \begin{array}{l} \text{und wiederhole} \\ \text{für nächsten} \end{array} \right\} \text{Schritt}$$

Noch einige Bemerkungen zu dieser Regel:

- Sicherheitshalber sollte man $|y|$ durch $|h\vec{y}'|$ ersetzen, wenn $|y| \approx 0$.
- Außerdem sollte man sicherheitshalber eine maximale Schrittweite $h < h_{\max}$ nicht überschreiten.
- Man kann noch andere konservativere Regeln für die Wahl von h' verwenden, z.B. in den *Numerical Recipes* (Kapitel 17.2): Wähle bei *Verkleinerung* ein noch kleineres $h' = h \left(\frac{\varepsilon |\vec{y}|}{|\Delta y|} \right)^{1/(n-1)}$ (größerer Exponenten, also stärkere Verkleinerung), um den zusätzlichen Effekt zu kompensieren, dass bei kleinerem h auch mehr Schritte und damit mehr Fehler gemacht werden.

4.5 Integration Newtonscher Bewegungsgleichungen

Wir betrachten nochmal speziell Newtonsche Bewegungsgleichungen. Spezielle Lösungsverfahren sind dort noch der Verlet- und der Leapfrog-Algorithmus.

Wir betrachten jetzt nochmal den in der Physik so wichtigen Fall von Newtonschen Bewegungsgleichungen mit einem Kraftfeld $\vec{F}(\vec{r}, t)$ (das nicht von den Geschwindigkeiten abhängen soll), die wir wieder als DGL 1. Ordnung in Orten und Geschwindigkeiten schreiben:

$$\boxed{\begin{aligned}\dot{\vec{r}} &= \vec{v} \\ \dot{\vec{v}} &= \frac{1}{m} \vec{F}(\vec{r}, t) = \vec{a}(\vec{r}, t)\end{aligned}} \quad (4.20)$$

Wir erinnern nochmal daran, dass für ein N -Teilchensystem in 3 Raumdimensionen die Vektoren \vec{r} und \vec{v} $3N$ -dimensional sind.

Zur numerischen Lösung können natürlich die bereits bekannten Verfahren benutzt werden, also mit **Euler-Verfahren**

$$\boxed{\begin{aligned}\vec{r}_{n+1} &= \vec{r}_n + \vec{v}_n h + \mathcal{O}(h^2) \\ \vec{v}_{n+1} &= \vec{v}_n + \vec{a}_n h + \mathcal{O}(h^2)\end{aligned}} \quad (4.21)$$

oder das **Prädiktor-Korrektor Verfahren**

$$\boxed{\begin{aligned}\vec{r}_{n+1,p} &= \vec{r}_n + \vec{v}_n h + \mathcal{O}(h^2) \\ \vec{a}_{n+1,p} &= \vec{a}(\vec{r}_{n+1,p}, t_{n+1}) \\ \vec{v}_{n+1} &= \vec{v}_n + \frac{1}{2}(\vec{a}_{n+1,p} + \vec{a}_n)h + \mathcal{O}(h^3) \\ \vec{r}_{n+1} &= \vec{r}_n + \frac{1}{2}(\vec{v}_{n+1} + \vec{v}_n)h + \mathcal{O}(h^3)\end{aligned}} \quad (4.22)$$

In der letzten Zeile haben wir verwendet, dass man anstatt des Prädiktors $\vec{v}_{n+1,p}$ auch direkt das bereits berechnete \vec{v}_{n+1} verwenden kann, $\vec{v}_{n+1,p} = \vec{v}_n + \vec{a}_n h = \vec{v}_{n+1} + \mathcal{O}(h^2)$, wegen Übereinstimmung bis $\mathcal{O}(h^2)$. Ebenso kann man die Runge-Kutta-Verfahren speziell auf die DGL (4.20) umschreiben (selbst als Übung...).

Wie bereits am Ende von Kapitel 4.3 erwähnt, ist das Runge-Kutta Verfahren 4. Ordnung – eventuell mit Schrittweitenanpassung – das Verfahren der Wahl für wenige ($N = 1, 2, 3, \dots$) Teilchen. Daneben werden speziell in **MD-Simulationen** mit sehr vielen Teilchen (z.B. $N = 10^2 - 10^6$) oft zwei weitere Verfahren verwendet, der **Verlet-Algorithmus** und der **Leapfrog-Algorithmus**

4.5.1 Verlet-Algorithmen

Der Verlet-Algorithmus (benannt nach Loup Verlet (geb. 1931), ein französischer Physiker, der den Algorithmus in MD-Simulationen popularisiert hat [9]) kann über eine Taylor-Entwicklung bis 2. Ordnung hergeleitet werden. Dazu machen wir eine Taylorentwicklung “rückwärts” und “vorwärts”:

$$\begin{aligned}\vec{r}_{n+1} &= \vec{r}_n + \vec{v}_n h + \frac{1}{2} \vec{a}_n h^2 + \dots \\ \vec{r}_{n-1} &= \vec{r}_n - \vec{v}_n h + \frac{1}{2} \vec{a}_n h^2 + \dots\end{aligned}$$

Addition ergibt den **Verlet-Algorithmus**

$$\boxed{\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n h^2 + \mathcal{O}(h^4)} \quad (4.23)$$

mit einem Fehler $\mathcal{O}(h^4)$, da sich ungerade Terme exakt heben bei der Addition. Die Rekursion (4.23) entspricht einer direkten Diskretisierung der Newtonschen Bewegungsgleichung $\ddot{\vec{r}} = \vec{a}(\vec{r}, t)$ mit der Formel (3.4) für die zweite Zeitableitung. Die Geschwindigkeiten \vec{v}_n kommen nicht vor in (4.23); sie müssen im Verlet-Algorithmus nachträglich berechnet werden mittels der symmetrischen 2-Punkt Formel (3.2),

$$\vec{v}_n = \frac{1}{2h} (\vec{r}_{n+1} - \vec{r}_{n-1}) + \mathcal{O}(h^2) \quad (4.24)$$

Der Verlet-Algorithmus (4.23) und (4.24) hat folgende Eigenschaften:

- Er ist 3. Ordnung in \vec{r} und 1. Ordnung in \vec{v} .
- Dabei benötigt er nur eine Berechnung der Kräfte $m\vec{a}_n$ pro Zeitschritt.
- In der Form (4.23) und (4.24) ist er nicht “selbststartend”, da im Normalfall \vec{r}_0 und \vec{v}_0 als Anfangsbedingungen gegeben sind aber \vec{r}_0 und \vec{r}_{-1} benötigt werden. Man benutzt dann $\vec{r}_{-1} = \vec{r}_0 - \vec{v}_0 h + \frac{1}{2}\vec{a}_0 h^2$ als Startwert.

Äquivalent (und selbststartend) zum Verlet-Algorithmus in der Form (4.23) und (4.24) ist der **Geschwindigkeits-Verlet-Algorithmus**

$$\begin{aligned} \vec{r}_{n+1} &= \vec{r}_n + \vec{v}_n h + \frac{1}{2}\vec{a}_n h^2 \\ &\text{berechne damit } \vec{a}_{n+1} = \vec{a}(\vec{r}_{n+1}, t_{n+1}) \\ \vec{v}_{n+1} &= \vec{v}_n + \frac{1}{2}(\vec{a}_{n+1} + \vec{a}_n)h \end{aligned} \quad (4.25)$$

Die Äquivalenz der beiden Verlet-Algorithmen kann man sich folgendermaßen klarmachen. Nach (4.25) gilt

$$\begin{aligned} \vec{r}_{n+2} &= \vec{r}_{n+1} + \vec{v}_{n+1} h + \frac{1}{2}\vec{a}_{n+1} h^2 \\ \vec{r}_n &= \vec{r}_{n+1} - \vec{v}_n h - \frac{1}{2}\vec{a}_n h^2 \end{aligned}$$

Addition ergibt

$$\begin{aligned} \vec{r}_{n+2} + \vec{r}_n &= 2\vec{r}_{n+1} + (\vec{v}_{n+1} - \vec{v}_n)h + \frac{1}{2}(\vec{a}_{n+1} - \vec{a}_n)h^2 \\ &\stackrel{(4.25)}{=} 2\vec{r}_{n+1} + \vec{a}_{n+1} h^2 \end{aligned}$$

also wieder (4.23).

4.5.2 Leapfrog-Algorithmus

Ebenfalls äquivalent zu den Verlet-Algorithmen ist der **Leapfrog** (Bocksprung) - Algorithmus, der seinen Namen deshalb erhielt, weil \vec{r} und \vec{v} abwechselnd in Halbschritten upgedatet werden (wie beim Bockspringen):

$$\begin{aligned} \vec{v}_{n+1/2} &= \vec{v}_{n-1/2} + \vec{a}_n h \\ \vec{r}_{n+1} &= \vec{r}_n + \vec{v}_{n+1/2} h \\ &\text{berechne damit } \vec{a}_{n+1} = \vec{a}(\vec{r}_{n+1}, t_{n+1}) \end{aligned} \quad (4.26)$$

Dann kann wieder $\vec{a}_{n+1} = \vec{a}(\vec{r}_{n+1}, t_{n+1})$ berechnet werden und damit wiederum $\vec{v}_{n+3/2}$ usw.

Man zeigt wieder leicht, dass der Leapfrog-Algorithmus die gleichen Trajektorien \vec{r}_n wie die Verlet-Algorithmen erzeugt. Ein Nachteil dabei ist, dass die Geschwindigkeiten $\vec{v}_{n+1/2}$ immer nur auf halbzahligen Schritten berechnet werden. Daher können kinetische und potentielle Energie in diesem Verfahren nicht zu gleichen Zeiten berechnet werden.

In MD-Simulationen wird typischerweise ein Verlet-Algorithmus mit fester kleiner Schrittweite benutzt (weil die Berechnung von \vec{a}_n sehr aufwendig wird, da $\mathcal{O}(N^2)$ Wechselwirkungs Kräfte zwischen den Teilchen zu berechnen sind).

4.6 Implizite Verfahren und steife DGL-Systeme

Bei impliziten Verfahren wird im numerischen Integrationsschritt nicht explizit nach dem neuen Funktionswert aufgelöst. In steifen DGL-Systemen haben verschiedene Freiheitsgrade sehr unterschiedliche Zeitskalen. Dies führt zu Stabilitätsproblemen, die mit impliziten Verfahren behandelt werden können.

In der Physik hat man es häufiger mit DGL-Systemen zu tun, die Vorgänge mit sehr unterschiedlichen Zeit- oder Längenskalen beschreiben. Man kann auch dann durchaus die bisher vorgestellten Euler- und Runge-Kutta Verfahren verwenden, muss allerdings eine der kleinsten Zeit- oder Längenskala angepasste Schrittweite in Kauf nehmen. Dies ist anschaulich klar, wir werden unten auch genauer einsehen, dass dies letztendlich Stabilitätsgründe hat.

Wenn diese Möglichkeit unpraktikabel bleibt, kann man versuchen etwas mehr Aufwand zu betreiben und anstatt der bisher vorgestellten *expliziten* sogenannte *implizite* Verfahren zu verwenden. Implizite Verfahren werden im Folgenden ansatzweise eingeführt und ihre Anwendung an Hand eines einfachen linearen Systems erläutert.

4.6.1 Implizite Verfahren

Euler- und Runge-Kutta Verfahren sind **explizite Verfahren**, wo die Rekursion für jeden Zeitschritt explizit nach \vec{y}_{n+1} aufgelöst ist. Bei einem **impliziten Verfahren** kann man dagegen nicht explizit nach \vec{y}_{n+1} auflösen, um eine “echte” Rekursion $\vec{y}_{n+1} = \vec{y}_{n+1}(\vec{y}_n, \vec{y}_{n-1}, \dots)$ zu bekommen. Z.B. kann man in der Integraldarstellung (4.9) eines Integrationsschrittes die Trapezregel verwenden und bekommt wie in (4.10) zunächst

$$\vec{y}_{n+1} = \vec{y}_n + \frac{h}{2} [\vec{f}(t_n, \vec{y}_n) + \vec{f}(t_{n+1}, \vec{y}_{n+1})] + \mathcal{O}(h^3) \quad (4.27)$$

Dies stellt erst einmal eine **implizite Gleichung** für \vec{y}_{n+1} dar, da \vec{y}_{n+1} auf beiden Seiten von (4.27) vorkommt.

Wir haben mit dem Prädiktor-Korrektor Verfahren (4.11) und (4.12) eine Möglichkeit kennengelernt das Problem approximativ zu lösen, indem wir mit Hilfe eines Euler-Verfahrens erst einmal ein genähertes $\vec{y}_{n+1,p}$ *explizit* “vorhergesagt” haben, das wir dann auf der rechten Seite der *impliziten* Gleichung benutzen, um daraus eine explizite zu machen. Prädiktor-Korrektor Verfahren verbinden also einen expliziten Prädiktor mit einem impliziten Korrektor.

Bei echten **impliziten Verfahren** versucht man dagegen, die implizite Gleichung (4.27) in jedem Zeitschritt *direkt* numerisch zu lösen. Zwei Fälle können unterschieden werden:

- (i) Wenn \vec{f} **linear** ist, $\vec{f}(t, \vec{y}) = \underline{\underline{A}}(t) \cdot \vec{y}$, ist dies relativ einfach und involviert nur Matrixinversionen in jedem Zeitschritt. Eventuell kann man sogar analytisch auflösen und das Verfahren damit wieder explizit machen. Solche Beispiele linearer DGLn werden wir im nächsten Abschnitt genauer betrachten.

- (ii) Wenn \vec{f} nicht-linear ist, ist nur eine *numerische* Lösung möglich, z.B. mit Newton-Raphson Verfahren, die wir später besprechen werden.

Sowohl das Prädiktor-Korrektor Verfahren als auch implizite Verfahren haben oft bessere Stabilitätseigenschaften. Dies macht man sich insbesondere bei sogenannten steifen DGL-Systemen zu Nutze.

4.6.2 Steife DGL-Systeme

In steifen DGL-Systemen haben verschiedene Freiheitsgrade sehr unterschiedliche Zeitskalen. Dieses Problem taucht in der Physik häufig auf. Ein Beispiel wäre eine lineare Kette aus mit Federn gekoppelten Massen, in der eine Feder viel härter ist und daher viel schneller reagiert. Dann muss sich die mögliche Schrittweite bei der numerischen Lösung mit den bekannten *expliziten* Verfahren immer nach der einzelnen harten Feder richten und typischerweise viel kleiner sein als bei homogen weichen Federn. Um dies zu demonstrieren, betrachten wir der Einfachheit halber ein System aus nur 2 gekoppelten Massen mit einer überdämpften Dynamik, siehe Abb. 4.4:

$$\begin{aligned} y'_1 &= -k_1(y_1 - y_2) \\ y'_2 &= -k_2y_2 + k_1(y_1 - y_2) \end{aligned}$$

oder

$$\vec{y}' = \underline{\underline{A}}\vec{y} \text{ mit } \underline{\underline{A}} = \begin{pmatrix} -k_1 & k_1 \\ k_1 & -k_1 - k_2 \end{pmatrix}.$$

Für eine sehr harte Feder $k_2 \gg k_1$ gilt für die beiden Eigenwerte der Matrix $\underline{\underline{A}}$ in führender Ordnung in k_1 : $\lambda_2 \approx -k_2 - 2k_1$ und $\lambda_1 \approx -k_1$, also $|\lambda_2| \gg |\lambda_1|$. Dann muss sich die Schrittweite h in einem expliziten Verfahren immer an der Zeitskala $1/|\lambda_2| \approx 1/k_2$ orientieren, die von der härteren Feder k_2 bestimmt wird.

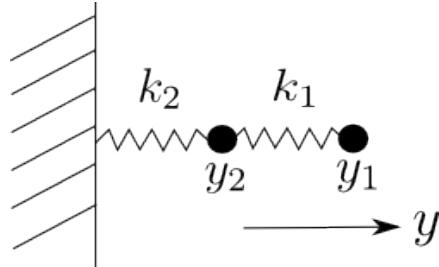


Abbildung 4.4: System aus 2 Massen mit Federn k_1 und k_2 .

Um dies einzusehen betrachten wir zunächst noch einmal das einfache Beispiel $y' = -ay$ mit $a > 0$ und der analytischen Lösung $y(t) = y(0)e^{-at}$ und untersuchen das zugehörige normal explizite oder **Vorwärts-Euler-Verfahren** $y_{n+1} = y_n - ah y_n = y_n(1 - ha)$ ähnlich wie in Kapitel 2.3 auf Stabilität. Die Euler-Rekursion wird offensichtlich durch

$$y_n = (1 - ha)^n y_0$$

gelöst und geht stabil exponentiell gegen 0 für alle $h < 2/a$, für die $|1 - ha| < 1$ gilt. Wenn jedoch $h > 2/a$ oszilliert die Lösung und wächst exponentiell an: das Verfahren wird *instabil*. Genau das passiert in der Regel, wenn in einer entsprechenden mehrdimensionalen DGL $\vec{y}' = -\underline{\underline{A}}\vec{y}$ für *einen* negativen Eigenwert λ_i der Matrix $\underline{\underline{A}}$ die Stabilitätsbedingung

$$h < 2/|\lambda_i|$$

verletzt ist. Dies wird zuerst für den betragsmäßig größten negativen Eigenwert, also den Teil der Dynamik mit der kleinsten Zeitskala (also oben die härteste Feder) passieren.

Man kann dieses Problem umgehen, indem man im einfachen Beispiel auf ein ganz einfaches **implizites Verfahren** zurückgreift, nämlich das **Rückwärts-Euler-Verfahren**

$$\vec{y}_{n+1} = \vec{y}_n + h\vec{f}(t_{n+1}, \vec{y}_{n+1}) + \mathcal{O}(h^2) \quad (4.28)$$

wo die Ableitung auf der rechten Seite am Intervallende ausgewertet wird. Für unser einfaches Beispiel $y' = -ay$ mit $a > 0$ kann man das Verfahren natürlich auch wieder explizit machen:

$$\begin{aligned} y_{n+1} &= y_n - hay_{n+1} \\ y_{n+1} &= \frac{1}{1 + ha} y_n \end{aligned}$$

mit der Lösung

$$y_n = \frac{1}{(1 + ha)^n} y_0$$

Hier gibt es für $a > 0$ nun für *beliebiges* h kein Stabilitätsproblem und die Lösung geht *immer* gegen 0.

Ähnlich kann man auch für die entsprechende mehrdimensionalen DGL $\vec{y}' = -\underline{\underline{A}} \cdot \vec{y}$ vorgehen

$$\vec{y}_{n+1} = \vec{y}_n - h\underline{\underline{A}} \cdot \vec{y}_{n+1} \quad (4.29)$$

$$\vec{y}_{n+1} = (\underline{\underline{1}} + h\underline{\underline{A}})^{-1} \cdot \vec{y}_n \quad (4.30)$$

Dieses Verfahren wird sich bzgl. der negativen Eigenwerte und der zugehörigen Eigenmoden absolut stabil verhalten. Allerdings muss nun im Normalfall, dass man die inverse Matrix in (4.30) nicht einfach analytisch geschlossen angeben kann, das Inverse numerisch bestimmt werden. Wenn $\underline{\underline{A}}_n = \underline{\underline{A}}(t_n)$ zeitabhängig ist, muss dieses Inverse sogar in *jedem* Zeitschritt bestimmt werden.

4.7 Weitere Verfahren

Wir stellen hier als alternative Verfahren Prädiktor-Korrektor Verfahren höherer Ordnung und Bulirsch-Stoer Verfahren kurz vor.

Für praktische Zwecke reichen in der Physik im Normalfall Euler-, Runge-Kutta und Verlet-Verfahren. Daneben gibt es aber noch andere Verfahren, die alle ihre Stärken und Schwächen haben. Zwei davon sollen hier kurz beschrieben werden.

4.7.1 Prädiktor-Korrektor Verfahren höherer Ordnung

Es gibt noch aufwendigere **Prädiktor-Korrektor Verfahren** höherer Ordnung als die einfache Version (4.11) und (4.12) 2. Ordnung, siehe auch *Numerical Recipes* [2], Kapitel 17.6. Die Grundidee dabei ist, dass man \vec{y}_{n+1} durch Extrapolation aus der “Vergangenheit” gewinnt. Dabei werden bereits berechnete Funktionswerte \vec{y}_i mit $i \leq n$ und entsprechende Ableitungen $\vec{y}'_i = \vec{f}(t_i, \vec{y}_i)$ verwendet, wobei auch Werte $i < n$ bei aufwendigeren Prädiktor-Korrektor Verfahren zum Einsatz kommen. Daher sind diese Verfahren **Mehrschrittverfahren** im Gegensatz zu Euler- und Runge-Kutta Verfahren. Ein potentieller Vorteil solcher Mehrschrittverfahren ist, dass die aufwendigere Berechnung auch genauer ist und größere Schrittweiten ermöglicht. Bei “glatten” Funktionen \vec{f} auf der rechten Seite der DGL funktioniert dies auch gut, bei schnell veränderlichen rechten Seiten \vec{f}

ist der Vorteil nicht vorhanden. Ein Nachteil ist, dass diese aufwendigeren Verfahren auch nicht so einfach zu implementieren sind.

Die Idee bei Prädiktor-Korrektor Verfahren ist, den Integranden in der Integraldarstellung (4.9) eines Integrationsschrittes

$$\vec{y}_{n+1} = \vec{y}_n + \int_{t_n}^{t_{n+1}} dt \vec{f}(t, \vec{y}(t))$$

durch ein interpolierendes Polynom zu nähern, das durch die Werte $\vec{y}'_i = \vec{f}(t_i, \vec{y}_i)$ an den bereits berechneten Funktionswerten \vec{y}_i zu Zeiten t_i mit $i \leq n$ geht. Dies ergibt dann den **Prädiktor** $\vec{y}_{n+1,p}$, der auf den Ableitungen \vec{y}'_i mit $i \leq n$ beruht:

$$\vec{y}_{n+1,p} = \vec{y}_n + h(\beta_1 \vec{y}'_n + \beta_2 \vec{y}'_{n-1} + \dots)$$

mit Koeffizienten β_1, β_2 usw. die aus der Interpolation folgen. Ist der Prädiktor $\vec{y}_{n+1,p}$ bekannt, kann auch ein entsprechender Wert der Ableitung $\vec{y}'_{n+1,p} = \vec{f}(t_{n+1}, \vec{y}_{n+1,p})$ zur Zeit t_{n+1} in die Interpolation einbezogen werden. Dies ergibt dann den **Korrektor** \vec{y}_{n+1} , der auf den Ableitungen \vec{y}'_i mit $i \leq n$ beruht:

$$\vec{y}_{n+1} = \vec{y}_n + h(\tilde{\beta}_0 \vec{y}'_{n+1,p} + \tilde{\beta}_1 \vec{y}'_n + \tilde{\beta}_2 \vec{y}'_{n-1} + \dots)$$

mit Koeffizienten $\tilde{\beta}_0, \tilde{\beta}_1, \tilde{\beta}_2$ usw. die wieder aus der Interpolation folgen mit einem zusätzlichen Koeffizienten $\tilde{\beta}_0$, da nun auch $\vec{y}'_{n+1,p}$ zur Zeit t_{n+1} zur Interpolation verwendet wird. Konkrete Werte für die β_i und $\tilde{\beta}_i$ in verschiedenen Verfahren können in [2] (Kapitel 17.6) oder [2] (Kapitel 12) gefunden werden.

Wir geben nur ein Beispiel, das jeweils mit Polynomen 2-ten Grades arbeitet, das **Adams-Bashforth-Moulton Verfahren 3. Ordnung**:

Prädiktor: $\vec{y}_{n+1,p} = \vec{y}_n + \frac{h}{12}(23\vec{y}'_n - 16\vec{y}'_{n-1} + 5\vec{y}'_{n-2}) + \mathcal{O}(h^4)$	(4.31)
Korrektor: $\vec{y}_{n+1} = \vec{y}_n + \frac{h}{12}(5\vec{y}'_{n+1,p} + 8\vec{y}'_n - \vec{y}'_{n-1}) + \mathcal{O}(h^4)$	

wobei $\vec{y}'_{n+1,p} = \vec{f}(t_{n+1}, \vec{y}_{n+1,p})$ in der Korrektor-Formel mit dem Prädiktor berechnet wird.

Ein grundsätzlicher Vorteil von Prädiktor-Korrektor Verfahren ist, dass die Differenz zwischen Prädiktor und Korrektor sofort eine Fehlerabschätzung liefert, die wiederum zur adaptiven Schrittweitenanpassung genutzt werden könnte.

4.7.2 Bulirsch-Stoer Verfahren

Bei Bulirsch-Stoer Verfahren besteht die Idee darin, einen Schritt H mit Hilfe von mehreren Runge-Kutta Schritten mit verschiedenen kleineren Schrittweiten $h = H, H/2, H/4, \dots, H/2^n, \dots$ zu berechnen und dann das Ergebnis nach $h = 0$ bzw. $n \rightarrow \infty$ zu interpolieren. Dies ist eine ähnliche Idee wie bei der Romberg-Integration.

Bulirsch-Stoer Verfahren sind sehr genau, allerdings muss \vec{f} offensichtlich sehr oft berechnet werden. Dies wird aber oft dadurch aufgewogen, dass sehr große Schrittweiten H erreicht werden z.B. im Vergleich zu Runge-Kutta Verfahren. Details zu diesem Verfahren können in den *Numerical Recipes* [2] (Kapitel 17.3) gefunden werden.

4.7.3 Programmpakete/Solver

Gerade für gewöhnliche DGLn gibt es viele gute Solver-Routinen, z.B. in den *Numerical Recipes* [1], [2] oder in den NAG-Routinen. In diesen sind die oben vorgestellten Verfahren schon fertig

implementiert. Es lohnt sich durchaus, sich damit vertraut zu machen, wie solche Routinen in C, C++ oder Fortran-Programme eingebunden werden.

Auch in MD-Paketen wie Gromacs sind Solver für die Newtonschen Bewegungsgleichungen oft schon implementiert.

Daneben gibt es auch in Mathematica, Maple oder Matlab, die eigene Skriptsprachen verwenden, gute numerische Solver für gewöhnliche DGLn.

Daher greift man in der Praxis immer seltener auf selbst geschriebene Implementationen zurück.

4.8 Literaturverzeichnis Kapitel 4

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis*. 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.
- [4] R. W. Hamming. *Numerical Methods for Scientists and Engineers*. 2nd. New York, NY, USA: Dover Publications, Inc., 1986.
- [5] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.
- [6] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [7] W. Kinzel und G. Reents. *Physics by Computer*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
- [8] D. Frenkel und B. Smit. *Understanding Molecular Simulation*. 2nd. Orlando, FL, USA: Academic Press, Inc., 2001.
- [9] L. Verlet. *Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*. Phys. Rev. **159** (1967), 98–103.

4.9 Übungen Kapitel 4

1. Runge-Kutta 4. Ordnung

Schreiben Sie ein Programm, dass die Newtonsche Bewegungsgleichung für ein Teilchen in einem Kraftfeld $\vec{F}(\vec{r})$,

$$\begin{aligned}\dot{\vec{r}} &= \vec{v} \\ \dot{\vec{v}} &= \frac{1}{m} \vec{F}(\vec{r}),\end{aligned}\tag{4.32}$$

mit Hilfe des Runge-Kutta-Verfahrens 4. Ordnung mit fester Schrittweite löst. Das Programm sollte ein Unterprogramm enthalten, in dem Sie das Kraftfeld $\vec{F}(\vec{r})$ angeben können. Es folgen Aufgaben 2 und 3 mit 2 verschiedenen Kraftfeldern als Anwendung.

Sie sollen das Programm zumindest für 3 Raumdimensionen ($\vec{r}, \vec{v}, \vec{F} \in \mathbb{R}^3$) schreiben, können es aber auch so schreiben, dass Sie allgemein in D Raumdimensionen arbeiten können.

2. Harmonischer Oszillator

Sie sollten Programme immer erst an einfachen Problemen testen, deren Lösung Sie kennen, bevor Sie ein kompliziertes Problem angehen. Unser einfaches Testproblem für das Programm aus Aufgabe 1 ist der harmonische Oszillator mit

$$\frac{1}{m} \vec{F}(\vec{r}) = -\vec{r}\tag{4.33}$$

a) Verifizieren Sie für Anfangsbedingungen $\vec{r}(0)$ beliebig, $\vec{v}(0) = 0$, dass Sie eine harmonische Schwingung erhalten. Was passiert für $\vec{v}(0) \neq 0$ und $\vec{v}(0) \parallel \vec{r}(0)$?

b) Testen Sie, wie klein Sie die Schrittweite machen müssen, damit ihr Oszillator bei mehreren Oszillationen immer wieder seine maximale Anfangsauslenkung erreicht. Testen Sie die Energieerhaltung.

3. Kepler-Ellipsen

Das kompliziertere Problem, das wir nun in 3 Raumdimensionen behandeln wollen, ist das Kepler-Problem mit $V(r) = -mG/r$ oder

$$\frac{1}{m} \vec{F}(\vec{r}) = -G \frac{\vec{r}}{r^3}\tag{4.34}$$

mit einer Konstanten G , wobei Sie erst einmal $G = 1$ setzen.

a) Berechnen Sie numerisch die Bahn des Teilchens für $\vec{r}(0) = (1, 0, 0)$. Finden Sie eine Anfangsgeschwindigkeit, so dass das Teilchen eine Ellipse beschreibt. Wie klein müssen Sie die Schrittweite wählen, damit sich die Ellipse auch wirklich schließt? Welches Problem bekommen Sie bei sehr kleinen Anfangsgeschwindigkeiten?

b) Überprüfen Sie numerisch Energieerhaltung und das 2. Keplergesetz (Drehimpulserhaltung). Freiwillige Zusatzaufgabe: Überprüfung auch des 3. Keplerschen Gesetzes

c) Testen Sie, ob der Lenz-Runge-Vektor $\vec{\Lambda} = \frac{1}{Gm} \vec{p} \times \vec{L} - \vec{r}/r$ in ihrer Numerik erhalten ist. Auf welchen Punkt der Bahn zeigt der Lenz-Runge-Vektor?

d) Testen Sie die Fehler und Stabilität Ihrer Integration, indem Sie N Runge-Kutta Schritte machen, dann die Zeit umkehren, d.h., den Geschwindigkeitsvektor des Teilchens umkehren $\vec{v}_u(t = 0) \equiv -\vec{v}(t = Nh)$ und wieder für N Runge-Kutta Schritte integrieren. Prüfen Sie nach, ob Sie wieder am Ausgangspunkt $\vec{r}_u(t = Nh) = \vec{r}(t = 0)$ mit umgekehrter Ausgangsgeschwindigkeit $\vec{v}_u(t = Nh) = -\vec{v}(t = 0)$ landen. Versuchen Sie, N möglichst groß zu wählen.

- e) Was passiert mit ihrer Ellipsenbahn, wenn Sie das Potential abändern zu $V(r) = -mG/r^\alpha$, wobei $\alpha \neq 1$? Betrachten Sie z.B. numerische Lösungen zu $\alpha = 0.9$ und $\alpha = 1.1$. Ist der Lenz-Runge-Vektor noch erhalten?

4. Lineares Pendel

Lösen Sie numerisch die Bewegungsgleichung eines **linearen**, gedämpften getriebenen Pendels

$$\ddot{\theta} = -\frac{\dot{\theta}}{Q} - \theta + A \cos \omega t \quad (4.35)$$

mit Hilfe des Runge-Kutta-Verfahrens 4. Ordnung mit fester Schrittweite h .

- a) Energieerhaltung:

Betrachten Sie zunächst den Fall $A = 0$ und $Q \rightarrow \infty$. Wie lautet die erhaltene Energie in diesem Grenzfall? Wie klein müssen Sie das Verhältnis h zur Schwingungsdauer T wählen, damit Sie numerische Energieerhaltung finden?

- b) PhasenraumporTRAITS:

Fertigen Sie numerisch ein PhasenraumporTRAIT an, indem Sie die Bewegung in der $\theta/\pi-\dot{\theta}$ -Ebene plotten, d.h. Punkte $(\theta(t)/\pi, \dot{\theta}(t))$ für viele Zeiten t . Benutzen Sie dabei Anfangsbedingungen $\theta(0) = 0$ und $\dot{\theta}(0) = 0$ und plotten Sie mit “ 2π -periodischen Randbedingungen” für θ , so dass θ immer im Intervall $[-\pi, \pi[$ liegt. Betrachten Sie Parameter $A = 1.5$, $\omega = 2/3$ und $Q = 0.25$, $Q = 0.5$ und $Q = 1$. Erklären Sie die Ergebnisse an Hand der bekannten analytischen Lösung des Problems im Limes großer Zeiten.

5. Nichtlineares Pendel

Lösen Sie nun mit dem gleichen Algorithmus die Bewegungsgleichung eines **nichtlinearen**, gedämpften getriebenen Pendels

$$\ddot{\theta} = -\frac{\dot{\theta}}{Q} - \sin \theta + A \cos \omega t. \quad (4.36)$$

- a) Energieerhaltung:

Testen Sie wieder numerisch die Energieerhaltung im Grenzfall $A = 0$ und $Q \rightarrow \infty$.

- b) PhasenraumporTRAITS:

Fertigen Sie wieder numerisch wie in Aufgabe 4 PhasenraumporTRAITS an für $A = 1.5$, $\omega = 2/3$ und $Q = 0.5$, $Q = 1$, $Q = 1.2$, $Q = 1.3$ und $Q = 1.4$. Als Anfangsbedingungen können Sie wieder $\theta(0) = 0$ und $\dot{\theta}(0) = 0$ wählen. Vergleichen Sie Ihre Portraits mit den entsprechenden für den linearen Fall.

6. Poincaré-Schnitte

Beim Poincaré-Schnitt für das Pendel nehmen wir den Punkt im Phasenraum nur *einmal* pro Periode auf, also beispielsweise zu Zeiten

$$t_n = nT = 2\pi n/\omega, \quad n = 0, 1, 2, \dots \quad (4.37)$$

Der Poincaré-Schnitt besteht also aus diskreten Punkten $(\theta(t_n)/\pi, \dot{\theta}(t_n))$ in der $\theta/\pi-\dot{\theta}$ -Ebene.

- a) Verifizieren Sie numerisch, dass bei einer streng periodischen Bewegung beim *linearen* Pendel der Poincaré-Schnitt aus nur einem Punkt besteht. Dabei sollte eine Einschwingphase von einigen n abgewartet werden.

- b) Fertigen Sie für das *nichtlineare* Pendel Poincaré-Schnitte an für $A = 1.5$ und $\omega = 2/3$ als Funktion von Q für $Q > 1/2$. Plotten Sie dazu jeweils den Wert $\dot{\theta}(t_n)$ für große n als Funktion von Q im Bereich $1/2 < Q < 1.4$. Verwenden Sie Anfangsbedingungen $\theta(0) = 0$ und verschiedene Werte

für $\dot{\theta}(0)$, z.B. $\dot{\theta}(0) = 0, -3$ und eventuell noch andere Werte im Intervall $[-5, 5]$. Was beobachten Sie speziell im Bereich $1.2 < Q < 1.4$, wenn Sie mit verschiedenen Anfangsbedingungen für $\dot{\theta}(0)$ starten?

7. Schwingende Saite

Wir lösen die Bewegungsgleichungen für eine diskrete transversal schwingende Saite aus 5 identischen Massen m mit Koordinaten $y_i(t)$ ($i = 1, \dots, 5$), die über Federn mit Federkonstante k gekoppelt sind. Die Randbedingungen sollen zwei festen Massen mit $y_0 = y_6 = 0$ entsprechen. Die Bewegungsgleichungen lauten dann (siehe Physik3)

$$\begin{aligned} m\ddot{y}_i &= \frac{\sigma}{a}(y_{i+1} - 2y_i + y_{i-1}) \quad (i = 2, 3, 4) \\ m\ddot{y}_1 &= \frac{\sigma}{a}(y_2 - 2y_1) \\ m\ddot{y}_5 &= \frac{\sigma}{a}(y_4 - 2y_5), \end{aligned} \tag{4.38}$$

wobei σ die Spannung der Saite und a der Abstand der Massen ist. Die Anfangsbedingungen bei $t = 0$ können Sie frei wählen, z.B. können Sie die Saite in eine Dreiecksconfiguration bringen mit $y_1(0) = 1 = y_5(0)$, $y_2(0) = 2 = y_4(0)$ und $y_3(0) = 3$.

- a)** Führen Sie eine Zeiteinheit τ ein, so dass die Bewegungsgleichungen in der reskalierten Zeit $\tilde{t} = t/\tau$ dimensionslos werden (also $\sigma/a = 1$, $m = 1$ entsprechen). Wie hängt τ mit der Wellengeschwindigkeit zusammen?
- b)** Lösen Sie die Bewegungsgleichungen des dimensionslosen Systems numerisch mit Hilfe des Verlet-Algorithmus mit fester Schrittweite h .
- c)** Messen Sie Gesamtenergie, kinetische Energie $T = \frac{1}{2} \sum_{i=1}^5 \dot{y}_i^2$ und Federenergie $U = \frac{1}{2} \sum_{i=1}^6 (y_i - y_{i-1})^2$ (wobei $y_0 = y_6 = 0$). Wie klein muss h sein, damit die Energieerhaltung gut erfüllt ist. Überzeugen Sie sich, dass kinetische und Federenergie für lange Zeiten keine stationären Werte annehmen, sondern immer oszillieren.

5 Molekulardynamik (MD) Simulation

Literatur zu diesem Teil:

Eine der beiden wichtigen Simulationsmethoden für klassische Vielteilchensysteme, zu der es entsprechend viel Literatur gibt. Sehr zu empfehlen ist Frenkel [1], an dem sich auch dieses Kapitel orientiert, aber auch Gould/Tobochnik [2] oder Thijssen [3].

5.1 Grundsätzliches

Wir diskutieren die Idee von MD-Simulationen, nämlich die mikroskopische Integration aller Newtonschen Bewegungsgleichungen mit anschließender Mittelung von Observablen, und die wesentlichen Elemente einer MD-Simulation.

Mit Hilfe von **Molekular-Dynamik-Simulationen** werden Vielteilchensysteme (Teilchenzahl $N \gg 1$) der statistischen Physik simuliert. Die erste MD-Simulation wurde von Alder und Wainwright [4] an einem zwei-dimensionalen System harter Scheiben durchgeführt.



Abbildung 5.1: Die Begründer der Molekulardynamik-Simulation. Links: Bernie Alder (geb. 1925), amerikanischer Physiker. Rechts: Thomas Wainwright (1927-2007), amerikanischer Physiker.

Die grundsätzliche Idee dabei ist

- (i) die **numerische Lösung der mikroskopischen Newtonschen Bewegungsgleichungen** eines Vielteilchensystems mit $N \gg 1$, um dann
- (ii) thermodynamische Mittelwerte von physikalischen Observablen durch **Zeitmittelung** zu gewinnen. Diese sollten nach Ergodenhypothese dann auch den **Ensemble-Mittelwerten** der statistischen Physik entsprechen.

Dabei hat man in einfachsten Version der MD ein abgeschlossenes autonomes System mit Energie- und Teilchenzahlerhaltung und festem Volumen. dann sind E , N und V fest und man arbeitet im **mikrokanonischen Ensemble**. Wir werden in der Hauptsache solche einfachen mikrokanonischen MD Simulationen diskutieren und nur in Kapitel 5.5 darauf eingehen, wie wir mit Hilfe von **Thermostaten** auch **kanonische Ensembles** in MD Simulationen realisieren können.

Im einfachsten Fall hat man N identische Teilchen (Masse m), die über paarweise Zentralkräfte miteinander wechselwirken:

$$\vec{F}_{ij} = \text{Kraft auf Teilchen } i \text{ durch WW. mit Teilchen } j = -\vec{\nabla}_{\vec{r}_i} V(|\vec{r}_i - \vec{r}_j|).$$

Dann ist das System autonom und die Gesamtenergie

$$E = \sum_{i=1}^N \frac{\vec{p}_i^2}{2m} + \sum_{i < j} V(\underbrace{|\vec{r}_i - \vec{r}_j|}_{\equiv \vec{r}_{ij}})$$

erhalten. Außerdem wird man über **Randbedingungen** ein festes Volumen V vorgeben.

Die **Ergodenhypothese** besagt, dass solche Vielteilchensysteme in der Regel (d.h. bei hinreichend chaotischer Dynamik) auf Grund der Wechselwirkungen oder "Stöße" zwischen den Teilchen "mischend" sein sollten. D.h. man kann annehmen, dass alle Zustände auf der Energiehyperfläche gleich oft besucht werden. Dann gilt **Zeitmittel = mikrokanonisches Ensemblemittel**. In der MD-Simulation führt man also ähnlich wie im Experiment letztendlich wieder ein mikroskopisches Zeitmittel durch und rechnet gar nicht in der Ensembletheorie, wie man es in der statistischen Physik gelernt hat.

Eine MD-Simulation enthält bereits so viele mikroskopische Information, das man man auch von einem **Computerexperiment** sprechen kann. Die Monte-Carlo Simulation wird später eine andere Methode bereit stellen um auf dem gleichen Level von mikroskopischer Information eine Simulation oder Computerexperiment durchzuführen. Die grundsätzlichen **Elemente einer MD-Simulation** sind folgende:

1) Definition des Modellsystems:

Dies beinhaltet die Definition der Kräfte, der Teilchenzahl N und des Simulationsvolumens V und insbesondere auch der Randbedingungen. Auf der Programmseite heißt das, man muss passende Datenstrukturen für Teilchenpositionen, -geschwindigkeiten und Kräfte definieren.

2) Initialisierung:

Die Anfangspositionen- und impulse der Teilchen müssen geeignet festgelegt werden.

3) Äquilibrierung:

Jede Simulation eines Vielteilchensystems muss sich in der Regel "warmlaufen", bis ein stationärer Zustand durch genügend viele Wechselwirkungen oder Stöße der Teilchen erreicht ist.

4) Messung:

Dies ist der wichtigste und auch längste Teil der Simulation, da hier die interessierenden Messgrößen letztendlich bestimmt werden. Wir messen in der Regel durch **Zeitmittelung** von statischen Observablen $O = O(\{\vec{r}\}, \{\vec{p}\})$, die als Funktionen der Orte und Impulse aller Teilchen ausdrückbar sein müssen. Neben der Zeitmittelung mittelt man oft auch über alle N Teilchen oder alle N_f Freiheitsgrade der Teilchen. Extensive Größen sind normalerweise ohnehin als Summe über alle Teilchen definiert, aber auch bei intensiven Größen lässt sich solch eine Mittelung oft durchführen.

So lässt sich z.B. die **Temperatur** mit Hilfe des Äquipartitionstheorems an jedem beliebigen Teilchen i messen:

$$\left\langle \frac{p_{i,\alpha}^2}{2m} \right\rangle = \frac{1}{2} k_B T \quad (5.1)$$

(wo α die Komponenten des Impulses indiziert). Da wir auch nicht nur über die Zeit, sondern auch über alle Freiheitsgrade (Teilchen) mitteln können, kann man damit auch eine **momen-**

momentane Temperatur zu einem beliebigen Zeitpunkt t messen

$$T(t) = \frac{2}{k_B N_f} \sum_{i=1}^N \frac{\vec{p}_i^2}{2m}, \quad (5.2)$$

wo $N_f = 3N - 3$ die Zahl der Freiheitsgrade des Systems ist; dies sind 3 Raumdimensionen pro Teilchen, wobei 3 Freiheitsgrade des (erhaltenen) Schwerpunktsimpuls abgezogen werden, den wir auf 0 setzen werden (siehe unten). Die in (5.2) definierte momentane Temperatur zeigt relative Fluktuationen von der Größenordnung $\sim 1/\sqrt{N_f}$, die dann erst durch eine zusätzliche Zeitmittelung unter einer gewünschten Schwankung gedrückt werden können.

5) Integration:

Sowohl während der Äquilibrierung 3) als auch während der eigentlichen MD-Simulation zur Messung 4) werden permanent die Newtonschen Bewegungsgleichungen in einer Zeitschleife gelöst. Hier werden wir den Verlet-Algorithmus verwenden.

Der grundsätzliche **Aufbau einer MD-Simulation** ist dann folgender:

```

init                      → Initialisierung [5.2]
t=0
do while (t<tequi)      → Äquilibrierung [5.4]
    forces
    integrate
    t=t+dt
enddo
do while (t<tmax)      → Zeitschleife
    forces          → Kraftberechnung [5.2]
    integrate       → Integration der Bewegungsgleichungen [5.3]
    t=t+dt
    measure         → Messung, Mittelung [5.4]
enddo

```

Die verschiedenen Teile werden in den folgenden Kapiteln im Detail behandelt.

5.2 Kräfte, Randbedingungen, Initialisierung

Wir besprechen verschiedene Arten von Wechselwirkungskräften, insbesondere wird die Lennard-Jones Wechselwirkung eingeführt. Die möglichen Komplikationen im Zusammenhang mit periodischen Randbedingungen und der Initialisierung des Systems werden diskutiert.

5.2.1 Kräfte

Die Physik des Systems steckt natürlich wesentlich in den Wechselwirkungskräften der Teilchen (Atome/Moleküle). Unser Thema wird die statistische Mechanik klassischer Teilchen sein, d.h. wir werden keine Quanteneffekte berechnen (metallische Bandstruktur, Bosekondensation, Supraleitung, ...), sondern klassische thermodynamische Phänomene wie Phasenübergänge zwischen den Aggregatzuständen fest, flüssig und gasförmig betrachten. Die Kräfte werden im Folgenden klassisch mechanisch behandelt, obwohl der Ursprung der Wechselwirkungskräfte immer Elektrodynamik *und* Quantenmechanik ist.

Im Folgenden werden die wichtigsten Arten von Kräften noch einmal kurz abgehandelt. Es gibt zunächst zwei große Klassen von Kräften, **kovalente Bindungskräfte (bonded interactions)**,

die kovalente Bindungen beschreiben, und **schwache Bindungskräfte (non-bonded interactions)**, die auf der Coulombwechselwirkung basierende schwächere Wechselwirkungen (Coulomb-, Dipol-, Van-der-Waals Wechselwirkungen) umfassen.

Kovalente Bindungskräfte

Wir betrachten zunächst die starke **kovalente Bindung**, zwischen zwei Atomen, die auf dem Zusammenspiel von quantenmechanischem Symmetrisierungspostulat (Pauli-Prinzip) und Coulombwechselwirkung beruht. Bei einer kovalenten elektronischen Bindung zwischen zwei Atomen bildet sich ein bindendes und antibindendes Molekülorbital aus zwei Atomorbitalen. Zur (näherungsweisen) quantenmechanischen Beschreibung dieses Problems gibt es die Valenzbindungstheorie (nach Heitler, London, Pauling) und die Molekülorbitaltheorie (nach Hund, Mulliken). Der einfachste Fall ist die σ -Bindung zwischen zwei kugelsymmetrischen s-Atomorbitalen A und B.

In der **Molekülorbitaltheorie** betrachtet man zunächst das Problem Molekülorbitale für *ein* Elektron zu finden, das mit beiden Kernen A und B wechselwirkt. Ansätze für Molekülorbitale sind meist Linearkombinationen von Atomorbitalen ψ_A und ψ_B ; in diesem einfachen Fall stellen sich im Variationsansatz $\psi_A + \psi_B$ und $\psi_A - \psi_B$ als stationäre Zustände heraus. Von diesen stellt der symmetrische Zustand $\psi_A + \psi_B$ einen *bindendes Molekülorbital* und $\psi_A - \psi_B$ ein *antibindendes Orbital* dar, siehe Abb. 5.2. Das bindende Orbital wird dann von 2 Elektronen nach Pauli-Prinzip besetzt. Die Wellenfunktion des Moleküls mit *zwei* Elektronen (1 und 2) ist dann $(\psi_A(1) + \psi_B(1))(\psi_A(2) + \psi_B(2))$ mit einem antisymmetrischen Singlett als Spinanteil. Insbesondere enthält diese Wellenfunktion noch "ionische" Anteile wie $\psi_A(1)\psi_A(2)$, wo beide Elektronen an einem Kern lokalisiert sind (gute Approximation, wenn Kerne A und B nah zusammen).

In der **Valenzbindungstheorie (Heitler-London-Theorie)** geht man für den quantenmechanischen Zustand des Moleküls mit *zwei* Elektronen (1 und 2) von einem Produktansatz mit den Wellenfunktionen der s-Atomorbitale, $\psi_A(1)\psi_B(2)$, aus, d.h. "ionische" Anteile wie $\psi_A(1)\psi_A(2)$ werden vollständig vernachlässigt (gute Approximation, wenn Kerne A und B weit entfernt). Beim *bindenden Zustand* liegen die Spinanteile als antisymmetrisches Singlett und die Ortsanteile der Wellenfunktionen dementsprechend in einem *symmetrierten Produktzustand* $\psi_A(1)\psi_B(2) + \psi_A(2)\psi_B(1)$ vor. Beim *antibindenden Zustand* liegen die Spinanteile dagegen als symmetrisches Triplet und die Ortsanteile dementsprechend in einem *antisymmetrierten Produktzustand* $\psi_A(1)\psi_B(2) - \psi_A(2)\psi_B(1)$ vor.

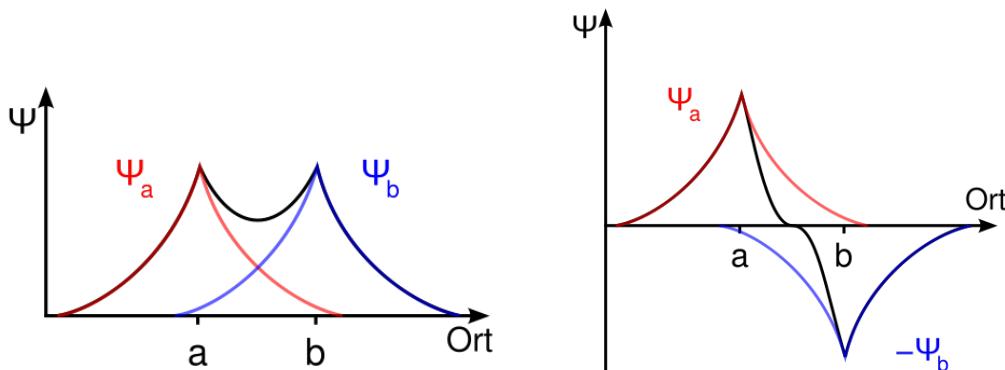


Abbildung 5.2: Bindendes (links) und antibindendes (rechts) Molekülorbital. Links kommt es zu einer erhöhten negativen Elektronenladungsdichte *zwischen* den positiv geladenen Kernen, was die Coulombenergie absenkt und zur Bindung führt. (Quelle: Wikipedia).

Dass in beiden Zugängen ein symmetrisches Produkt der Ortsanteile den Grundzustand darstellen muss, folgt entweder aus der Beobachtung, dass der Ortsanteil nur dann keine Knoten hat oder aus der Tatsache, dass sich dann die negativ geladenen Elektronen bevorzugt *zwischen* den positiv geladenen Kernen aufhalten, was die Coulombwechselwirkung minimiert. Dies entspricht dann auch der Vorstellung, dass sich die beiden in der kovalenten Bindung beteiligten Atome ein Elektronenpaar „teilen“, das zwischen den Kernen eine bindende „Elektronenwolke“ bildet. Typische Werte für Bindungsenergien sind $\Delta E \simeq 140k_B T \simeq 3.5\text{eV}$ (bei Raumtemperatur $T = 293\text{K}$) für eine einfache kovalente C–C Kohlenstoffbindung.

Die **Quantenchemie**, die auch einen wichtigen Zweig der Computerphysik bzw. Computerchemie darstellt, befasst sich damit Bindungsenergien mit numerischen Methoden genau zu berechnen. Dazu werden meist numerische Implementationen der Molekülorbitaltheorie verwendet, die mit Linearkombinationen von Atomorbitalen (LCAO-Methode) arbeiten. Dies wird hier aber nicht Thema sein.

Bei der **metallischen Bindung** werden *viele* Elektronen (Elektronengas) im periodischen Potential der Kerne delokalisiert. Die entsprechenden Quantenzustände spalten dann in ein ganzes *Energieband* auf, das typischerweise eine Breite von $E_B \simeq 1 - 3\text{ eV} \sim 40 - 120k_B T$ hat. In einem Metall ist das Leitungsband nicht vollständig besetzt und die Bindungsenergie pro Elektron ergibt sich als über die besetzten Zustände gemittelte Energiedifferenz zum atomaren Zustand, siehe Abb. 5.3 (rechts).

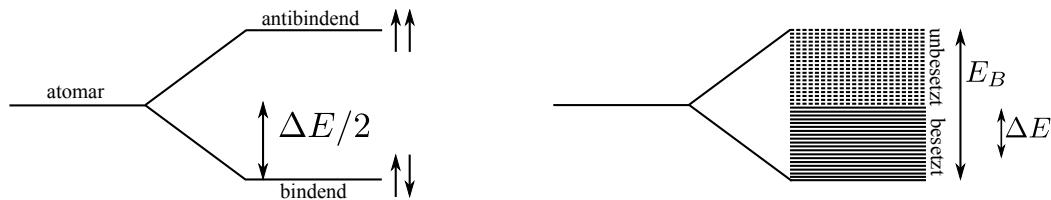
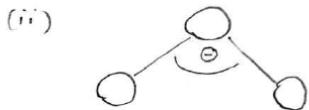


Abbildung 5.3: Links: Energieaufspaltung in 2 Zustände (bindend, antibindend) bei der kovalenten σ -Bindung. Der bindende Zustand wird von dem Elektronenpaar zweifach besetzt und die Gesamt-Bindungsenergie ΔE ist die zweifache Energiedifferenz zwischen bindendem Zustand und ursprünglichem atomaren Zustand. Für einen C–C Bond ist $\Delta E \simeq 140k_B T$. Rechts: Energieaufspaltung in ein Energieband bei der metallischen Bindung. Typisch sind Bandbreiten $E_B \simeq 1 - 3\text{ eV}$. Die Bindungsenergie ΔE pro Elektron ist die über die besetzten Zustände gemittelte Energiedifferenz zum atomaren Zustand.

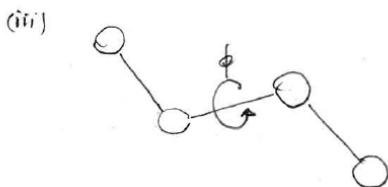
Für klassische MD-Simulationen sind nicht die Bindungsenergien selbst, sondern die Energieänderungen relevant, wenn eine kovalente Bindung durch Bewegung der Atomrümpfe verformt wird, da diese die Kräfte zwischen Atomen bestimmen. Dafür werden Energien mit gewissen Potentialparametern angegeben. Diese Bindungspotentiale werden für MD-Simulationen oft in empirischen **„Kraftfeldern“ (force fields)** zusammengefasst (als gebräuchliche Kraftfelder haben sich beispielsweise Amber, Charmm, Gromos etabliert), in denen die Potentialparameter üblicherweise durch Vergleich von MD-Simulationen mit gewissen experimentellen thermodynamischen Größen optimiert worden sind. Die wichtigsten Potentiale und ihrer Parameter für kovalente Bindungen sind:



(i) Bond-Dehnung:
Gleichgewichts-Bondlänge r_0 , Dehnbarkeit k ,
Potential $V(r) = \frac{1}{2}k(r - r_0)^2$



(ii) Bond-Biegung: 3-Körper-Potential,
Gleichgewichts-Bondwinkel θ_0 ,
Biegesteifigkeit K_θ ,
Potential $V(\theta) = \frac{1}{2}K_\theta(\theta - \theta_0)^2$



(iii) Bond-Torsion: 4-Körper-Potential

Schwache Bindungs Kräfte

Alle schwachen Bindungs Kräfte basieren auf der Coulomb-Wechselwirkung.

- (i) Geladene Teilchen im Abstand r mit Ladungen q_1 und q_2 wechselwirken mit der **Coulomb-Wechselwirkung**

$$V(r) = \frac{1}{4\pi\varepsilon_0} \frac{q_1 q_2}{r}.$$

- (ii) Dipole (polare Moleküle) wechselwirken mit der schwächeren (und richtungsabhängigen) **Dipol-Dipol-Wechselwirkung**

$$V(r) = \frac{1}{4\pi\varepsilon_0} \left[\frac{\vec{p}_1 \cdot \vec{p}_2}{r^3} - \frac{3(\vec{p}_1 \cdot \vec{r})(\vec{p}_2 \cdot \vec{r})}{r^5} \right].$$

mit der Abstandsabhängigkeit $V(r) \propto \frac{1}{r^3}$. Je nach Orientierung der Dipole kann die Wechselwirkung anziehend oder abstoßend sein.

- (iii) Induzierte Dipole wechselwirken mit der **Van-der-Waals Wechselwirkung** (auch **London-Kraft** oder **Dispersionskraft**). Wird ein Dipolmoment durch eine zufällige thermische (oder quantenmechanische) Fluktuation erzeugt, ergibt sich momentan ein E-Feld $E \propto r^{-3}$. In diesem E-Feld kann ein weiteres Dipolmoment \vec{p} induziert werden in einem anderen Molekül. Bei einer **Polarisierbarkeit** α gilt $\vec{p} = \alpha \vec{E} \propto \alpha r^{-3}$. Damit ergibt sich für die Wechselwirkungsenergie zwischen spontan erzeugtem und induzierten Dipolmoment

$$V(r) = -\vec{p} \cdot \vec{E} \propto -\frac{\alpha}{r^3} \frac{1}{r^3} \propto -\frac{\alpha}{r^6}.$$

Die Van-der-Waals Wechselwirkung ist immer **anziehend**. Die Van-der-Waals Anziehung ist zwar schwach aber allgegenwärtig, da sie auch für jedes neutrale Teilchen auftritt, wobei ihre Stärke dann von den Polarisierbarkeiten abhängt. Dadurch summieren sich die eigentlich schwachen Van-der-Waals Wechselwirkungen über viele Teilchen oft zu relativ starken Adhäsionskräften auf makroskopischen Skalen (die z.B. auch für die Haftung eines Geckos an einer vertikalen Wand verantwortlich sind). Daher spielen Van-der-Waals Wechselwirkungen im Folgenden eine dominierende Rolle in Vierteilchensystemen von neutralen Teilchen.

Lennard-Jones-Potential

Daneben gibt es noch eine starke kurzreichweitige Abstoßung zwischen Atomen (nach Born), wenn sich die beiden Elektronenwolken stark überlagern. Diese **Born-Abstoßung** kann als “harter Kern” einer Atoms angesehen werden, in den kein anderes Atom eindringen kann und wird oft durch ein Potenzgesetz mit einem großen Exponenten beschrieben, üblicherweise mit dem Exponenten 12:

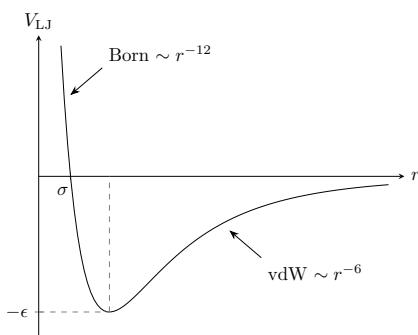
$$V(r) \propto \frac{1}{r^{12}}.$$



Abbildung 5.4: Links: John Edward Lennard-Jones (1894–1954), Mathematiker und theoretischer Physiker, Mitbegründer der modernen Computerchemie. Mitte: Johannes Diderik van der Waals (1837–1923), niederländischer Physiker (Nobelpreis 1910). Rechts: Fritz London (1900–1954), deutsch-amerikanischer Physiker. (Quelle: Wikipedia, AIP).

Born-Abstoßung und Van-der-Waals Anziehung sind für ein **einatomiges neutrales Gas** tatsächlich die wichtigsten zu berücksichtigenden Wechselwirkungen und werden üblicherweise im sogenannten **Lennard-Jones Potential** zusammengefasst:

$$V_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right]. \quad (5.3)$$



Das Lennard-Jones Potential hat ein Energieminimum $V_{\min} = -\epsilon$ bei $r = 2^{1/6}\sigma$. Der Nulldurchgang ist bei $r = \sigma$. Die zugehörige Kraft ist

$$\vec{F}_{\text{LJ}}(\vec{r}) = \frac{\vec{r}}{r} \frac{48\epsilon}{r} \left[\left(\frac{\sigma}{r}\right)^{12} - \frac{1}{2} \left(\frac{\sigma}{r}\right)^6 \right]. \quad (5.4)$$

Das Lennard-Jones Potential enthält zwei Parameter: Der Parameter $\epsilon (> 0)$ gibt die **Energie-skala** an und ist proportional zur Stärke der Van-der-Waals Anziehung. Der Parameter σ ist eine **Längenskala**, die von der Größe der Teilchen bzw. ihres harten Kerns bestimmt wird.

Ein einfaches **Beispiel** für ein System, wo die Lennard-Jones Wechselwirkung tatsächlich eine sehr gute Approximation an gemessene Wechselwirkungspotentiale darstellt, sind fluide Edelgasphasen, z.B. ein **Argongas**, da Edelgasatome neutral und ungebunden sind. Realistische Parameter für eine MD-Simulation von Argon mit einem Lennard-Jones Potential sind $\sigma \simeq 3.4\text{\AA}$, $\epsilon \simeq k_B 120\text{K} \simeq 0.4k_B T$ (bei Raumtemperatur $T = 293\text{K}$) und $m \simeq 6.6 \cdot 10^{-23}\text{g}$.

Anhand dieser Werte kann man sich auch bereits **typische Zeitskalen** eine MD-Simulation klar machen. Dazu führen wir in der Newtonschen Bewegungsgleichung $m\ddot{\vec{r}}_i = \sum_{j(\neq i)} \vec{F}_{\text{LJ}}(\vec{r}_{ij})$ dimensionslose **reduzierte** Größen

$$\tilde{r} = \frac{r}{\sigma} \quad \text{und} \quad \tilde{t} = \frac{t}{\tau}$$

und fragen, wie τ zu wählen ist, damit die Newtonsche Bewegungsgleichung in reduzierten Größen parameterfrei wird:

$$m \frac{\sigma}{\tau^2} \frac{d^2 \tilde{r}}{d\tilde{t}^2} = -\frac{48\varepsilon}{\sigma} \sum_{j(\neq i)} \frac{\tilde{r}_{ij}}{\tilde{r}_{ij}^{13}} \left[\frac{1}{\tilde{r}_{ij}^{13}} - \frac{1}{2} \frac{1}{\tilde{r}_{ij}^7} \right]. \quad (5.5)$$

Wir sehen, dass die Wahl $m\sigma/\tau^2 = 48\varepsilon/\sigma$ oder

$$\tau = \left(\frac{m\sigma^2}{48\varepsilon} \right)^{1/2} \simeq 3 \cdot 10^{-13} \text{s} \quad (\text{für Argon}) \quad (5.6)$$

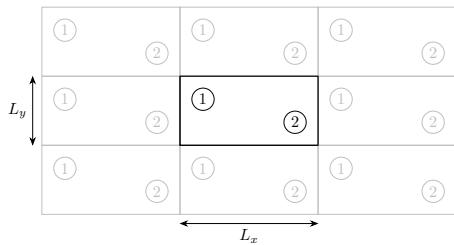
zum Ziel führt. Bei der MD-Simulation sollte nun ein typischer numerischer Zeitschritt h bei der Integration der Bewegungsgleichungen ein Bruchteil von τ sein, also z.B. $h = 0,01\tau$. Selbst bei 10^6 MD-Simulationsschritten h führt dies dann zu einer Gesamt-Simulationsdauer von nur $3 \cdot 10^{-9} \text{s} \simeq 3 \text{ ns}$! Dies ist ein typisches Problem von MD-Simulationen. Sie geben zwar eine mikroskopisch korrekte Dynamik und können auf einem “all-atom” Niveau durchgeführt werden, allerdings sind nur sehr **kleine Simulationszeiten** unterhalb der ns-Zeitskala erreichbar (insbesondere bei komplexeren Systemen als einem einfachen einatomigen Gas).

5.2.2 Randbedingungen

Wir können natürlich (noch) keine 10^{23} Teilchen simulieren sondern typischerweise $10^2 - 10^6$. Um dennoch realistische Teilchendichten $\rho = N/V$ zu erreichen, muss das Simulationsvolumen V entsprechend klein sein. Typischerweise wählt man eine **endliche Box** mit Maßen $L_x \times L_y \times L_z$, die durch entsprechende **Randbedingungen** realisiert wird.

Eine Möglichkeit besteht darin, einfach ein begrenzendes, abstoßendes, externes “Wand-Potential” einzuführen.

Oft bevorzugt man jedoch in einer Simulation **periodische Randbedingungen**, da auf diese Art auch bei fester Teilchendichte ρ immer noch ein “quasi-unendliches” System realisiert werden kann, indem die Box mit Maßen $L_x \times L_y \times L_z$ in alle 3 Raumrichtungen periodisch fortgesetzt wird.



Periodische Randbedingungen in 2 Raumdimensionen. Die eigentliche Simulationsbox mit Maßen $L_x \times L_y$ und ihre periodischen Bilder in beide Raumrichtungen. Die Simulation findet in $0 < x < L_x$ und $0 < y < L_y$ statt.

Bei der Implementierung periodischer Randbedingungen tauchen 2 Probleme auf:

- (i) Es muss sichergestellt sein, dass ein Teilchen, was sich aus der Simulationsbox herausbewegt, wieder korrekt periodisch am anderen Ende des Systems eingesetzt wird.

Dies kann durch die Vorschriften



$$x' = x - L_x \text{floor}(x/L_x)$$

$$y' = y - L_y \text{floor}(y/L_y)$$

erreicht werden, wo $\text{floor}(x)$ die x nächste *kleinere* integer-Zahl bezeichnet.

- (ii) Bei der Berechnung von Paarwechselwirkungen muss darauf geachtet werden, dass ein Teilchen i nicht nur mit einem Teilchen j , sondern auch mit allen periodischen Bildern von j wechselwirkt. Wir müssen also eine **Summation über Bildteilchen** vornehmen bei der Kraftberechnung.

Die “direkte” Kraft auf Teilchen i von Teilchen j auf Grund eines Paar-Wechselwirkungspotentials $V(r)$ ist

$$\vec{F}_{ij} = -\frac{\vec{r}_{ij}}{r_{ij}} V'(r_{ij}).$$

Mit $n\vec{L} = (n_x L_x, n_y L_y, n_z L_z)$ und $\vec{n} = (n_x, n_y, n_z) \in \mathbb{Z}^3$ ist dann die Kraft auf i von Teilchen j und allen periodischen Bildern von j

$$\vec{F}_{ij} = - \sum_{\vec{n} \in \mathbb{Z}^3} \frac{\vec{r}_{ij} + n\vec{L}}{|\vec{r}_{ij} + n\vec{L}|} V'(|\vec{r}_{ij} + n\vec{L}|).$$

Die Gesamtkraft auf i geht in die Bewegungsgleichungen ein:

$$\vec{F}_i = \sum_{j(\neq i)} \vec{F}_{ij} = - \sum_{j(\neq i)} \sum_{\vec{n} \in \mathbb{Z}^3} \frac{\vec{r}_{ij} + n\vec{L}}{|\vec{r}_{ij} + n\vec{L}|} V'(|\vec{r}_{ij} + n\vec{L}|). \quad (5.7)$$

Hier gibt es keine Beiträge von $i = j$, auch nicht über periodische Bildteilchen von i selbst, da diese sich immer mit i mitbewegen und so keine Kraft erzeugen können.

Die Summe $\sum_{\vec{n} \in \mathbb{Z}^3}$ über alle Bilder wird sehr problematisch bei einer **Coulomb-Wechselwirkung** wegen der *Langreichweite* $V(r) \propto 1/r$, die dazu führt dass

$$\sum_{\vec{n}} \frac{1}{|\vec{r} + n\vec{L}|} = \infty.$$

Allerdings gibt es immer gleich viele positive und negative Ladungen, da das Gesamtsystem neutral sein sollte. Dann ist die Summe $\sum \vec{n}$ alternierend und bedingt konvergent. Um solche bedingten konvergenten Coulomb-Summationen schnell und genau auszuführen, gibt es spezielle Summationstechniken wie die **Ewald-Summation**, die die Summe teilweise im Fourieraum ausführen.

Bei der **Lennard-Jones Wechselwirkung** mit $V_{\text{LJ}} \propto r^{-6}$ gibt es kein prinzipielles Summationsproblem, da diese *kurzreichweite* ist und alle Summen konvergieren. Trotzdem muss die Summe $\sum_j \sum_{\vec{n}}$ in (5.7) abgeschnitten werden bei einer numerischen Berechnung. Dabei gibt es 2 Möglichkeiten:

- (a) Ein **einfacher Cutoff**, wo alle Terme mit $|\vec{r}_{ij} + n\vec{L}| > r_c \equiv \text{cutoff}$ abgeschnitten werden, d.h. wir benutzen effektiv ein Potential

$$V(r) = \begin{cases} V_{\text{LJ}}(r) & r < r_c \\ 0 & r > r_c, \end{cases}$$

was aber *unstetig* ist bei $r = r_c$. Besser ist daher, **Cutoff und Potentialverschiebung** zu kombinieren,

$$V(r) = \begin{cases} V_{\text{LJ}}(r) - V_{\text{LJ}}(r_c) & r < r_c \\ 0 & r > r_c, \end{cases}$$

was das Potential stetig macht und die Kräfte nicht ändert.

- (b) Wir können auch lediglich die periodischen Bilder abschneiden, d.h. wir beschränken die Summe $\sum_{\vec{n} \in \mathbb{Z}^3}$ auf *einen* Term, der $|\vec{r}_{ij} + n\vec{L}|$ minimiert, also das *nächste* Teilchen unter allen periodischen Bildern darstellt (“**minimales Bild**”). Diese Cutoff-Prozedur ist äquivalent zu Möglichkeit (a) mit $r_c = \frac{1}{2} \min(L_x, L_y, L_z)$.

5.2.3 Initialisierung

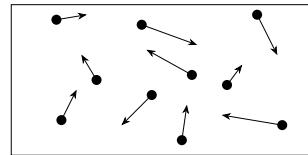
Bei der Initialisierung des Systems sind folgende Punkte zu beachten:

- **Energieerhaltung:** Der MD-Algorithmus sollte die Gesamtenergie E erhalten, daher ist E bereits durch die Anfangspositionen und Anfangsgeschwindigkeiten festgelegt.
- **Schwerpunktsgeschwindigkeit:** Die Schwerpunktsgeschwindigkeit $\vec{v}_s = \frac{1}{N} \sum_{i=1}^N \vec{v}_i$ ist ebenfalls erhalten. Um zu vermeiden, dass das System als Ganzes driftet (die interessante Physik z.B. der Phasenübergänge ergibt sich aus den Relativbewegungen der Teilchen), wollen wir $\vec{v}_s(0)$ am Anfang so wählen, dass der Schwerpunkt ruht, also $\vec{v}_s = 0$ gilt bei $t = 0$. Insbesondere soll vermieden werden, dass die Schwerpunktsbewegung zur Temperatur (5.2) beiträgt, was unphysikalisch wäre (ein perfektes Gitter, in dem alle Teilchen relativ zueinander ruhen, das aber als Ganzes durch den Raum driftet, sollte trotzdem $T = 0$ haben).
- **Teilchenabstand:** Bei $t = 0$ sollte ein ausreichend großer Abstand zwischen den Teilchen realisiert sein, um das Auftreten großer Kräfte zu vermeiden. Solche großen Kräfte können bei unvorsichtiger Wahl des Integrationsschrittes zum "explosionsartigen Auseinanderfliegen" von Teilchen führen. Vermeidung solcher Situationen am Anfang erlaubt eine einfachere und bessere Äquilibrierung (siehe unten).

Eine typische Initialisierung z.B. für ein einatomiges Gas wird daher folgendermaßen aussehen:

- (i) Wir ordnen die Teilchen gleichmäßig, z.B. auf kubischen Gitterplätzen an (um kleine Abstände zu vermeiden).
- (ii) Wir geben jedem Teilchen eine zufällige Geschwindigkeit. Dann ziehen wir die Schwerpunktsgeschwindigkeit bei jedem Teilchen wieder ab:

$$\vec{v}_i \rightarrow \vec{v}_i - \frac{1}{N} \sum_i \vec{v}_i \Rightarrow \vec{v}_s(0) = 0.$$



- (iii) Wir reskalieren $\vec{v}_i \rightarrow \alpha \vec{v}_i$ alle Geschwindigkeiten, um eine bestimmte Gesamtenergie E oder kinetische Energie E_{kin} (und damit z.B. eine bestimmte Temperatur $T = 2E_{kin}/k_B N_f$ nach (5.2) bei $t = 0$ zu erzeugen.

5.3 Integration

Wir erarbeiten einige spezielle Aspekte des Verlet-Algorithmus, die ihn für MD-Simulationen prädestinieren: Energieerhaltung, Phasenraumvolumenerhaltung und Zeitumkehrinvarianz.

Um die Newtonschen Bewegungsgleichungen in einer MD-Simulation zu lösen, benutzen wir den **Verlet-Algorithmus** [5], siehe auch Kapitel 4.5. Die Gründe sind folgende:

- 1) Die Berechnung aller Paarwechselwirkungs Kräfte \vec{F}_{ij} kostet die meiste Rechenzeit, da dies $\mathcal{O}(N^2)$ Operationen erfordert. Der Verlet-Algorithmus erfordert nur **eine Kraftberechnung** pro Schritt und ist dabei aber immer noch 3. Ordnung in \vec{r} und 1. Ordnung in \vec{v} .
- 1a) Der Verlet-Algorithmus ist damit zwar nicht übermäßig genau (ein Runge-Kutta 4. Ordnung wäre genauer). Dies ist für MD-Simulationen mit $N \gg 1$ Teilchen nicht nötig, da typischerweise nur **Ensemble-Mittelwerte** über alle N Trajektorien interessieren.
- 2a) Der Verlet-Algorithmus hat eine gute **Energieerhaltung** für lange Zeiten.
- 2b) Der Verlet-Algorithmus **erhält das Phasenraumvolumen**, wie vom Liouville-Theorem im

Rahmen der analytischen Mechanik vorhergesagt. Dies ist eine Folge davon, dass der Verlet-Algorithmus **sympplektisch** ist (diese Algorithmen sind kanonische Transformationen in jedem Zeitschritt).

- 3) Der Verlet-Algorithmus ist **zeitumkehrinvariant** (bis auf numerisches Rauschen), d.h. bei Zeitumkehr $t \rightarrow -t$ ($\vec{r} \rightarrow \vec{r}$, $\vec{p} \rightarrow -\vec{p}$) werden die *gleichen* Bahnen umgekehrt durchlaufen.

Die Eigenschaften 2a), 2b) und 3) folgen elegant aus einer Formulierung des Verlet-Algorithmus durch einen **Liouville-Operator** (nach Tuckerman *et al.* [6]). Bevor wir dies zeigen können, wollen wir den Liouville-Operator einführen und einige seiner Eigenschaften ableiten.

Dazu betrachten wir zunächst eine beliebige Funktion $f(\vec{\Gamma})$ im N -Teilchen Phasenraum $\mathbb{P} = \mathbb{R}^{3N} \times \mathbb{R}^{3N}$ der Vektoren $\vec{\Gamma} = (\{\vec{r}\}, \{\vec{p}\}) = (\vec{r}_1, \dots, \vec{r}_N, \vec{p}_1, \dots, \vec{p}_N)$. Die **Zeitentwicklung** der Funktion f ,

$$\dot{f} = \sum_{i=1}^N \left(\dot{\vec{r}}_i \cdot \vec{\nabla}_{\vec{r}_i} f + \dot{\vec{p}}_i \cdot \vec{\nabla}_{\vec{p}_i} f \right) \equiv i\hat{L}f, \quad (5.8)$$

definiert den **Liouville-Operator** \hat{L} . Mit einer **Hamiltonfunktion** H und den Hamiltonschen Bewegungsgleichungen

$$\dot{\vec{r}}_i = \vec{\nabla}_{\vec{p}_i} H \quad \text{und} \quad \dot{\vec{p}}_i = -\vec{\nabla}_{\vec{r}_i} H \quad (5.9)$$

gilt weiter

$$i\hat{L} = \sum_{i=1}^N \left(\dot{\vec{r}}_i \cdot \vec{\nabla}_{\vec{r}_i} + \dot{\vec{p}}_i \cdot \vec{\nabla}_{\vec{p}_i} \right) = \sum_i \left[(\vec{\nabla}_{\vec{p}_i} H) \cdot \vec{\nabla}_{\vec{r}_i} - (\vec{\nabla}_{\vec{r}_i} H) \cdot \vec{\nabla}_{\vec{p}_i} \right] = \{., H\}, \quad (5.10)$$

wo $\{f, g\} = \sum_i \vec{\nabla}_{\vec{r}_i} f \cdot \vec{\nabla}_{\vec{p}_i} g - \vec{\nabla}_{\vec{p}_i} f \cdot \vec{\nabla}_{\vec{r}_i} g$ die Poissonklammer in \mathbb{P} bezeichnet. Der Liouville-Operator \hat{L} beschreibt auch die Zeitentwicklung einzelner Vektoren $\vec{\Gamma}$ im Phasenraum (Spezialfall $f(\vec{\Gamma}) = \vec{\Gamma}$) und statt (5.9) kann man auch schreiben:

$$\dot{\vec{\Gamma}} = i\hat{L}\vec{\Gamma}. \quad (5.11)$$

Die Lösung von (5.8) für ein autonomes System mit nicht explizit zeitabhängiger Hamiltonfunktion $H(\vec{\Gamma})$ und damit nach (5.10) auch einem nicht zeitabhängigem Liouville-Operator \hat{L} ist

$$f(t) = \exp(i\hat{L}t)f(0) \equiv \hat{U}(t)f(0), \quad (5.12)$$

was den **klassischen Propagator** oder **Zeitentwicklungsoperator** $\hat{U}(t)$ definiert. Wegen (5.8) ist eine zu (5.12) äquivalente Aussage die Operator-DGL

$$\dot{\hat{U}} = i\hat{L}\hat{U}. \quad (5.13)$$

Ebenso beschreibt der Propagator $\hat{U}(t)$ auch wieder die Zeitentwicklung einzelner Vektoren $\vec{\Gamma}$ im Phasenraum (Spezialfall $f(\vec{\Gamma}) = \vec{\Gamma}$):

$$\vec{\Gamma}(t) = \hat{U}(t)\vec{\Gamma}(0). \quad (5.14)$$

Wir beweisen nun zwei wichtige Sätze über Eigenschaften der Operatoren \hat{L} und \hat{U} . Zunächst zeigen wir:

$$\hat{L} = \hat{L}^+ \text{ ist hermitesch} \quad \left(\text{bezgl. Skalarprodukt } \langle f, g \rangle \equiv \int d^{3N}\vec{r} \int d^{3N}\vec{p} f^*(\{\vec{r}, \vec{p}\}) g(\{\vec{r}, \vec{p}\}) \right). \quad (5.15)$$

Ab hier werden wir uns der einfacheren Darstellung halber auf $N = 1$ beschränken. Die Verallgemeinerung auf beliebige Teilchenzahlen N ist aber problemlos, es ist lediglich die Dimensionalität der Vektoren \vec{r} und \vec{p} von 3 auf $3N$ anzupassen.

Beweis:

Nach Definition von Hermitizität ist zu zeigen: $\langle f, \hat{L}g \rangle = \langle \hat{L}f, g \rangle$.

$$\langle f, i\hat{L}g \rangle = \int d^3\vec{r} \int d^3\vec{p} f^* (\vec{\nabla}_{\vec{p}} H \cdot \vec{\nabla}_{\vec{r}} g - \vec{\nabla}_{\vec{r}} H \cdot \vec{\nabla}_{\vec{p}} g). \quad (5.16)$$

Wir benutzen den Gauß-Integralsatz und die Beziehung

$$\vec{\nabla} \cdot (f^* g \vec{a}) = (\vec{\nabla} f^*) \cdot \vec{a} g + f^* g (\vec{\nabla} \cdot \vec{a}) + f^* \vec{a} \cdot \vec{\nabla} g,$$

um "partiell zu integrieren". Im 1. Term in (5.16) benutzen wir die Beziehung mit $\vec{\nabla}_{\vec{r}}$, $\vec{a} = \vec{\nabla}_{\vec{p}} H$, im 2. Term in (5.16) mit $\vec{\nabla}_{\vec{p}}$, $\vec{a} = \vec{\nabla}_{\vec{r}} H$. Dies führt auf

$$\begin{aligned} \langle f, i\hat{L}g \rangle &= - \int d^3\vec{r} \int d^3\vec{p} \left[(f^* g) (\vec{\nabla}_{\vec{r}} \cdot \vec{\nabla}_{\vec{p}} H) + g \vec{\nabla}_{\vec{r}} f^* \cdot \vec{\nabla}_{\vec{p}} H \right. \\ &\quad \left. - (f^* g) (\vec{\nabla}_{\vec{p}} \cdot \vec{\nabla}_{\vec{r}} H) - g \vec{\nabla}_{\vec{p}} f^* \cdot \vec{\nabla}_{\vec{r}} H \right] \\ &= - \langle i\hat{L}f, g \rangle, \end{aligned}$$

wobei Randterme für quadratintegrable f, g verschwinden und die unterstrichenen Terme sich wegheben. Die unterstrichenen Terme verschwinden sogar einzeln für einen Hamiltonian der Form $H = \vec{p}^2/2m + V(\vec{r})$, weil $\vec{\nabla}_{\vec{p}} H$ nicht mehr von \vec{r} abhängt und umgekehrt. Also folgt $\langle f, \hat{L}g \rangle = \langle \hat{L}f, g \rangle$ und damit ist (5.15) bewiesen.

Weiter wollen wir zeigen:

$$\hat{L} = \hat{L}^+ \text{ hermitesch} \iff \hat{U}(t) \text{ unitär } \hat{U}^{-1}(t) = \hat{U}^+(t)$$

und $\hat{U}^{-1}(t) = \hat{U}(-t)$.

(5.17)

Beweis:

Der Beweis der Äquivalenz in der ersten Zeile erfolgt wie in der Quantenmechanik mit (5.12) und (5.13):

Wenn $\hat{L} = \hat{L}^+$, dann folgt $\hat{U}\hat{U}^+ = \exp(i\hat{L}t) \exp(-i\hat{L}^+t) = \exp(i\hat{L}t) \exp(-i\hat{L}t) = \hat{1}$. Umgekehrt folgt aus $\hat{1} = \hat{U}(t)\hat{U}^+(t)$ durch Differentiation nach t auf beiden Seiten:

$$0 = \dot{\hat{U}}\hat{U}^+ + \hat{U}\dot{\hat{U}}^+ = i\hat{L}\hat{U}\hat{U}^+ - i\hat{U}(\hat{L}\hat{U})^+ = i(\hat{L} - \hat{L}^+).$$

Dann bleibt noch die zweite Zeile in (5.17) zu zeigen:

$$\hat{U}^{-1}(t) = \hat{U}^+(t) = \exp(-i\hat{L}^+t) = \exp(-i\hat{L}t) = \hat{U}(-t).$$

Damit ist (5.17) gezeigt.

Insbesondere die Unitaritätseigenschaft (5.17) des Zeitentwicklungsoperators hat wichtige physikalische Konsequenzen:

Phasenraumvolumenerhaltung 2b) und Zeitumkehrinvarianz 3)
folgen aus der Unitarität des Propagators \hat{U} .

(5.18)

Beweis zu 2b):

\mathbb{P} -Volumenerhaltung heißtet, dass die Jacobi-Determinante für den Koordinatenwechsel, der genau der Zeitentwicklung (5.14) entspricht, den Betrag 1 hat:

$$\left| \det \frac{\partial(\vec{r}(t), \vec{p}(t))}{\partial(\vec{r}(0), \vec{p}(0))} \right| = 1. \quad (5.19)$$

Um dies zu zeigen, lassen wir den Zeitentwicklungsoperator $\hat{U}(t)$ nach Definition (5.12) auf eine Funktion $f(\vec{\Gamma}(0))$ der Phasenraumpunkte bei $t = 0$ wirken und untersuchen, wie sich die Norm der Funktion bezgl. unseres Skalarproduktes entwickelt ($\langle \dots \rangle_{\vec{\Gamma}(0)}$ heißt Integration bezgl. $\vec{\Gamma}(0)$ im Skalarprodukt):

$$\begin{aligned} \langle \hat{U}f(\vec{\Gamma}(0)) | \hat{U}f(\vec{\Gamma}(0)) \rangle_{\vec{\Gamma}(0)} &= \langle f(\vec{\Gamma}(t)) | f(\vec{\Gamma}(t)) \rangle_{\vec{\Gamma}(t)} \\ &\stackrel{\text{Def. } \langle \dots \rangle}{=} \langle f(\vec{\Gamma}(t)) | f(\vec{\Gamma}(t)) \rangle_{\vec{\Gamma}(t)} \left| \det \frac{\partial \vec{\Gamma}(0)}{\partial \vec{\Gamma}(t)} \right| \\ &\stackrel{(5.19)}{=} \langle f(\vec{\Gamma}(t)) | f(\vec{\Gamma}(t)) \rangle_{\vec{\Gamma}(t)} \\ &\stackrel{\text{Umbenenn. } \vec{\Gamma}(t) \rightarrow \vec{\Gamma}(0)}{=} \langle f(\vec{\Gamma}(0)) | f(\vec{\Gamma}(0)) \rangle_{\vec{\Gamma}(0)}. \end{aligned}$$

Dies bedeutet aber nach Definition genau Unitarität von $\hat{U}(t)$ (Erhaltung von Skalarprodukten). Damit ist 2b) gezeigt. Es gilt also

$$\left| \det \frac{\partial(\vec{r}(t), \vec{p}(t))}{\partial(\vec{r}(0), \vec{p}(0))} \right| = \left| \det \hat{U}(t) \right| = 1.$$

Die letzte Gleichheit folgt aus der Unitarität von \hat{U} (wie für eine unitäre Matrix): $\det \hat{U} = \det \hat{U}^+ = \det \hat{U}^{-1} = 1 / \det \hat{U}$.

Beweis zu 3):

Bei Zeitumkehr $t \rightarrow -t$ transformiert sich $\hat{U}(t) \rightarrow \hat{U}(-t) = \hat{U}^{-1}(t)$ nach (5.17). Also gilt auch $\hat{U}(-t)f(t) = \hat{U}^{-1}(t)f(t) = f(0)$ nach (5.12), d.h. bei Zeitumkehr kehrt sich auch die Bewegung genau um.

Aus (5.18) ergibt sich nun die **Idee**, dass Integrationsalgorithmen, die sich durch einen unitären Zeitentwicklungsoperator \hat{U} (in der diskretisierten Zeit t_n) darstellen lassen, automatisch Phasenraumvolumenerhaltung 2b) und Zeitumkehrinvarianz 3) erfüllen sollten. Wir wollen im Weiteren zeigen, dass dies genau auf den Verlet-Algorithmus zutrifft. Das Ziel ist es, den Verlet-Algorithmus letztlich aus einer Näherung in $\hat{U}(\Delta t) = \exp(i\hat{L}\Delta t)$ zu gewinnen, die auf der einen Seite genügend genau ist für kleine Δt und auf der anderen Seite die Unitarität von \hat{U} erhält.

Dazu betrachten wir die Zerlegung

$$i\hat{L} = \underbrace{\dot{\vec{r}} \cdot \vec{\nabla}_{\vec{r}}}_{\equiv i\hat{L}_1} + \underbrace{\dot{\vec{p}} \cdot \vec{\nabla}_{\vec{p}}}_{\equiv i\hat{L}_2}, \quad (5.20)$$

wo \hat{L}_1 und \hat{L}_2 alleine i.Allg. nicht mehr hermitesch sind. Für einen Hamiltonian der Form $H = \vec{p}^2/2m + V(\vec{r})$, wo $\dot{\vec{r}} = \vec{\nabla}_{\vec{p}}H$ nicht mehr von \vec{r} abhängt und $\dot{\vec{p}} = -\vec{\nabla}_{\vec{r}}H$ nicht mehr von \vec{p} abhängt, sind jedoch auch \hat{L}_1 und \hat{L}_2 hermitesch. Damit sind auch $\exp(i\hat{L}_1 t)$ und $\exp(i\hat{L}_2 t)$ wieder unitär.

Es gilt

$$\begin{aligned} \exp(i\hat{L}_1 t) f(\vec{r}(0), \vec{p}(0)) &\stackrel{(*)}{=} \sum_{n=0}^{\infty} \frac{1}{n!} (\dot{\vec{r}}(0)t)^n \frac{\partial^n}{\partial \vec{r}(0)^n} f(\vec{r}(0), \vec{p}(0)) \\ &\stackrel{\text{Taylor}}{=} f(\vec{r}(0) + \dot{\vec{r}}(0)t, \vec{p}(0)) \end{aligned} \quad (5.21)$$

mit $\dot{\vec{r}}(0) = \frac{1}{m}\vec{p}(0)$, was dazu führt, dass der Operator $\partial/\partial\vec{r}(0)$ mit $\dot{\vec{r}}(0) = \frac{1}{m}\vec{p}(0)$ vertauscht. Dies wurde in (*) benutzt. Analog gilt

$$\exp(i\hat{L}_2 t) f(\vec{r}(0), \vec{p}(0)) = f(\vec{r}(0), \vec{p}(0) + \dot{\vec{p}}(0)t) \quad (5.22)$$

mit $\dot{\vec{p}}(0) = \vec{F}(\vec{r}(0))$. Der Operator $\exp(i\hat{L}_1 t)$ generiert also eine *Translation in \vec{r}* , während $\exp(i\hat{L}_2 t)$ eine *Translation in \vec{p}* erzeugt.

Nun gilt aber $\exp(i\hat{L}t) = \exp(i(\hat{L}_1 + \hat{L}_2)t) \neq \exp(i\hat{L}_1 t) \exp(i\hat{L}_2 t)$. Um den Zeitentwicklungsoperator \hat{U} entsprechend (5.20) zu zerlegen, müssen wir also anders vorgehen. Dabei hilft die **Trotter-Identität**

$$\exp(\hat{A} + \hat{B}) = \lim_{P \rightarrow \infty} \left(e^{\hat{A}/2P} e^{\hat{B}/P} e^{\hat{A}/2P} \right)^P = \left(e^{\hat{A}/2P} e^{\hat{B}/P} e^{\hat{A}/2P} \right)^P e^{\mathcal{O}(1/P^2)}. \quad (5.23)$$

Die Trotter-Formel ist wichtig in der Quantenmechanik im Zusammenhang mit Pfadintegralen und etwas aufwendiger im Beweis, den wir deshalb hier nicht geben wollen (siehe z.B. Schulman [7]). Angewandt auf $\exp(i\hat{L}t) = \exp(i(\hat{L}_1 + \hat{L}_2)t)$ mit $P = t/\Delta t$ führt die Trotter-Formel (5.23) auf die Näherung

$$\exp(i\hat{L}t) = \underbrace{\left(e^{i\hat{L}_2 \Delta t/2} e^{i\hat{L}_1 \Delta t} e^{i\hat{L}_2 \Delta t/2} \right)^{t/\Delta t}}_{\equiv \hat{U}_V(\Delta t)} e^{\mathcal{O}(\Delta t^2)}. \quad (5.24)$$

Die Zeitentwicklung lässt sich also näherungsweise als durch $t/\Delta t$ -fache Anwendung des **Verlet-Operators** $\hat{U}_V(\Delta t)$ darstellen. Dieser Operator ist **unitär**, weil $\exp(i\hat{L}_1 t)$ und $\exp(i\hat{L}_2 t)$ unitär sind, und es gilt auch $\hat{U}_V^{-1}(\Delta t) = \hat{U}_V(-\Delta t)$.

Was bewirkt der Verlet-Operator $\hat{U}_V(\Delta t)$ bei Anwendung auf eine Funktion $f(\vec{r}(0), \vec{p}(0))$?

$$\begin{aligned} \hat{U}_V(\Delta t) f(\vec{r}(0), \vec{p}(0)) &\stackrel{(5.22)}{=} e^{i\hat{L}_2 \Delta t/2} e^{i\hat{L}_1 \Delta t} f(\vec{r}(0), \vec{p}(0) + \dot{\vec{p}}(0) \frac{\Delta t}{2}) \quad \text{mit } \dot{\vec{p}}(0) = \vec{F}(\vec{r}(0)) \\ &\equiv \vec{p}(\Delta t/2) \\ &\stackrel{(5.21)}{=} e^{i\hat{L}_2 \Delta t/2} f(\underbrace{\vec{r}(0) + \dot{\vec{r}}(\Delta t/2) \Delta t}_{\equiv \vec{r}(\Delta t)}, \vec{p}(\Delta t/2)) \quad \begin{aligned} \dot{\vec{r}}(\Delta t/2) &= \frac{1}{m} \vec{p}(\Delta t/2) \\ &= \frac{1}{m} \vec{p}(0) + \frac{1}{m} \frac{\Delta t}{2} \vec{F}(\vec{r}(0)) \end{aligned} \\ &\stackrel{(5.22)}{=} f(\vec{r}(\Delta t), \underbrace{\vec{p}(\Delta t/2) + \dot{\vec{p}}(\Delta t) \frac{\Delta t}{2}}_{\equiv \vec{p}(\Delta t)}) \quad \begin{aligned} \dot{\vec{p}}(\Delta t) &= \vec{F}(\vec{r}(\Delta t)) \end{aligned} \\ &= f(\vec{r}(\Delta t), \vec{p}(\Delta t)). \end{aligned}$$

Zusammengenommen lesen wir folgende Transformation der Argumente \vec{r} und \vec{p} ab:

$$\begin{aligned} \vec{p}(\Delta t) &= \vec{p}(0) + (\vec{F}(0) + \vec{F}(\Delta t)) \frac{\Delta t}{2} \\ \vec{r}(\Delta t) &= \vec{r}(0) + \Delta t \dot{\vec{r}}(\Delta t/2) \\ &= \vec{r}(0) + \frac{1}{m} \vec{p}(0) \Delta t + \frac{1}{2m} \vec{F}(0) \Delta t^2. \end{aligned} \quad (5.25)$$

Dies ist aber genau der **Geschwindigkeits-Verlet-Algorithmus** (4.25) aus Kapitel 4.5!

Damit haben wir gezeigt, dass sich der Verlet-Algorithmus als Anwendung des unitären Verlet-Operators $\hat{U}_V(\Delta t)$ darstellen lässt. Da für diesen auch $\hat{U}_V^{-1}(\Delta t) = \hat{U}_V(-\Delta t)$ gilt, folgt dann nach (5.18) automatisch, dass der Verlet-Algorithmus Phasenraumvolumenerhaltung 2b) und Zeitumkehrinvarianz 3) in jedem Zeitschritt erfüllt!

Nun wollen wir noch den Punkt 2a) **Energieerhaltung** des Verlet-Algorithmus diskutieren. Nach einer Zeit t mit Schrittweite Δt gilt mit der exakten Zeitentwicklung für die Hamiltonfunktion H

Energieerhaltung:

$$\begin{aligned} H(t) &= \hat{U}(t)H(0) = e^{i\hat{L}t}H(0) \\ \frac{d}{dt} \left(e^{i\hat{L}t}H(0) \right) &\stackrel{(5.13)}{=} i\hat{L}H(t) \stackrel{(5.10)}{=} \{H, H\} = 0 \\ \Rightarrow e^{i\hat{L}t}H(0) &= H(t) = H(0). \end{aligned}$$

Der Verlet-Algorithmus mit der näherungsweisen Zeitentwicklung $(\hat{U}_V(\Delta t))^{t/\Delta t}$ erfüllt die Energieerhaltung nicht mehr ganz exakt:

$$\begin{aligned} H(t) &= (\hat{U}_V(\Delta t))^{t/\Delta t}H(0) \stackrel{\text{Trotter}}{=} e^{i\hat{L}t+\mathcal{O}(\Delta t^2)}H(0) \\ &= H(0)e^{\mathcal{O}(\Delta t^2)}. \end{aligned}$$

Wir sehen aber, dass der Fehler in der Energie nur $\mathcal{O}(\Delta t^2)$ ist. Obwohl der Verlet-Algorithmus eigentlich nur 1. Ordnung in den Geschwindigkeiten ist, ist er 2. Ordnung für die Energie. Damit gewährleistet er eine vergleichsweise gute Energieerhaltung.

5.4 Messung von Observablen

Wir beschreiben die Messung von thermodynamischen Observablen in einer mikrokanonischen MD-Simulation, insbesondere die Messungen von kinetischer und potentieller Energie, Temperatur und Druck. Wir definieren die Paarverteilungsfunktion als Strukturfunktion und erläutern ihre Messung in einer MD-Simulation. Außerdem leiten wir den Zusammenhang mit der Virial-Zustandsgleichung und der Virialentwicklung her. Schließlich diskutieren wir noch den Nachweis von Phasenübergängen in einer MD-Simulation am Beispiel eines einfachen flüssig-Gas Übergangs.

Die Standardaufgabe in der MD-Simulation besteht darin, einen **thermodynamischen Mittelwert** einer statischen Observablen $O(\{\vec{r}\}, \{\vec{p}\})$ auszurechnen. Dies geschieht durch **Zeitmittelung** über eine MD-Simulationszeit t_{MD} ,

$$\begin{aligned} \langle O \rangle_{MD} &= \frac{1}{t_{MD}} \sum_{n=1}^{t_{MD}} O(\{\vec{r}(n\Delta t)\}, \{\vec{p}(n\Delta t)\}) \\ &= \langle O \rangle = \text{mikrokanonisches Ensemble-Mittel}. \end{aligned} \tag{5.26}$$

Neben der Zeitmittelung mittelt man wenn möglich auch über alle N Teilchen. Ist man an zeitlich veränderlichen, oder momentanen Werten interessiert, wie z.B. der Temperatur $T(t)$ aus (5.2), kann man oft immer noch über alle N Teilchen mitteln.

Grundsätzlich muss aber jede zu messende **statische Größe** durch ein **Zeitmittel** $\langle \dots \rangle$ einer Funktion $O(\{\vec{r}\}, \{\vec{p}\})$ ausgedrückt werden. Die Wahl der Observablen ist bei Energien und Temperatur mehr oder weniger offensichtlich, beim Druck etwas trickreicher. Mit Hilfe der Paarverteilungsfunktion werden wir lernen, wie wir Observablen für die Struktur bzw. räumliche Korrelationen im System formulieren. Wir orientieren uns wieder an einem klassischen N -Teilchen System mit Paarwechselwirkungen und periodischen Randbedingungen, z.B. einem einatomigen Gas mit paarweiser Lennard-Jones Wechselwirkung.

5.4.1 Zeitmittel und Äquilibrierung

Beim **Zeitmittel** einer Observablen ist eine Berechnung (5.26), in der *jeder* Zeitschritt beiträgt, nicht unbedingt besser als eine Mittelung mit gewissen Zeitabständen. Nach kurzen Zeiten sieht das

System nämlich immer noch fast genauso aus und man mittelt prinzipiell über sehr viele fast gleiche Zustände, wenn jeder Zeitschritt zum Mittel herangezogen wird. Dann reduzieren diese zusätzlichen Messungen den Fehler des Mittelwertes nicht.

Optimal ist es daher, jeweils so lange zwischen Messungen zur Mittelung zu warten, bis das System dekorreliert ist, d.h. seinen Ausgangszustand wieder vergessen hat. Diese charakteristische Zeit nennt man **Autokorrelationszeit**. In der Praxis macht man keinen Fehler, wenn man in sehr kurzen Zeitabständen misst, man verliert höchstens Computerzeit, wenn die Messung selbst ein rechenintensiver Vorgang ist. Oft ist man auch an **dynamischen Größen** interessiert, z.B. an Diffusionszeiten eines Teilchens oder der Änderung von Energie, Temperatur oder Druck und muss daher ohnehin in jedem Zeitschritt messen.

Man macht auch keinen Fehler, wenn man in zu langen Zeitabständen misst; dann muss man lediglich länger warten, um den Fehler oder die Schwankung des Mittelwertes unter eine gewünschte Fehlergrenze zu bringen. Die **relative Schwankung** eines Mittelwertes aus N_m unkorrelierten Messungen (also mit zeitlichem Abstand größer als die Autokorrelationszeit) ist nach dem zentralen Grenzwertsatz von der Größenordnung $\sim 1/\sqrt{N_m}$. Misst man zu häufig, sind Messungen nicht unkorreliert und der Fehler wird nicht gedrückt; misst man zu selten, dauert es einfach länger eine gegebene Zahl N_m von Messungen zu erreichen.

Gleiches gilt für die anfängliche **Äquilibrierungszeit**. Das System muss auch erstmal seinen Anfangszustand vergessen, bevor mit der Messung begonnen werden sollte. Die dafür notwendige Zeit ist von der gleichen Größenordnung wie die Autokorrelationszeit. Hier liefert das Äquipartitionstheorem eine gute Möglichkeit, diese Zeitskala näherungsweise am Anfang der Simulation zu bestimmen. Bei unserem einfachen Beispiel des einatomigen Gases werden sich im Laufe der Äquilibrierungsphase kinetische und potentielle Energie angleichen (während die Gesamtenergie konstant bleibt). □ Wenn man eine laufende Mittelung durchführt über immer längere Zeiten oder wenn man Momentanwerte misst, indem man die Energien nur über alle Teilchen mittelt, kann man so bestimmen, wie lange dieser Prozess dauert und damit die Äquilibrierungszeit abschätzen.

Man kann auch zwei Messungen in sehr verschiedenen Anfangszuständen beginnen und den Zeitpunkt feststellen, wann die Messungen konvergieren. Dies gibt auch ein einfaches Maß für die notwendige Äquilibrierungszeit.

5.4.2 Energie, Temperatur

Die **potentielle Energie** E_{pot} ist durch das Zeitmittel aller potentiellen Energien gegeben: $E_{pot} = \langle V_{tot}(\{\vec{r}_i\}) \rangle$. Bei Paarwechselwirkungen gilt

$$E_{pot} = \langle V_{tot}(\{\vec{r}_i\}) \rangle = \frac{1}{2} \sum_{i \neq j} \langle V(\vec{r}_{ij}) \rangle$$

ohne Selbstenergiebeiträge von $i = j$. Bei **periodischen Randbedingungen** ist auch über Wechselwirkungen mit allen Bildern zu summieren (allerdings *nicht* über unphysikalische Wechselwirkungen zwischen Bildern)

$$E_{pot} = \frac{1}{2} \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{i \neq j} \left\langle V(\vec{r}_{ij} + n\vec{L}) \right\rangle.$$

Die **kinetische Energie** E_{kin} erhält man durch

$$E_{kin} = \sum_i \left\langle \frac{\vec{p}_i^2}{2m} \right\rangle.$$

¹ Kinetische und potentielle werden nicht genau gleich sein, stehen aber in der Regel in einem festen Verhältnis. Dies ist eine Folge des Virialsatzes der analytischen Mechanik. Für Potentiale $V(r) \propto r^k$ gilt $E_{kin} = (k/2)E_{pot}$ im zeitlichen Mittel.

Über das **Äquipartitionstheorem** misst man mit Hilfe der kinetischen Energie auch die **Temperatur** T . In (5.2) hatten wir bereits eine momentane Temperatur

$$T(t) = \frac{2}{k_B N_f} \sum_{i=1}^N \frac{\vec{p}_i^2}{2m}$$

eingeführt (mit $N_f = 3N - 3$ auf Grund der Abspaltung der Schwerpunktsbewegung, die auf Null gesetzt werden sollte). Ein zusätzliches Zeitmittel gibt dann

$$T = \frac{2}{k_B N_f} \sum_{i=1}^N \left\langle \frac{\vec{p}_i^2}{2m} \right\rangle = \frac{2}{k_B N_f} E_{kin}. \quad (5.27)$$

Die **spezifische Wärme** $C_V = \frac{\partial E}{\partial T} = \frac{1}{k_B T^2} (\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2)$ ist nur bei einer kanonischen MD-Simulation eine sinnvolle Observable.

5.4.3 Druck

Den **Druck** p kann man in Prinzip direkt messen, wenn man Randbedingungen gewählt hat mit einem externem begrenzendem abstoßendem ‘‘Wand-Potential’’ $V_{ex}(\vec{r})$. Das Potential sollte sich nur in der Richtung $\vec{n}(\vec{r})$ normal zur Wand ändern, so dass die entsprechende Kraft $\vec{F}_{ex,i} = -\vec{\nabla}V_{ex}(\vec{r}_i)$ auf ein Teilchen i immer senkrecht auf der Wand steht und als die von der Wand augeübte Gegenkraft zum Druck p verstanden und gemessen werden kann:

$$p = \left\langle \sum_{i=1}^N \vec{n}(\vec{r}_i) \cdot \vec{F}_{ex,i} \right\rangle. \quad (5.28)$$

Man kann den Druck aber auch messen, ohne die Wandkraft auswerten zu müssen und daher auch ohne explizites Wand-Potential z.B. mit den periodischen Randbedingungen, die wir gewöhnlich wählen. Zur Druckmessung greift man dann auf die **Virialgleichung** (**Virialsatz**) zurück. Das Ergebnis ist, dass der Druck p auch durch folgende Kraftmittelung bestimmbar ist:

$$\begin{aligned} pV &= Nk_B T - \frac{1}{3} \left\langle \sum_{i=1}^N \vec{r}_i \cdot \underbrace{\vec{\nabla}_{\vec{r}_i} V_{tot}}_{= -\sum_{j(\neq i)} \vec{F}_{ij}} \right\rangle \\ &\stackrel{\vec{F}_{ij} = -\vec{F}_{ji}}{=} Nk_B T + \frac{1}{6} \left\langle \sum_{i \neq j} \vec{r}_{ij} \cdot \vec{F}_{ij} \right\rangle. \end{aligned} \quad (5.29)$$

Dieser Virialsatz ist im Prinzip eine Zustandsgleichung, die Korrekturen zur idealen Gasgleichung (erster Term) auf Grund von Paarwechselwirkungen enthält. In Simulationen wird diese viriale Zustandsgleichung aber zur Druckmessung verwendet.

Zur Herleitung dieser Beziehung starten wir mit der Größe

$$G \equiv \sum_{i=1}^N \vec{r}_i \cdot \vec{p}_i. \quad (5.30)$$

Bei einer Bewegung in einem beschränkten Volumen V mit beschränkten Geschwindigkeiten ist G beschränkt. Daher verschwindet der Mittelwert der zeitlichen Ableitung:

$$\left\langle \frac{d}{dt} G \right\rangle \stackrel{\text{Def., Zeitmittel}}{=} \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt \left(\frac{d}{dt} G \right) = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} (G(\tau) - G(0)) = 0.$$

Berechnung von $\frac{d}{dt}G$ und Einsetzen ergibt

$$0 = \left\langle \sum_{i=1}^N \dot{\vec{r}}_i \cdot \vec{p}_i \right\rangle + \left\langle \sum_{i=1}^N \vec{r}_i \cdot \dot{\vec{p}}_i \right\rangle$$

mit $\dot{\vec{p}}_i = \vec{F}_i = -\underbrace{\vec{\nabla}_{\vec{r}_i} V_{tot}}_{\text{von WW.}} + \underbrace{\vec{F}_{ex,i}}_{\text{von Wand augeübte Gegenkraft zum Druck } p}$

$$0 = \underbrace{2E_{kin}}_{= 3k_B TN} - \left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{\nabla}_{\vec{r}_i} V_{tot} \right\rangle + \underbrace{\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_{ex,i} \right\rangle}_{= -3pV}.$$

Die Größe $\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_i \right\rangle$ heißt auch **Virial** (nach Clausius, von lat. *vis* = Kraft). Das Virial hat einen Anteil $-\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{\nabla}_{\vec{r}_i} V_{tot} \right\rangle$ von Wechselwirkungen zwischen Teilchen und einen Anteil $\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_{ex,i} \right\rangle$ von externen Kräften. Der Virialsatz (5.29) zur Druckmessung ist bewiesen, wenn wir die Gleichheit $\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_{ex,i} \right\rangle = -3pV$ für den externen Anteil gezeigt haben.

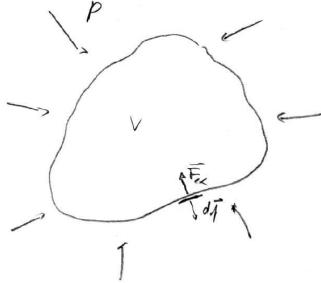


Abbildung 5.5: Illustration der Druckkraft.

Dazu machen wir uns klar, dass nur Teilchen i am Rand die Druckkraft $\vec{F}_{ex,i}$ spüren und außerdem führen wir eine Kontinuumsapproximation durch,

$$\left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_{ex,i} \right\rangle = \sum_{n,m} \int_{\partial V} df_n \sigma_{nm}(\vec{r}) r_m,$$

indem wir einen Spannungstensor σ_{nm} einführen; $\sum_n \sigma_{nm}(\vec{r}) df_n$ misst die m -te Komponente der gesamten externen Kraft auf das Flächenelement $d\vec{f}$ bei \vec{r} :

$$\begin{aligned} \left\langle \sum_{i \text{ in } d\vec{f} \text{ bei } \vec{r}} (\vec{F}_{ex,i})_m \right\rangle &= \sum_n \sigma_{nm}(\vec{r}) df_n \\ &= \langle m\text{-te Komponente der Kraft} \rangle \text{ auf Flächenelement } d\vec{f} \text{ bei } \vec{r}. \end{aligned}$$

Die Gegenkraft zu einem isotropen homogenen Druck entspricht einem Spannungstensor $\sigma_{nm} = -p\delta_{nm}$ und daher folgt mit dem Satz von Gauß die Gleichheit:

$$\sum_{n,m} \int_{\partial V} df_n \sigma_{nm}(\vec{r}) r_m = \sum_{n,m} \int_V dV \partial_n (\sigma_{nm}(\vec{r}) r_m) = - \sum_n \int_V dV p \partial_n r_n = -3pV.$$

Damit ist der Virialsatz (5.29) zur Druckmessung gezeigt. Die Herleitung zeigt, dass hier nicht die Wand-Kraft \vec{F}_{ex} direkt gemessen wird wie in (5.28), sondern die Wand-Kraft zuerst durch die inneren Wechselwirkungs Kräfte ausgedrückt wird, die dann einfacher zu messen sind, auch ohne ein explizites Wand-Potential einführen zu müssen.

5.4.4 Paarverteilung

Wir haben schon bei der Druckmessung (5.29) gesehen, dass Paarabstände \vec{r}_{ij} wesentlich eingehen und daher Positions korrelationen von Teilchenpaaren i, j eine wichtige Rolle spielen können. Wir werden diese Korrelationen mit Hilfe der **Paarverteilungsfunktion** $g(r)$ messen. Diese gibt zum einen direkt Auskunft über die **Struktur** des Systems, z.B. über Nachbarschaftsverhältnisse oder Anordnung der Teilchen in den verschiedenen Phasen. Zum anderen kann die Paarverteilungsfunktion benutzt werden, um die **Virial-Zustandsgleichung** (5.29) in eine andere Form zu bringen.

(i) Wir beginnen mit der **mittleren lokale Teilchendichte** $g^{(1)}(\vec{r})$ am Ort \vec{r} ,

$$\left\langle \sum_{i=1}^N \delta(\vec{r} - \vec{r}_i) \right\rangle \equiv \rho g^{(1)}(\vec{r}), \quad (5.31)$$

wobei $\rho = N/V$ die **Teilchendichte** ist. Die lokale Teilchendichte hat folgende Eigenschaften:

- Die **Normierung** ist durch $\rho \int_V d^3 \vec{r} g^{(1)}(\vec{r}) = N$ gegeben.
- Bei Translationsinvarianz gilt $g^{(1)}(\vec{r}) = \text{const} = 1$.
- Bei periodischen Randbedingungen werden Bilder mit einbezogen,

$$\begin{aligned} \rho g^{(1)}(\vec{r}) &= \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{i=1}^N \left\langle \delta(\vec{r} - (\vec{r}_i + n\vec{L})) \right\rangle \\ \Rightarrow g^{(1)}(\vec{r}) &= g^{(1)}(\vec{r} + n\vec{L}), \end{aligned}$$

und $g^{(1)}(\vec{r})$ ist auch periodisch. Die Normierung im ursprünglichen Volumen V bleibt erhalten.

(ii) Als nächstes definieren wir die **mittlere Teilchenpaardichte** $g^{(2)}(\vec{r}, \vec{r}')$ für ein Paar bei \vec{r} und \vec{r}' ,

$$\left\langle \sum_{i \neq j} \delta(\vec{r} - \vec{r}_i) \delta(\vec{r}' - \vec{r}_j) \right\rangle \equiv \rho^2 g^{(2)}(\vec{r}, \vec{r}') \quad (5.32)$$

mit folgenden Eigenschaften:

- Die **Normierung** ist $\rho^2 \int_V d^3 \vec{r} \int d^3 \vec{r}' g^{(2)}(\vec{r}, \vec{r}') = N(N-1)$.
- Bei Translationsinvarianz hängt $g^{(2)}$ nur von Relativvektoren ab, $g^{(2)}(\vec{r}, \vec{r}') = g^{(2)}(\vec{r} - \vec{r}')$.
- Bei periodischen Randbedingungen gilt

$$\begin{aligned} \rho^2 g^{(2)}(\vec{r}, \vec{r}') &= \sum_{\vec{n} \in \mathbb{Z}^3} \sum_{\vec{m} \in \mathbb{Z}^3} \sum_{i \neq j}^N \left\langle \delta(\vec{r} - (\vec{r}_i + n\vec{L})) \delta(\vec{r}' - (\vec{r}_j + m\vec{L})) \right\rangle \\ \Rightarrow g^{(2)}(\vec{r}, \vec{r}') &= g^{(2)}(\vec{r} + n\vec{L}, \vec{r}' + m\vec{L}) \end{aligned}$$

und $g^{(2)}(\vec{r}, \vec{r}')$ ist periodisch in beiden Argumenten. Die Normierung im ursprünglichen Volumen V bleibt erhalten.

- Bei **Isotropie**, also insbesondere wenn die Paarwechselwirkung $V = V(|\vec{r}|)$ ein Zentralpotential ist gilt, und bei **Translationsinvarianz** gilt

$$g^{(2)}(\vec{r} - \vec{r}') = g(|\vec{r} - \vec{r}'|), \quad (5.33)$$

d.h. $g^{(2)}$ hängt nur vom Relativabstand der Teilchen ab.

(iii) Die Beziehung (5.33) definiert bei Isotropie und Translationsinvarianz die **radiale Paarverteilungsfunktion** $g(r)$ mit der Bedeutung

$$\boxed{\rho g(r) 4\pi r^2 dr = \text{mittlere Teilchenzahl in } [r, r+dr], \\ \text{wenn ein Teilchen bei } r=0.} \quad (5.34)$$

Die **Normierung** ist

$$\int dr \rho g(r) 4\pi r^2 dr = \rho \int d^3\vec{r} g^{(2)}(\vec{r} - \vec{r}') \\ \stackrel{\text{Translationsinv.}}{=} \frac{1}{V} \rho \int d^3\vec{r} \int d^3\vec{r}' g^{(2)}(\vec{r} - \vec{r}') = N - 1$$

im Einklang damit, das wir noch $N - 1$ andere Teilchen im System haben außer dem Teilchen bei $r = 0$.

Die **Messung** der Paarverteilungsfunktion $g(r)$ erfolgt über ein **Histogramm der Paarabstände**:

- Bei periodischen Randbedingungen macht eine Messung von $g(r)$ nur Sinn auf einem Intervall $0 < r < L/2$, wenn $V = L \times L \times L$ das Simulationsvolumen ist.
- Für das Histogramm wird das Intervall $0 < r < L/2$ in N_H “Bins” der Länge $\Delta r = L/2N_H$ aufgeteilt, die jeweils um $r_l = \Delta r/2 + (l-1)\Delta r$ ($l = 1, \dots, N_H$) zentriert sind.
- Gemessen wird zu jeder Zeit t :
 $p_l(t) \equiv$ Zahl der Paare $i \neq j$ mit Abstand $(l-1)\Delta r = r_l - \Delta r/2 < |\vec{r}_{ij}| < r_l + \Delta r/2 = l\Delta r$.
Bei periodischen Randbedingungen werden dabei auch Paare mit periodischen Bildern von Teilchen berücksichtigt (wenn $r < L/2$ braucht man nur die *nächsten* periodischen Bilder berücksichtigen).

Dann erfolgt eine Zeitmittelung, um den Mittelwert $\langle p_l \rangle$ zu bestimmen, der direkt proportional zu $g(r_l)$ ist:

$$\begin{aligned} \langle p_l \rangle &\stackrel{(5.32)}{=} \rho^2 V \int_{r \in [r_l - \Delta r/2, r_l + \Delta r/2]} d^3\vec{r} g^{(2)}(\vec{r}) \\ &= N \rho g(r_l) \underbrace{\frac{4\pi}{3} ((l\Delta r)^3 - ((l-1)\Delta r)^3)}_{\Delta V_l} \\ \Rightarrow g(r_l) &= \frac{\langle p_l \rangle}{\Delta V_l \rho N}. \end{aligned}$$

Die Paarverteilungsfunktion $g(r)$ hat folgende **Eigenschaften**:

- $g(r) \approx 1$ für $r \rightarrow \infty$,
es gibt keine Korrelationen zwischen unendlich weit entfernten Teilchen:

$$\left\langle \sum_{i \neq j} \delta(\vec{r} - \vec{r}_i) \delta(\vec{r}' - \vec{r}_j) \right\rangle \stackrel{|\vec{r} - \vec{r}'| \rightarrow \infty}{\approx} \sum_{i \neq j} \langle \delta(\vec{r} - \vec{r}_i) \rangle \langle \delta(\vec{r}' - \vec{r}_j) \rangle = \frac{N(N-1)}{N^2} \rho^2 \stackrel{N \gg 1}{\approx} \rho^2,$$

daher gilt wegen der Definition (5.32) $g(r) \approx 1$.

- $g(r) = 1$ für ein ideales Gas, da es dort keine Korrelationen gibt und alle Teilchen unabhängig sind.
- Wenn wir “harte” Teilchen haben, die sich für Entfernungen kleiner als ein Teilchendurchmesser D nicht durchdringen können, können sich keine Teilchen im Abstand $r < D$ aufhalten, so dass $g(r) = 0$ für $r < D$ gilt.

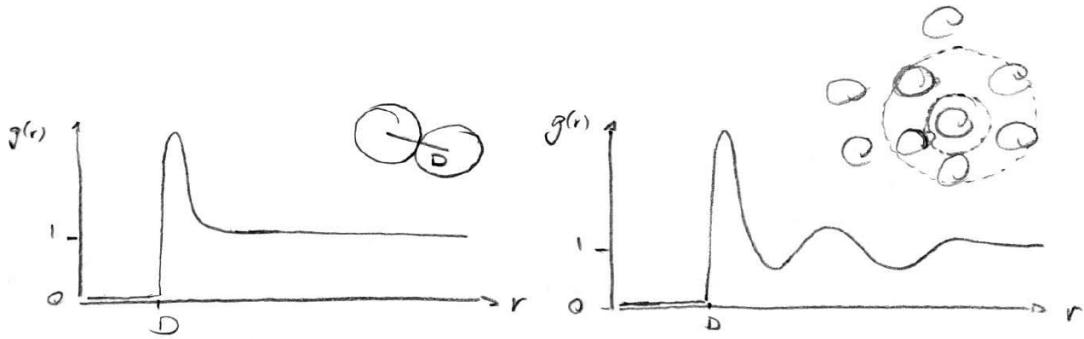


Abbildung 5.6: Links: Typische Paarverteilungsfunktion $g(r)$ in der Gasphase von harten Teilchen mit Durchmesser D mit anziehender Wechselwirkung. Die Paarverteilung hat nur ein Maximum beim wahrscheinlichsten Teilchenabstand. Rechts: Typische Paarverteilungsfunktion $g(r)$ in der flüssigen Phase mit mehreren Maxima. Die Teilchen besitzen eine Nahordnung und ordnen sich in Schalen an.

- In der **Gasphase** einer einfachen Substanz hat $g(r)$ typischerweise nur ein Maximum bei einem r , das den wahrscheinlichsten Teilchenabstand anzeigt. Er sollte ungefähr im Minimum eines anziehenden Wechselwirkungspotentials liegen und etwas größer sein als der Teilchendurchmesser, siehe Abb. 5.6 und 5.7 links. Bei sehr kleinen Dichten wird das Maximum von $g(r)$ genau dort liegen, wo $V(r)$ minimal wird, da z.B. für nur 2 Teilchen die Abstandsverteilung genau die Boltzmann-Verteilung $g(r) \approx \exp(-V(r)/k_B T)$ wird, siehe Gl. (5.36) unten.
- Bei einer **Flüssigkeit** hat $g(r)$ typischerweise mehrere Maxima bei r , die Vielfachen des Potentialminiums sind und deren Höhen mit zunehmendem r monoton abnehmen. Zwischen den Maxima gibt es Minima und $g(r)$ oszilliert um den Wert 1, der für $r \rightarrow \infty$ angenommen wird. Die Maxima haben ihre Ursache in einer typischen **Nahordnung** der Teilchen in einer Flüssigkeit, bei der sich Teilchen in "Schalen" anordnen, siehe Abb. 5.6, rechts und 5.7, Mitte.
- In einer **kristallinen Phase** hat $g(r)$ dagegen unendlich viele Maxima auf Grund der langreichweitigen Kristallordnung, siehe Abb. 5.7 rechts.

5.4.5 Paarverteilung und Virialentwicklung

Die Paarverteilungsfunktion $g(r)$ kann auch benutzt werden, um die **Virialgleichung** (5.29) auf eine anschaulichere Form zu bringen:

$$\begin{aligned}
\frac{1}{2} \left\langle \sum_{i \neq j} \vec{r}_{ij} \cdot \vec{F}_{ij} \right\rangle &= -\frac{1}{2} \left\langle \sum_{i \neq j} (\vec{r}_i - \vec{r}_j) \cdot \vec{\nabla} V(\vec{r}_i - \vec{r}_j) \right\rangle \\
&\stackrel{\text{Def. } g^{(2)}}{=} -\frac{1}{2} \int d^3 \vec{r} \int d^3 \vec{r}' \rho^2 g^{(2)}(\vec{r}, \vec{r}') (\vec{r} - \vec{r}') \cdot \vec{\nabla} V(\vec{r} - \vec{r}') \\
&= -\frac{1}{2} \rho^2 V \int d^3 \vec{r} g(r) \vec{r} \cdot \frac{\vec{r}}{r} V'(r) \quad \text{für Zentralpot. mit } \vec{\nabla} V = \frac{\vec{r}}{r} V'(r) \\
&= -2\pi \rho^2 V \int_0^\infty dr r^3 g(r) V'(r).
\end{aligned}$$

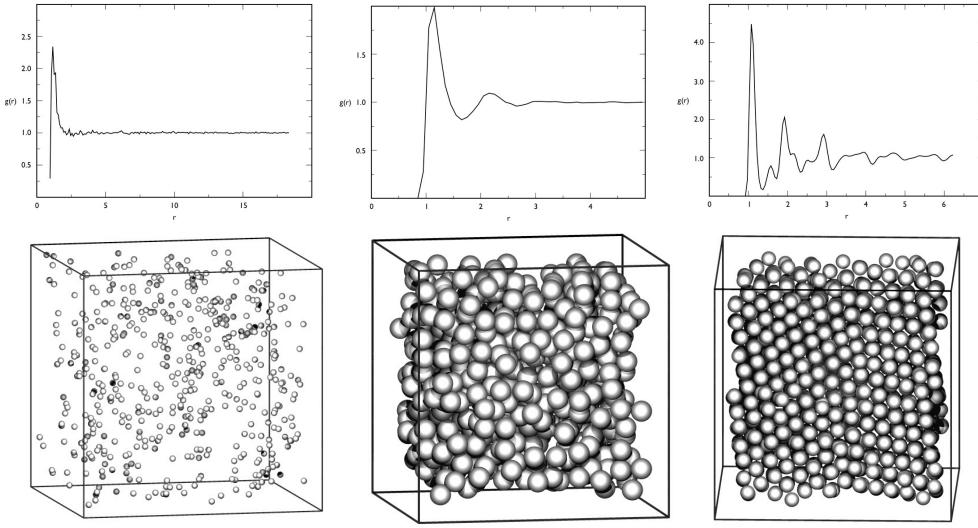


Abbildung 5.7: Oben: Paarverteilungsfunktion $g(r)$ für Lennard-Jones System bei $k_B T = 1.0\epsilon$ und für $\sigma = 1$ aus MD-Simulation. Unten: Typische Teilchenkonfigurationen. Links: Gas bei Dichte $\rho = N/V = 0.01\sigma^{-3}$. Mitte: Flüssigkeit bei Dichte $\rho = N/V = 0.5\sigma^{-3}$. Rechts: Kristall bei Dichte $\rho = N/V = 1\sigma^{-3}$. (Quelle: Iacobella, Christopher R. (2006). Molecular Dynamics simulation of a Lennard-Jones gas. Glotzer group. Depts of Chemical Engineering, Materials Science & Engineering, Macromolecular Science, and Physics, University of Michigan.)

Damit kann (5.29) umgeschrieben werden zur sogenannten **Virial-Zustandsgleichung**

$$\boxed{\frac{pV}{Nk_B T} = 1 - \frac{2\pi}{3} \frac{\rho}{k_B T} \int_0^\infty dr r^3 g(r) V'(r).} \quad (5.35)$$

Wir sehen, dass sich für eine attraktive (repulsive) Wechselwirkung mit $V'(r) > 0$ ($V'(r) < 0$) auch ein kleinerer (größerer) Druck p einstellt wie erwartet.

Um aus (5.35) eine Zustandsgleichung in thermodynamischen Größen zu gewinnen, benötigen wir noch eine Theorie für die Paarverteilung $g(r)$. Hier gibt es in der Theorie der Flüssigkeiten zahlreiche Näherungsmethoden (Percus-Yevick, hypernetted chain, usw.). Wir wollen hier die einfachste Theorie kurz vorstellen für ein **Fluid mit geringer Dichte**. Diese Entwicklung für kleine Dichten wird auf die ersten Ordnungen der **Virialentwicklung** führen.

Dafür betrachten wir zunächst nur zwei Teilchen, von denen das erste Teilchen bei $r = 0$ sitzt. Da wir alle anderen Teilchen ignorieren wollen, ist das zweite Teilchen dann Boltzmannverteilt mit dem Paarwechselwirkungspotential $V(r)$. Daher bekommt man

$$\boxed{g(r) \approx e^{-V(r)/k_B T}.} \quad (5.36)$$

Einsetzen in die Virial-Zustandsgleichung (5.35) ergibt ($\beta = 1/k_B T$)

$$\begin{aligned} \int_0^\infty dr r^3 e^{-\beta V(r)} V'(r) &= \int_0^\infty dr r^3 \left(-\frac{1}{\beta}\right) \frac{d}{dr} \left(e^{-\beta V(r)} - 1\right) \\ &\stackrel{\text{partiell}}{=} 3k_B T \int_0^\infty dr r^2 \left(e^{-\beta V(r)} - 1\right), \end{aligned}$$

wo wir die “-1” in der ersten Zeile eingeführt haben, um bei der partiellen Integration sicherzustellen

dass bei $r \rightarrow \infty$ mit $V(r) \rightarrow 0$ auch die Randterme verschwinden. Dies ergibt dann in (5.35)

$$\frac{pV}{Nk_B T} = 1 - \underbrace{2\pi \int_0^\infty dr r^2 (e^{-\beta V(r)} - 1)}_{\equiv B_2(T)} \rho. \quad (5.37)$$

2. Virialkoeffizient

Dies sind die ersten beiden Terme $\propto \rho^0$ und $\propto \rho^1$ der **Virialentwicklung**, die eine Entwicklung in der Dichte ρ ist.

Für das **Lennard-Jones Potential** $V(r) = V_{\text{LJ}}(r)$ erhalten wir für den 2. Virialkoeffizienten

$$B_2(T) \approx v_0 - \frac{a}{k_B T}$$

mit $v_0 \sim \sigma^3 \sim$ ausgeschlossenes Volumen

$$\frac{aN}{V} \sim \varepsilon \frac{N\sigma^3}{V} \sim$$
 mittlere Wechselwirkungsenergie pro Teilchen (5.38)

Laut Gl. (5.37) stammen positive Anteile an $B_2(T)$ von Regionen mit $V(r) > 0$ (diese sind typischerweise durch die abstoßenden Potentialanteile bestimmt) und negative Anteile von Regionen mit $V(r) < 0$ (diese sind hauptsächlich durch die anziehenden Potentialanteile bestimmt). Entsprechend stammt in (5.38) der erste positive Beitrag v_0 von der Born-Abstoßung für $r < \sigma$ und gibt somit das ausgeschlossene Volumen wieder. Der zweite negative Beitrag stammt von der van-der-Waals Anziehung. Mit dem Volumen pro Teilchen $v \equiv V/N = 1/\rho$ können wir dann die Virialentwicklung (5.37) in folgende Form bringen:

$$\left(p + \frac{a}{v^2} \right) (v - v_0) = k_B T. \quad (5.39)$$

Dies ist genau die bekannte **Van-der-Waals Zustandsgleichung** mit dem **Eigenvolumen** v_0 und dem **Binnendruck** a/v^2 , der angibt, wie die attraktive Wechselwirkung zwischen den Lennard-Jones Teilchen den Druck auf die Wände reduziert.

Reale Van-der-Waals Gase mit der Zustandsgleichung (5.39) zeigen eine Instabilität für

$$k_B T < k_B T_c = \frac{8}{27} \frac{a}{v_0} \sim \varepsilon \quad \text{und} \quad p < p_c = \frac{a}{27v_0^2} \sim \frac{\varepsilon}{v_0},$$

die einer **Kondensation**, d.h. einem **flüssig-gasförmig Phasenübergang 1. Ordnung** entspricht, der dann im Koexistenzbereich mit der üblichen **Maxwell-Konstruktion** beschrieben wird. Der Punkt (p_c, T_c) ist ein **kritischer Punkt**, wo der Phasenübergang kontinuierlich wird. Das typische Phasendiagramm eines einfachen Edelgases wie Argon ist schematisch in Abb. 5.8 gezeigt.

5.4.6 Nachweis von Phasenübergängen

Am **Beispiel des Kondensationsüberganges** eines einfachen einatomigen Gases wie dem Lennard-Jones-Fluid wollen wir noch kurz diskutieren, wie man in einer MD-Simulation einen **Phasenübergang 1. Ordnung** nachweisen und untersuchen kann.

- 1) Zum einen können wir einfach Kurven der Gesamtenergie (oder auch der potentiellen oder kinetischen Energie) $E = E(T)$ als Funktion der Temperatur (oder auch des Drucks) messen, die am Phasenübergang 1. Ordnung eine **Diskontinuität** aufweisen sollten.

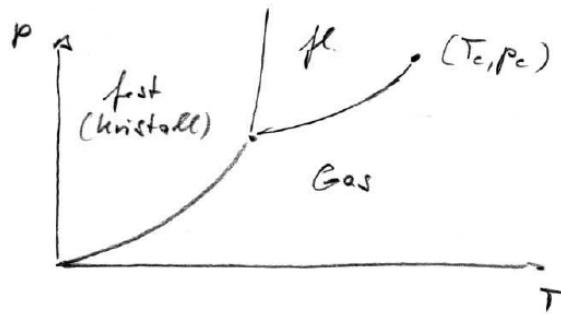
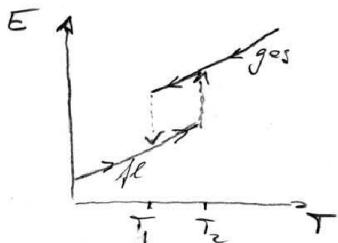
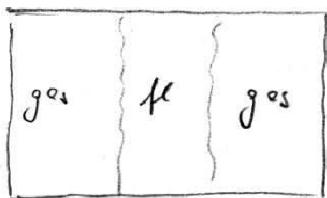


Abbildung 5.8: Schematisches Phasendiagramm von einem Edelgas wie Argon im p - T -Diagramm. Alle Linien sind Phasenübergänge 1. Ordnung, im kritischen Punkt (p_c, T_c) ist der flüssig-Gas Übergang kontinuierlich. Die Van-der-Waals Zustandsgleichung (5.39) beschreibt den flüssig-Gas Übergang.



Ein Problem dabei ist die **Hysterese** an einem Phasenübergang 1. Ordnung. Die eigentliche Übergangstemperatur T_{PT} (die durch die Gleichheit der freien Energien $G(T, p)$ der beiden koexistierenden Phasen bestimmt ist) liegt irgendwo zwischen den Endpunkten der Hystereseschleife, $T_1 < T_{PT} < T_2$ und ist daher schlecht zu bestimmen auf diese Art.

- 2) Eine bessere Methode besteht darin, direkt **koexistierende Phasen** zu simulieren. Bei periodischen Randbedingungen muss man dazu ein System mit zwei Grenzflächen, wie in der Abb. skizziert, präparieren.



Bei einer mikrokanonischen Simulation bei festem E im Koexistenzbereich, können sich die Phasen ineinander umwandeln und die Grenzflächen verschieben. Dadurch wird sich die Temperatur genau auf $T = T_{PT}$ einstellen: Ist z.B. $T < T_{PT}$, wird das System durch Kondensation die flüssige Phase weiter vergrößern; Dadurch wird aber latente Wärme frei, die wiederum die Temperatur erhöht. Entsprechendes passiert bei $T > T_{PT}$, so dass das System nach $T = T_{PT}$ konvergiert, was eine bequeme Bestimmung von T_{PT} erlaubt.

Man kann dann auch eine Druckmessung durchführen und so die Dampfdruckkurve $p = p(T_{PT})$ bestimmen, wenn man verschiedene Energien durchsimuliert. Offensichtlich ist diese Methode aufwendiger bei der Präparation und Initialisierung des Systems.

- 3) Man kann auch die **Paarverteilung** $g(r)$ für verschiedene E messen. Sobald man mehr als ein Maximum findet, ist dies ein Anzeichen, dass man in der flüssigen Phase ist. Auch hier kann es Probleme mit der Hysterese geben. Diese Methode ist außerdem nicht ganz eindeutig: auch ein System ohne thermodynamischen flüssig-Gas Übergang wie z.B. harte Kugeln zeigt einen Übergang von einem zu mehreren Maxima in $g(r)$.

5.5 Kanonische MD Simulation

MD-Simulationen im kanonischen Ensemble benötigen Thermostaten. Wir erläutern den isokinetischen, den Berendsen- und den Nosé-Hoover Thermostaten.

Bisher haben wir die MD-Simulation bei fester erhaltener Gesamtenergie E mikrokanonisch durchgeführt. Man kann aber auch bei fester Temperatur T im **kanonischen Ensemble** simulieren. Dann sollte man eine Methode finden, die zu **Boltzmannverteilten** Energien

$$p(E) \sim e^{-E/k_B T} \quad (5.40)$$

führt. Dazu verwendet man sogenannte **Thermostaten**, die das umgebende Wärmebad, mit dem das System Energie austauschen kann, ersetzen müssen.

5.5.1 Isokinetischer Thermostat

Sehr einfach zu implementieren ist der isokinetische Thermostat, bei dem in *jedem* Zeitschritt alle **Geschwindigkeiten reskaliert** werden

$$\vec{v}_i(t) \rightarrow \alpha(t) \vec{v}_i(t), \quad (5.41)$$

so dass die momentane Temperatur aus (5.2)

$$T(t) = \frac{2}{k_B N_f} \sum_{i=1}^N \frac{\vec{p}_i^2}{2m}$$

($N_f = 3N - 3$ auf Grund der Abspaltung der Schwerpunktsbewegung, die auf Null gesetzt werden sollte) fest auf der vorgegebenen Temperatur T bleibt. Dies wird durch die Wahl

$$\alpha(t) = \left(\frac{T}{T(t)} \right)^{1/2}$$

(5.42)

erreicht. Wir bemerken, dass dadurch auch $\sum_i \vec{v}_i = 0$ erhalten bleibt und keine Schwerpunktsgeschwindigkeit generiert wird. Der Name **isokinetischer Thermostat** röhrt daher, dass durch (5.41) und (5.42) auch die kinetische Energie konstant gehalten wird.

Ein Problem beim einfachen isokinetischen Thermostaten ist jedoch, dass dieser i.Allg. *nicht* zur gewünschten Boltzmann-Verteilung (5.40) der Energien führt.

5.5.2 Berendsen-Thermostat

Einer ganz ähnlichen Idee wie der isokinetische Thermostat folgt der **Berendsen-Thermostat** [8]. Auch dort werden Geschwindigkeiten wie in (5.41) umskaliert, allerdings mit einem Reskalierungsfaktor $\alpha(t)$, der eine zusätzliche **Zeitkonstante** τ_T enthält, die eine zusätzliche **Zeitverzögerung** des Berendsen-Thermostaten beschreibt:

$$\alpha(t) = \left[1 + \frac{\Delta t}{\tau_T} \left(\frac{T}{T(t)} - 1 \right) \right]^{1/2},$$

(5.43)

wobei Δt der Integrationszeitschritt der MD-Simulation ist. Für $\tau_T = \Delta t$ antwortet der Berendsen-Thermostat instantan und ist äquivalent zum isokinetischen Thermostaten (5.42). Für $\tau_T \gg \Delta t$ wird die Antwort allerdings stark verlangsamt.

Auch der Berendsen-Thermostat hat das Problem, dass er i.Allg. *nicht* zur Boltzmann-Verteilung (5.40) der Energien führt.

5.5.3 Nosé-Hoover Thermostat

Eine bessere Lösung stellt der Nosé-Hoover Thermostat dar (nach Shuichi Nosé (1951-2005) [9] und William Graham Hoover [10]), der wirklich eine kanonische Boltzmann-Verteilung (5.40) der Energien gewährleistet.

Beim Nosé-Hoover Thermostaten beschreibt man das Wärmebad durch *eine* neue dynamische Hilfsvariable s mit einer “Masse” Q und einem kanonischen Impuls $p_s = Q\dot{s}$. Die Hilfsvariable $s(t)$ hat die Bedeutung eines **Skalenfaktors für die Zeit**

$$s = \frac{d\tau}{dt},$$

wobei τ eine **virtuelle Zeit** und t die reale Zeit sind. Die Idee ist nun, auch **virtuelle Orte** $\vec{\rho}_i$ und **Impulse** $\vec{\pi}_i$ zu definieren mit einer zugehörigen Hamiltonfunktion $H_{\text{Nose}}(\{\vec{\rho}\}, \{\vec{\pi}\}, s, p_s)$, so dass das mikrokanonische Ensemble für H_{Nose} äquivalent zum kanonischen Ensemble für $H(\{\vec{r}\}, \{\vec{p}\})$ wird.

Die virtuellen Orte können gleich den realen Orten gewählt werden,

$$\vec{\rho}_i = \vec{r}_i.$$

Damit gilt dann für die virtuellen Geschwindigkeiten

$$\dot{\vec{\rho}}_i = \frac{d\vec{\rho}_i}{d\tau} = \frac{1}{s} \frac{d\vec{r}_i}{dt} = \frac{1}{s} \dot{\vec{r}}_i.$$

Die Nosé Lagrange-Funktion lautet

$$\boxed{L_{\text{Nose}} = \underbrace{\sum_i \frac{m}{2} s^2 \dot{\vec{\rho}}_i^2 - V_{\text{tot}}(\{\vec{\rho}\})}_{= L(\{\vec{r}\}, \{\dot{\vec{r}}\})} + \underbrace{\frac{Q}{2} \dot{s}^2 - \frac{C}{\beta} \ln s}_{= L_s(s, \dot{s})}} \quad (5.44)$$

mit einer Konstanten C und $\beta = 1/k_B T$. Der virtuelle Impuls ist dann

$$\vec{\pi}_i = \frac{\partial L_{\text{Nose}}}{\partial \dot{\vec{\rho}}_i} = ms^2 \dot{\vec{\rho}}_i = ms \dot{\vec{r}}_i = s \vec{p}_i$$

und der zu s konjugierte Impuls

$$p_s = \frac{\partial L_{\text{Nose}}}{\partial \dot{s}} = Q\dot{s}.$$

Damit erhält man den Nosé-Hamiltonian

$$\boxed{H_{\text{Nose}} = \underbrace{\sum_i \frac{1}{2m} \frac{\vec{\pi}_i^2}{s^2} + V_{\text{tot}}(\{\vec{\rho}\})}_{= H(\{\vec{r}\}, \{\vec{p}\})} + \frac{p_s^2}{2Q} + \frac{C}{\beta} \ln s.} \quad (5.45)$$

Man kann nun zeigen, dass für das kanonische Mittel einer Observablen O gilt:

$$\begin{aligned} \underbrace{\langle O(\{\vec{r}\}, \{\vec{p}\}) \rangle_{NVT}}_{\text{kanonisches Mittel mit } H} &= \underbrace{\langle O(\{\vec{\rho}\}, \{\vec{\pi}/s\}) \rangle_{\text{Nose}}}_{\text{mikrokanonisches Mittel mit } H_{\text{Nose}}} \\ &= \text{Nosé-Zeitmittel in } \left\{ \begin{array}{c} \text{realer} \\ \text{virtueller} \end{array} \right\} \text{Zeit,} \\ &\text{wenn } C = \left\{ \begin{array}{c} N_f \\ N_f + 1 \end{array} \right\} \text{ gewählt wird.} \end{aligned}$$

Beweis:

Ein vollständiger Beweis findet sich in Frenkel [1], wir zeigen hier lediglich $Z_{mk,\text{Nose}}(E, N, V) = \text{const} Z_k(T, N, V)$ für die Zustandssummen. Das heißt, dass die mikrokanonische Zustandssumme $Z_{mk,\text{Nose}}(E, N, V)$ (in virtuellen Größen berechnet) mit dem Nosé-Hamiltonian H_{Nose} mit der kanonischen Zustandssumme $Z_k(T, N, V)$ im NVT-Ensemble bis auf eine Konstante übereinstimmt, wenn $C = N_f + 1 = 3N + 1$ gewählt wird. Dazu bilden wir zunächst die mikrokanonische Zustandssumme in den virtuellen Impulsen $\vec{\pi}_i = s\vec{p}_i$ und $\vec{r}_i = \vec{r}_i$ und den beiden Wärmebadkoordinaten s und p_s :

$$\begin{aligned} Z_{mk,\text{Nose}}(E, N, V) &= \int ds \int dp_s \int d^{3N} \vec{\pi} \int d^{3N} \vec{p} \delta [H_{\text{Nose}}(\{\vec{p}\}, \{\vec{\pi}\}, s, p_s) - E] \\ &\stackrel{(5.45)}{=} \int ds \int dp_s \int d^{3N} \vec{p} \int d^{3N} \vec{r} s^{3N} \delta \left[H(\{\vec{r}\}, \{\vec{p}\}) + \underbrace{\frac{p_s^2}{2Q} + Ck_B T \ln s - E}_{= f(s)} \right] \end{aligned}$$

Nun benutzen wir die Regel zum Variablenwechsel in einer δ -Funktion,

$$\int ds \delta(f(s)) = \int ds \frac{\delta(s - s_0)}{|f'(s)|}, \quad \text{wenn } f(s_0) = 0 \text{ einzige Nullstelle.}$$

Hier gilt

$$\begin{aligned} f(s_0) = 0 &\iff s_0 = \exp \left(\frac{1}{Ck_B T} \left(E - H - \frac{p_s^2}{2Q} \right) \right) \\ f'(s) &= Ck_B T \frac{1}{s} \end{aligned}$$

und damit

$$\begin{aligned} Z_{mk,\text{Nose}}(E, N, V) &= \int ds \int dp_s \int d^{3N} \vec{p} \int d^{3N} \vec{r} s^{3N+1} \frac{1}{Ck_B T} \delta \left[s - \exp \left(\frac{1}{Ck_B T} \left(E - H - \frac{p_s^2}{2Q} \right) \right) \right] \\ &= \int dp_s \int d^{3N} \vec{p} \int d^{3N} \vec{r} \frac{1}{Ck_B T} \exp \left(\frac{3N+1}{Ck_B T} \left(E - H(\{\vec{r}\}, \{\vec{p}\}) - \frac{p_s^2}{2Q} \right) \right). \end{aligned}$$

Wenn wir $C = 3N + 1 = N_f + 1$ wählen, folgt

$$\begin{aligned} Z_{mk,\text{Nose}}(E, N, V) &= \left[\int d^{3N} \vec{p} \int d^{3N} \vec{r} \exp \left(-\frac{1}{k_B T} H(\{\vec{r}\}, \{\vec{p}\}) \right) \right] \times \\ &\quad \left[\int dp_s \frac{1}{Ck_B T} \exp \left(\frac{1}{k_B T} \left(E - \frac{p_s^2}{2Q} \right) \right) \right] \\ &= Z_k(T, N, V) \times \text{const} \end{aligned}$$

wie behauptet.

Insgesamt folgt aus der Dynamik virtueller Größen mit H_{Nose} folgende Dynamik für die *realen* Größen \vec{r}, \vec{p}, t und die neue Hilfsgröße $\xi \equiv p_s/Q$

$$\begin{aligned} \dot{\vec{r}}_i &= \frac{\vec{p}_i}{m} \\ \dot{\vec{p}}_i &= -\vec{\nabla}_{\vec{r}_i} V_{\text{tot}} - \xi \vec{p}_i \\ \dot{\xi} &= \left(\sum_i \frac{\vec{p}_i^2}{m} - \frac{C}{\beta} \right) \frac{1}{Q}, \end{aligned} \tag{5.46}$$

wobei $C = N_f$ gewählt werden muss. Dies ist der **Nosé-Hoover Thermostat**, wobei der zusätzliche Freiheitsgrad ξ die Kopplung an das Wärmebad ausmacht. Der Nosé-Hoover Thermostat hat folgende Eigenschaften:

- (5.46) zeigt, dass die E_{kin} nach

$$E_{kin} = \sum_i \frac{\vec{p}_i^2}{2m} = \frac{C}{2\beta} = \frac{N_f}{2} k_B T$$

relaxiert.

- Der Gesamtimpuls $\sum_i \vec{p}_i = 0$ ist erhalten für Paarkräfte. Daher ist $N_f = 3N - 3$.
- Wenn Q groß (klein), reagiert das Wärmebad langsam (schnell).
- Mit (5.46) gilt Zeitmittel = kanonisches Mittel = Nosé-Zeitmittel und der Nosé-Hoover Thermostat führt wirklich zu einer Boltzmann-Verteilung der Energien.

5.6 Literaturverzeichnis Kapitel 5

- [1] D. Frenkel und B. Smit. *Understanding Molecular Simulation*. 2nd. Orlando, FL, USA: Academic Press, Inc., 2001.
- [2] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [3] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [4] B. Alder und T. Wainwright. *Phase Transition for a Hard Sphere System*. J. Chem. Phys. **27** (1957), 1208–1211.
- [5] L. Verlet. *Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*. Phys. Rev. **159** (1967), 98–103.
- [6] M. Tuckerman, G. J. Martyna und B. J. Berne. *Reversible multiple time scale molecular dynamics*. J. Chem. Phys. **97** (1992), 1990–2001.
- [7] L. Schulman. *Techniques and Applications of Path Integration*. Wiley, 1996.
- [8] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, a. DiNola und J. R. Haak. *Molecular dynamics with coupling to an external bath*. J. Chem. Phys. **81** (1984), 3684–3690.
- [9] S. Nosé. *A unified formulation of the constant temperature molecular dynamics methods*. J. Chem. Phys. **81** (1984), 511–519.
- [10] W. G. Hoover. *Canonical dynamics: Equilibrium phase-space distributions*. Phys. Rev. A **31** (1985), 1695–1697.

5.7 Übungen Kapitel 5

1. 2D Lennard-Jones-Fluid

Schreiben Sie eine Molekulardynamik Simulation für N identische Teilchen der Masse $m = 1$ mit paarweiser Lennard-Jones-Wechselwirkung

$$V(r) = 4 \left[\left(\frac{1}{r} \right)^{12} - \left(\frac{1}{r} \right)^6 \right] \quad (5.47)$$

(d.h. Längen werden in Einheiten von σ und Energien bzw. $k_B T$ in Einheiten von ϵ gemessen). Benutzen Sie periodische Randbedingungen in einem zweidimensionalen System der Größe $A = L \times L$. Verwenden Sie einen Cutoff $r_c = L/2$ bei der Kraftberechnung. Verwenden Sie den Verletz-Algorithmus mit Zeitschritt $h = 0.01$ (oder kleiner bei hohen Temperaturen) zur Integration.

a) Initialisierung:

Setzen Sie die $N = 16$ Teilchen zu Beginn auf Plätze $\vec{r}(0) = \frac{1}{8}(1+2n, 1+2m)L$ mit $n, m = 0, \dots, 3$ in der Box $[0, L] \times [0, L]$. Wählen Sie die Anfangsgeschwindigkeiten so, dass $\sum_{i=1}^N \vec{v}_i(0) = 0$, d.h. dass die Schwerpunktsgeschwindigkeit zu Beginn gleich 0 ist. Schreiben Sie das Programm so, dass Sie die Geschwindigkeiten umskalieren können, um bei einer gegebenen "Anfangstemperatur" $T(t=0)$ starten zu können.

b) Messung/Äquilibrierung:

Berechnen Sie die Schwerpunktsgeschwindigkeit $\frac{1}{N} \sum_{i=1}^N \vec{v}_i$ als Funktion der Zeit. Berechnen Sie die Temperatur $T(t)$ als Funktion der Zeit.

Berechnen Sie die potentielle Energie $E_{\text{pot}}(t) = \sum_{i < j=1}^N V(|\vec{r}_i - \vec{r}_j|)$ und die kinetische Energie $E_{\text{kin}}(t) = \sum_{i=1}^N \frac{1}{2} \vec{v}_i^2$ als Funktion der Zeit.

Nach wie vielen Zeitschritten äquilibriert Ihr System?

c) Messung:

Nachdem Ihr System äquilibriert ist, messen Sie die Temperatur T und die Paarkorrelationsfunktion $g(r)$. Dazu führen Sie nach der Äquilibrierungsphase eine Mittelung über 10^4 bis 10^6 Zeitschritte durch (abhängig davon, wie lange Sie warten möchten).

Messen Sie diese Größen für $N = 16$, $L = 8$ und bei zwei verschiedenen "Anfangstemperaturen" $T(0) = 2$ und $T(0) = 0.2$.

Versuchen Sie, dass System in einen flüssigen und/oder festen Zustand zu bringen, indem Sie für die Systemgröße L schrittweise erniedrigen für festes $N = 16$ (Dichteerhöhung). Untersuchen Sie Anfangstemperaturen $T(0) = 0.5$ und $T(0) = 2$. Benutzen Sie die Paarkorrelationsfunktion, um den flüssigen (oder festen) Zustand zu erkennen.

d) Thermostat:

Implementieren Sie einen einfachen isokinetischen Thermostaten (Geschwindigkeitsumskalierung). Untersuchen Sie wieder das System $N = 16$ und $L = 8$ und senken Sie die Temperatur T von $T = 1$ in Schritten $\Delta T = 0.1$ bei einer Dichte $\rho = 0.5$. Wann bildet sich eine flüssige oder feste Phase? Sehen Sie Phasenkoexistenz? (Benutzen Sie wieder die Paarkorrelationsfunktion, um den flüssigen oder festen Zustand zu erkennen.)

6 Partielle Differentialgleichungen

Literatur zu diesem Teil:

Dies ist ein zentrales Thema in der numerischen Mathematik mit vielen physikalischen Anwendungen. Es gibt umfangreiche Literatur, sowohl von Seiten der numerischen Mathematik, z.B. Numerical Recipes [1, 2], als auch von Seiten der Computerphysik, z.B. Fitzpatrick [3], Koonin/Meredith [4], Gould/Tobochnik [5].

Die numerische Lösung **partieller Differentialgleichungen** ist ein wichtiges Gebiet der numerischen Mathematik mit zahlreichen Anwendungen in der Physik von elektromagnetischen, über mechanische zu hydrodynamischen und quantenmechanischen Problemen (und wahrscheinlich vieles mehr). Wir können dieses große Gebiet der numerischen Mathematik hier nur sehr unzureichend andiskutieren. Wir beginnen mit einer exemplarischen Diskussion von vier in der Physik wichtigen Typen/Beispielen von partiellen DGLn mit einfachen Lösungsverfahren durch Diskretisierung (finite Differenzen).

- 1) Die **Poisson-Gleichung** der Elektrostatik für das Potential $\phi(\vec{r})$ eine Ladungverteilung $\rho(\vec{r})$

$$\Delta\phi = -\rho(\vec{r}) \quad (6.1)$$

im Volumen V . Wir haben zur Diskussion der numerischen Lösung hier eine besonders einfache Form der Poisson-Gleichung gewählt, um im Weiteren Schreibarbeit zu sparen; gegenüber der Poisson-Gleichung der E-Statik in SI-Einheiten $\Delta\phi = -\rho(\vec{r})/\varepsilon_0$ fehlt der Faktor $1/\varepsilon_0$; gegenüber der Poisson-Gleichung der E-Statik in Gauß-Einheiten $\Delta\phi = -4\pi\rho(\vec{r})$ fehlt der Faktor 4π ; in vielen Büchern wird die numerische Lösung auch einfach an Hand von $\Delta\phi = \rho(\vec{r})$ ohne das Vorzeichen diskutiert. Die partielle DGL (6.1) ist **elliptisch** und es handelt sich um ein **statisches Randwertproblem**:

Die Randbedingungen für ϕ (Dirichlet) oder $\vec{\nabla}\phi$ (Neumann) sind auf dem **Rand** ∂V gegeben.

- 2) Die **Wellengleichung** für eine Funktion $u(x, t)$

$$\partial_t^2 u = v^2 \partial_x^2 u \quad (6.2)$$

Diese partielle DGL ist **hyperbolisch** und es handelt sich um ein **Anfangswertproblem** (Cauchy-Problem):

$u(x, 0)$ und $\dot{u}(x, 0)$ sind bei $t = 0$ gegeben, und wir fragen nach der **Zeitentwicklung**. Wenn $x \in [0, L]$ ein *endliches* Intervall (z.B. bei stehender Welle) müssen auch **Randbedingungen** $u(0, t)$ und $u(L, t)$ für $t \geq 0$ gegeben sein.

- 3a) Die **Diffusionsgleichung** z.B. für ein Konzentrationsfeld $c(x, t)$

$$\partial_t c = D \partial_x^2 c \quad (6.3)$$

Diese partielle DGL ist **parabolisch**, und es handelt sich ebenfalls um ein **Anfangswertproblem**.

- 3b) Die **Schrödinger-Gleichung** für die quantenmechanische 1-Teilchen Wellenfunktion $\psi(x, t)$

$$i\hbar \partial_t \psi = \hat{H} \psi \quad (6.4)$$

mit einem Hamiltonoperator $\hat{H} = -(\hbar^2/2m)\partial_x^2 + V(x)$ in Ortsdarstellung. Diese partielle DGL entspricht (für den freien Fall $V = 0$) Diffusion in “komplexer Zeit”, und es handelt sich ebenfalls um ein **Anfangswertproblem**.

Der numerische Ansatz ist in **allen** Fällen das **Differenzverfahren**:

Wir diskretisieren die Ortsableitungen auf einem (**mehrdimensionalen**) **Gitter**, der einfachste Diskretisierungsansatz für die t -Abhängigkeit in 2) und 3) ist die Euler-Diskretisierung, siehe auch Kapitel 4.2 zu gewöhnlichen DGLn. Eine wichtige Frage dabei wird die numerische **Stabilität** der Verfahren betreffen, die entscheidend von der Diskretisierung abhängen kann.

6.1 Poisson-Gleichung

Die eindimensionale Poisson-Gleichung mit Randbedingungen kann numerisch durch Schussverfahren und iterative Verfahren wie Jacobi-Iteration, Gauß-Seidel-Iteration oder Relaxationsverfahren gelöst werden. Für die zweidimensionale Poisson-Gleichung diskutieren wir ebenfalls Iterationsverfahren.

Wir wollen zuerst die **Poisson-Gleichung** für das elektrostatische Potential $\phi(\vec{r})$ einer Ladungsverteilung $\rho(\vec{r})$

$$\Delta\phi = -\rho(\vec{r})$$

(wieder ohne Faktor $1/\varepsilon_0$, siehe oben) mit Randbedingungen numerisch lösen. Wir beschränken uns auf die Fälle **einer** und **zwei Raumdimensionen**.



Abbildung 6.1: Links: Siméon Denis Poisson (1781-1840), französischer Physiker und Mathematiker. Mitte: Johann Peter Gustav Lejeune Dirichlet (1805-1859), deutscher Mathematiker. Rechts: Carl Neumann (1832-1925), deutscher Mathematiker. (Quelle: Wikipedia).

6.1.1 1D Poisson-Gleichung

Zunächst beginnen wir mit der **eindimensionalen Poisson-Gleichung**

$$\partial_x^2\phi = -\rho(x) \tag{6.5}$$

wobei es verschiedene Sorten von **Randbedingungen** gibt:

Dirichlet :	links $\phi(x_1) = \phi_l$	rechts $\phi(x_2) = \phi_r$
Neumann :	$\partial_x\phi(x_1) = -E_l$	$\partial_x\phi(x_r) = -E_r$
oder gemischt	$\alpha_{r,l}\phi(x_{r,l}) + \beta\partial_x\phi(x_{r,l}) = \gamma_{r,l}$	

Die eindimensionale Poissons-Gleichung (6.5) ist **keine** partielle DGL (die Funktion $\phi(x)$ hängt nur von einer Koordinate x ab), sondern ein **Randwertproblem für gewöhnliche DGL 2. Ordnung**. Wir werden an Hand der eindimensionalen Poisson-Gleichung aber einige wichtige numerische Lösungsalgorithmen (Schussverfahren, finite Differenzen, usw. siehe unten) einführen.

Außerdem treten **Randwertprobleme für DGLn 2. Ordnung** der allgemeinen Form

$$\boxed{\psi'' = f(x, \psi, \psi')} \quad (6.6)$$

mit Randbedingungen $B_1(x_1, \psi(x_1), \psi'(x_1)) = 0$ und $B_2(x_2, \psi(x_2), \psi'(x_2)) = 0$ sehr häufig in der Physik auf und für diese kann man die *gleichen numerischen Lösungsalgorithmen* wie für die eindimensionale Poissons-Gleichung (6.5) benutzen. Dies gilt insbesondere, wenn die DGL (6.6) eine *lineare* DGL ist. Wir wollen diesen Punkt mit einigen **Beispielen** für Randwertprobleme der Form (6.6) aus der Physik belegen:

- **Stationäre Eigenwertprobleme** nach Abseparation der Zeit:

- Stationäre Schrödingergleichung in 1D

$$-\frac{\hbar^2}{2m}\psi''(x) + V(x)\psi(x) = E\psi(x) \quad (= i\hbar\partial_t\psi)$$

nach Separationsansatz $\psi(x, t) = \psi(x)e^{-iEt/\hbar}$ mit Randbedingungen, z.B. für ein Teilchen in einer Box $\psi(x_1) = 0$ und $\psi(x_2) = 0$.

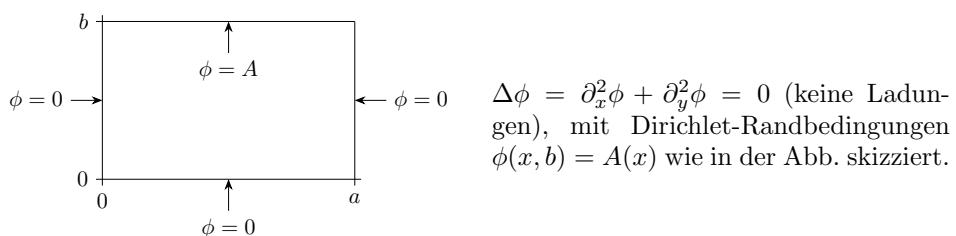
- Normalmoden von Wellengleichungen

$$v\partial_x^2 u = -\omega^2 u \quad (= \partial_t^2 u)$$

nach Separationsansatz $u(x, t) = u(x)e^{i\omega t}$ mit vorgegebenen Randbedingungen, die auch die Normalmoden erfüllen müssen.

- Reduktion höherdimensionaler Poisson-Gleichung oder Eigenwertprobleme durch Separation oder Fouriertrafo:

- Radiale Schrödingergleichung (H-Atom) für $\phi_{nl}(r)$ nach Separationsansatz $\psi(r, \varphi, \vartheta) = \phi_{nl}(r)Y_{lm}(\varphi, \vartheta)$. Randbedingungen für $\phi_{nl}(r)$ sind dann Normierbarkeit, d.h. $\phi_{nl}(\infty) = 0$. Der Startwert bei $r = 0$ kann zunächst beliebig gewählt werden und wird durch die Normierung nachher festgelegt.
- Poisson-Gleichung, z.B. folgendes Randwertproblem in zwei Dimensionen:



Nach Fouriertrafo bezüglich der x -Koordinate, $\tilde{\phi}(k_x, y) = \int dx e^{-ik_x x} \phi(x, y)$ oder $\phi(x, y) = \int \frac{dk_x}{2\pi} \tilde{\phi}(k_x, y) e^{ik_x x}$, bekommen wir eine Gleichung

$$\partial_y^2 \tilde{\phi} - k_x^2 \tilde{\phi} = 0$$

also wieder ein eindimensionales Problem der Form (6.6). Die Randbedingungen haben hier zwei Effekte. Zunächst führen sie dazu dass nur diskrete $k_{x,n} = n\pi/a$ in der Fourierdarstellung von $\phi(x, y)$ vorkommen, also gilt $\phi(x, y) = \sum_n \phi_n(y) \sin(k_{x,n}x)$. Zum anderen müssen wir das verbleibende eindimensionale Problem $\partial_y^2 \phi_n - k_{x,n}^2 \phi_n = 0$ mit Randbedingungen $\phi_n(0) = 0$ und $\phi_n(b) = A_n$ lösen, so dass $A(x) = \sum_n A_n \sin(k_{x,n}x)$ am oberen Rand gilt.

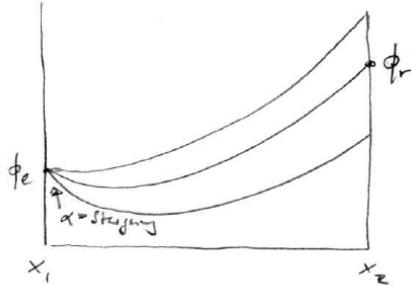
Nun wollen wir uns mit einigen numerischen Lösungsmethoden für die eindimensionale Poissons-Gleichung (6.5) vertraut machen. Wir werden im Folgenden Dirichlet-Randbedingungen $\phi(x_1) = \phi_l$ und $\phi(x_2) = \phi_r$ stellen. Andere Arten von Randbedingungen funktionieren analog.

Schussverfahren

Dort lösen wir zuerst das gewöhnliche (linksseitige) Anfangswertproblem

$$\partial_x^2 \phi(x) = -\rho(x) \quad \text{mit } \phi(x_1) = \phi_l \quad \text{und} \quad \partial_x \phi(x_1) = \alpha$$

numerisch mit den Verfahren aus Kapitel (4) (z.B. Euler oder Runge-Kutta), wobei α unser **Schussparameter** sein wird. Das Ergebnis ist eine numerisch definierte Funktion $f(\alpha) = \text{Lösung } \phi(x_2)$, die uns die Lösung am rechten Rand als Funktion der Steigung α am linken Rand gibt.



Die **Idee** ist nun folgende: Wir ändern den Schussparameter α am linken Rand x_1 , bis die Lösung am rechten Rand x_2 die Randbedingungen erfüllt, also bei Dirichletschen Randbedingungen den gewünschten Wert ϕ_r trifft.

Das bedeutet, wir müssen die Gleichung

$$f(\alpha) = \phi_r$$

numerisch lösen, um die zweite Dirichlet-Randbedingung $\phi(x_2) = \phi_r$ zu erfüllen. Dazu verwenden wir numerische Verfahren zum Lösen beliebiger Gleichungen wie z.B. das Newton-Raphson Verfahren, die wir später in Kapitel (7.2) kennenlernen werden.

Diskretisierung in x und iterative Verfahren

Mit Hilfe einer Diskretisierung der x -Koordinate wird aus der linearen DGL eine **lineare Finite-Differenzen-Gleichung**. Diese kann mit numerischen Lösungsmethoden für lineare Gleichungssysteme bearbeitet werden.

Das heißt, wir diskretisieren mit einer **Gitterkonstanten** Δ die x -Koordinate, so dass

$$\begin{aligned} \phi(x) &\rightarrow \phi_i \equiv \phi(i\Delta) & \rho(x) &\rightarrow \rho_i \\ x_1 &\rightarrow i = 0 & x_2 &\rightarrow i = \frac{x_2 - x_1}{\Delta} \equiv N \end{aligned}$$

Dann wird aus der Poisson-Gleichung (6.5)

$$\partial_x^2 \phi \approx \frac{\phi_{i-1} - 2\phi_i + \phi_{i+1}}{\Delta^2} = -\rho_i \quad (6.7)$$

im Inneren $i = 1, \dots, N - 1$, wobei wir die diskretisierte Darstellung (3.4) der 2. Ableitung aus Kapitel (3) verwendet haben. Am Rand gilt

$$\phi_0 = \phi(x_1) = \phi_l \quad \text{und} \quad \phi_N = \phi(x_2) = \phi_r \quad (6.8)$$

Die Gleichung (6.7) ist eine **lineare Gleichung**, die wir auch in Matrixform schreiben können

$$\underline{\underline{A}} \cdot \vec{\phi} = \vec{r} \quad (6.9)$$

mit

$$\begin{aligned} \underline{\underline{A}} &= \frac{1}{\Delta^2} \begin{pmatrix} -2 & 1 & & 0 \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \\ 0 & 1 & -2 & 1 \\ & & 1 & -2 \end{pmatrix} \quad \text{tridiagonale } (N-1) \times (N-1) \text{ Matrix} \\ \vec{\phi} &= \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{pmatrix} \quad \text{und} \quad \vec{r} = - \underbrace{\begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{N-1} \end{pmatrix}}_{\text{Quellen}} - \frac{1}{\Delta^2} \underbrace{\begin{pmatrix} \phi_l \\ 0 \\ \vdots \\ \phi_r \end{pmatrix}}_{\text{Randbed.}} \end{aligned}$$

Dieses lineare Gleichungssystem lässt sich natürlich mit den numerischen Standardmethoden für lineare Gleichungssysteme lösen, z.B. **Gauß-Jordan-Eliminierung** (in $\mathcal{O}(N^3)$ Operationen) oder durch **LU-Zerlegung** (benötigt für Tridiagonalmatrix nur $\mathcal{O}(N)$ Operationen). Es gibt aber auch spezielle **iterative Verfahren**, die gerade bei der Poisson-Gleichung oft zum Einsatz kommen: Dies sind die **Jacobi-Iteration**, die **Gauß-Seidel-Iteration** oder **Relaxationsverfahren**. Diese werden im Folgenden besprochen.

Jacobi-Iteration

Die Idee und Vorgehensweise der iterativen Verfahren ist dabei immer folgende:

- (i) Schreibe (6.7) als **Fixpunktgleichung**

$$\vec{\phi} = \vec{F}(\vec{\phi}) \quad (6.10)$$

d.h. mit dem zu bestimmenden Potentialvektor $\vec{\phi}$ sowohl auf der linken als auch auf der rechten Seite und einer Funktion \vec{F} : So kann (6.7) als

$$\phi_i = \frac{1}{2} (\phi_{i-1} + \phi_{i+1}) + \frac{1}{2} \Delta^2 \rho_i \quad (6.11)$$

geschrieben werden

- (ii) Löse die Fixpunktgleichung **iterativ**:

Anfangswert $\vec{\phi}^{(0)}$ raten $\rightarrow \vec{\phi}^{(1)} = \vec{F}(\vec{\phi}^{(0)}) \rightarrow \vec{\phi}^{(2)} = \vec{F}(\vec{\phi}^{(1)}) \rightarrow \dots \rightarrow \vec{\phi}^{(n+1)} = \vec{F}(\vec{\phi}^{(n)}) \rightarrow \dots$
Für (6.11) lautet die Rekursion

$$\phi_i^{(n+1)} = \frac{1}{2} (\phi_{i-1}^{(n)} + \phi_{i+1}^{(n)}) + \frac{1}{2} \Delta^2 \rho_i \quad (6.12)$$

im Inneren $i = 1, \dots, N-1$, während die Randbedingungen (6.8) ϕ_0 und ϕ_N festlegen.

Die Iteration (6.12) heißt **Jacobi-Iteration**. Der Fixpunkt der Iteration ist die gesuchte Lösung von (6.10). Also iterieren wir so lange, bis sich „nichts mehr ändert“, z.B. bis

$$|\vec{\phi}^{(n+1)} - \vec{\phi}^{(n)}| < \varepsilon \quad (6.13)$$

mit einem Genauigkeitsziel ε . Dabei kann man zeigen (Beweis später in Kapitel 7.1 mit dem Banachschen Fixpunktsatz), dass die Jacobi-Iteration immer **konvergiert**.

Gauß-Seidel-Iteration

Die Gauß-Seidel Iteration ist sehr ähnlich, der einzige Unterschied zur Jacobi-Iteration besteht darin, dass man für den Nachbar ϕ_{i-1} auf der rechten Seite in (6.12) schon die nächste Iteration benutzt:

$$\boxed{\phi_i^{(n+1)} = \frac{1}{2} (\phi_{i-1}^{(n+1)} + \phi_{i+1}^{(n)}) + \frac{1}{2} \Delta^2 \rho_i} \quad (6.14)$$

Das funktioniert, wenn $\phi_i^{(n+1)}$ immer in der Reihenfolge $i = 1, 2, \dots, N - 1$ mit (6.14) berechnet werden.

Auch hier gilt, dass der Fixpunkt der Iteration die gesuchte Lösung darstellt und das wir mit einem Genauigkeitsziel ε iterieren bis eine Bedingung (6.13) erfüllt ist. Auch beim Gauß-Seidel-Verfahren kann man (mit Banachschen Fixpunktsatz) zeigen, dass es immer **konvergiert**.

Relaxationsverfahren

Bei Relaxationsverfahren "mischen" wir einer Iteration nach Jacobi oder Gauß-Seidel noch die "alte" Lösung bei:

$$\boxed{\phi_i^{(n+1)} = (1 - \sigma)\phi_i^{(n)} + \sigma\phi_i^{(n+1)} \text{ (Jacobi/Gauß-Seidel)}} \quad (6.15)$$

Dabei hängt die optimale Wahl des Parameters σ vom Problem ab, oft stellt sich $\sigma \approx 1.5$ als gute Wahl heraus. **Konvergenz** zum Fixpunkt ist gewährleistet für alle $0 < \sigma < 2$.

6.1.2 2D Poisson-Gleichung

Nun betrachten wir die **zweidimensionale Poisson-Gleichung**

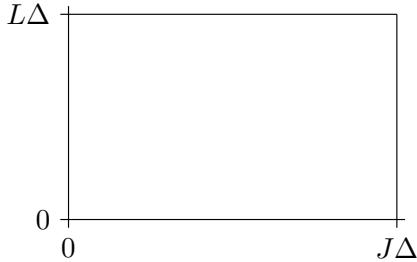
$$\boxed{\Delta\phi = \partial_x^2\phi + \partial_y^2\phi = -\rho(x, y)} \quad (6.16)$$

auf einer Fläche V , wobei es wieder verschiedene Sorten von **Randbedingungen** gibt: Wir können auf der Berandung ∂V der Fläche entweder ϕ vorgeben (Dirichlet) oder die Normalenableitung $\vec{\nabla}_n\phi$ (Neumann) bzw. Mischungen dieser beiden Randbedingungen. Die zweidimensionale Poisson-Gleichung ist eine echte **elliptische partielle DGL 2. Ordnung**.

In zwei Raumdimensionen benutzt man Verallgemeinerungen des **Diskretisierungsverfahrens**. Wir diskretisieren hier einfach in ein Quadratgitter mit Gitterkonstante Δ

$$\phi(x, y) \rightarrow \phi_{j,l} \equiv \phi(j\Delta, l\Delta) \quad \rho(x, y) \rightarrow \rho_{j,l}$$

Wir können dann z.B. für ein Rechteck V Dirichletrandbedingungen vorgeben:



Im Inneren von V haben wir Indizes $j = 1, \dots, J - 1$ und $l = 1, \dots, L - 1$. Auf dem Rand ∂V ist dann $j = 0$, $j = J$, $l = 0$, oder $l = L$ und wir geben bei Dirichletschen Randbedingungen dort Potentialwerte fest vor.

Dies führt dann wieder auf eine **lineare Finite-Differenzen-Gleichung**

$$\partial_x^2\phi + \partial_y^2\phi \approx \frac{\phi_{j+1,l} - 2\phi_{j,l} + \phi_{j-1,l}}{\Delta^2} + \frac{\phi_{j,l+1} - 2\phi_{j,l} + \phi_{j,l-1}}{\Delta^2} = -\rho_{j,l} \quad (6.17)$$

aus der man eine **Fixpunktgleichung**

$$\phi_{j,l} = \frac{1}{4} (\phi_{j+1,l} + \phi_{j-1,l} + \phi_{j,l+1} + \phi_{j,l-1}) + \frac{1}{4} \Delta^2 \rho_{j,l} \quad (6.18)$$

machen kann, die dann im Inneren von V gelten soll, während man für unser Beispiel einer rechteckigen Fläche V Potentialwerte $\phi_{0,l}$, $\phi_{J,l}$, $\phi_{j,0}$ und $\phi_{j,L}$ auf dem Rand vorgeben würde bei Dirichlet Randbedingungen.

(6.17) ist wieder ein **lineares Gleichungssystem** für einen $(J-1) \times (L-1)$ -dimensionalen Potentialvektor $\phi_{j,l}$. Wir können die gleichen numerischen Lösungsmethoden wie in einer Dimension benutzen, also Standardmethoden wie Gauß-Jordan-Eliminierung oder LU-Zerlegung oder aber auch die gleichen **iterativen Verfahren**, z.B. die **Jacobi-Iteration**

$$\phi_{j,l}^{(n+1)} = \frac{1}{4} (\phi_{j+1,l}^{(n)} + \phi_{j-1,l}^{(n)} + \phi_{j,l+1}^{(n)} + \phi_{j,l-1}^{(n)}) + \frac{1}{4} \Delta^2 \rho_{j,l} \quad (6.19)$$

mit $\phi_{j,l}$ fest auf dem Rand. Bei der **Gauß-Seidel-Iteration** kann man das Gitter jetzt in der Reihefolge $j = 1, 2, \dots, J-1$ und für jedes j dann $l = 1, \dots, L$ durchlaufen. Dann kann man für zwei Werte auf der rechten Seite immer bereits die nächste Iteration verwenden:

$$\phi_{j,l}^{(n+1)} = \frac{1}{4} (\phi_{j+1,l}^{(n)} + \phi_{j-1,l}^{(n+1)} + \phi_{j,l+1}^{(n)} + \phi_{j,l-1}^{(n+1)}) + \frac{1}{4} \Delta^2 \rho_{j,l} \quad (6.20)$$

Man kann die Operationen auf der rechten Seite der Jacobi-Iteration (6.19) oder der Gauß-Seidel-Iteration (6.20) [und genauso in einer Dimension in (6.12) oder in (6.14)] auch als *lokale Mittelwertbildung* interpretieren. In Abwesenheit von Ladungen stellt dies genau die **Mittelwerteigenschaft harmonischer Funktionen** (den Lösungen der Laplace-Gleichung $\Delta\phi = 0$) sicher:

$$\begin{aligned} & \text{Sei } K(\vec{r}, a) \text{ Kugel mit Radius } a \text{ um } \vec{r} \text{ mit Oberfläche } 4\pi a^2 \\ & \text{Dann gilt } \phi(\vec{r}) = \frac{1}{4\pi a^2} \int_{\partial K(\vec{r}, a)} d\tilde{f} \phi(\tilde{\vec{r}}) = \frac{1}{4\pi} \int_{\partial K(0,1)} d\tilde{f} \phi(\vec{r} + a\tilde{\vec{r}}) \end{aligned} \quad (6.21)$$

Beweis:

Wir definieren die Hilfsfunktion

$$\tilde{\phi}(a) \equiv \frac{1}{4\pi} \int_{\partial K(0,1)} d\tilde{f} \phi(\vec{r} + a\tilde{\vec{r}})$$

Für deren Ableitung nach a gilt

$$\begin{aligned} \tilde{\phi}'(a) &= \frac{1}{4\pi} \int_{\partial K(0,1)} d\tilde{f} \tilde{\vec{r}} \cdot \vec{\nabla} \phi(\vec{r} + a\tilde{\vec{r}}) = \frac{1}{4\pi} \int_{\partial K(0,1)} d\tilde{f} \cdot \vec{\nabla} \phi(\vec{r} + a\tilde{\vec{r}}) \\ &\stackrel{\text{Gauß}}{=} \frac{1}{4\pi} \int_{K(0,1)} d\tilde{V} \underbrace{\Delta \phi(\vec{r} + a\tilde{\vec{r}})}_{=0} = 0 \end{aligned}$$

$\tilde{\vec{f}} = \tilde{f}\tilde{\vec{r}}$ mit $|\tilde{\vec{r}}| = 1$ ist der Normalenvektor auf der Kugeloberfläche $K(0, 1)$). Daher ist $\tilde{\phi}(a) = \text{const}$ und aus Stetigkeitsgründen $\tilde{\phi}(a) = \lim_{a \rightarrow 0} \tilde{\phi}(a) = \phi(\vec{r})$. Damit ist die Behauptung gezeigt.

Allgemein lässt sich abschließend zur Lösung der Poisson-Gleichung in höheren Dimensionen folgendes feststellen:

- Die Lösung der Poisson-Gleichung reduziert sich auch in höheren Dimensionen nach Diskretisierung wieder auf die Lösung einer großen linearen Gleichungssystems (6.17).

- Probleme bereiten in höheren Dimensionen die Diskretisierung komplexer Geometrien, weil die Indizierung aufwendig wird
- Die Größe des Gleichungssystems ist gleich der Zahl der Gitterpunkte, d.h. in höheren Dimensionen kann die Größe des Gleichungssystems auch schnell zu einem Problem werden

6.2 Wellengleichung

Wellen-Gleichung und Advektionsgleichung sind Kontinuitätsgleichungen, bei denen ein explizites FTCS-Diskretisierungsschema instabil ist, wie eine von Neumann Stabilitätsanalyse zeigt. Dagegen bleibt das Lax-Schema stabil für genügend kleine Zeitschritte, die das Courant-Friedrichs-Lowy Kriterium erfüllen.

In der Physik spielen partielle DGLn, die sich in der Form von **Kontinuitätsgleichungen**

$$\partial_t \vec{u} = -\partial_x \vec{j}(\vec{u}) \quad (6.22)$$

mit einer Stromdichte $\vec{j}(\vec{u})$ schreiben lassen, eine große Rolle (weil man sich einfach oft für den Transport von global erhaltenen Größen wie Energie, Teilchen usw. interessiert, die solche Kontinuitätsgleichungen erfüllen müssen).

Die **Wellengleichung**

$$\partial_t^2 u = v^2 \partial_x^2 u$$

für eine Funktion $u(x, t)$ mit einer konstanten Geschwindigkeit v ist tatsächlich auch von dieser Form, was klar wird, wenn wir den Vektor $\vec{u} = (\partial_t u, v \partial_x u)$ betrachten:

$$\partial_t \vec{u} = \begin{pmatrix} \partial_t^2 u \\ v \partial_t \partial_x u \end{pmatrix} = \begin{pmatrix} v^2 \partial_x^2 u \\ v \partial_t \partial_x u \end{pmatrix} = \partial_x v \begin{pmatrix} v \partial_x u \\ \partial_t u \end{pmatrix} = -\partial_x \begin{pmatrix} 0 & -v \\ -v & 0 \end{pmatrix} \cdot \vec{u}$$

Diese Schreibweise ist beispielsweise analog zur Schreibweise der Maxwell-Gleichungen im Vakuum durch den dualen Feldstärketensor, $\partial_\alpha \tilde{F}^{\alpha\beta} = 0$, die auch die Form einer Kontinuitätsgleichung $\partial_\alpha j^\alpha = 0$ hat: Setzt man

$$\vec{u} = \begin{pmatrix} E \\ B \end{pmatrix} \quad \text{und} \quad \vec{j}(\vec{u}) = \begin{pmatrix} 0 & -v \\ -v & 0 \end{pmatrix} \cdot \vec{u}$$

ergibt sich in der Tat die elektromagnetische Wellengleichung

$$\begin{aligned} \partial_t E &= -v \partial_x B \\ \partial_t^2 E &= -v \partial_t \partial_x B = v^2 \partial_x^2 E \end{aligned}$$

Die Wellengleichung ist eine **hyperbolische partielle DGL 2. Ordnung**. Anstatt der Wellengleichung werden wir die einfachste Form einer Kontinuitätsgleichung, nämlich die **Advektionsgleichung**

$$\partial_t u = -v \partial_x u \quad (6.23)$$

mit einem Strom $j(u) = vu$ betrachten, die eine **partielle DGL 1. Ordnung** ist. Diese besitzt ähnlich wie die Wellengleichung auch Lösungen der Form $u = f(x - vt)$. Die Wellengleichung hat nach d'Alembert rechts- **und** linkslaufende Lösungen $u = f(x - vt) + g(x + vt)$ (weil sie 2. Ordnung ist, während die Advektionsgleichung nur 1. Ordnung ist).

Wie bei der Poisson-Gleichung **diskretisieren** wir, und zwar sowohl x mit einer Gitterkonstanten Δx als auch die Zeit t in Zeitschritte Δt :

$$u_j^n = u(x_j, t_n) \quad \text{mit} \quad x_j = j \Delta x, \quad t_n = n \Delta t$$

Das einfachste **Differenzenschema** von (6.23) benutzt eine Vorwärtsdifferenz in der Zeit – wie beim Euler-Verfahren (4.8) – und eine zentrale Differenz in der x -Koordinate (siehe (3.2)):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -\frac{j(u_{j+1}^n) - j(u_{j-1}^n)}{2\Delta x} = -v \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$$

Daher heißt dieses Schema auch **FTCS-Schema (forward in time, centered in space)**. Es ist in jedem Zeitschritt von erster Ordnung in der Zeit und zweiter Ordnung in der Ortskoordinate.

Es führt auf

$$u_j^{n+1} = u_j^n - \frac{v\Delta t}{2\Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (6.24)$$

Dies stellt ein **explizites Schema** dar, wo die Funktionswerte zum Zeitschritt $n + 1$ auf der linken Seite explizit nur durch Funktionswerte zur vorangehenden Zeit n ausgedrückt ist und ist daher sehr einfach zu iterieren.

Das explizite FTCS-Schema kann wie auf der Abbildung rechts grafisch dargestellt werden: Gestrichelte Linien verbinden die Punkte, die zur Diskretisierung der Zeitableitung herangezogen werden; durchgezogene Linien verbinden Punkte, die zur Diskretisierung der Ortsableitung herangezogen werden.

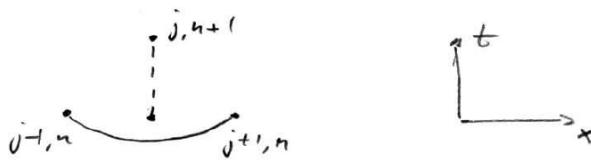


Abbildung 6.2: Links: John von Neumann (1903-1957), österreichisch-ungarischer Mathematiker, gilt als “Vater” der Informatik. Mitte: Peter Lax (geb. 1926), ungarischer Mathematiker. Rechts: Richard Courant (1888-1972), deutsch-amerikanischer Mathematiker. (Quelle: Wikipedia).

Wir wollen nun das explizite Schema auf **Stabilität** untersuchen. Dazu machen wir eine sogenannte **von Neumann Stabilitätsanalyse**,¹ in der wir Stabilität bezüglich einer Störung

$$\delta u_j^n = \xi(k)^n e^{ikj\Delta x} \quad (6.25)$$

untersuchen. Dies ist ein Separationsansatz: Der zweite Faktor $e^{ikj\Delta x}$ ist eine Eigenfunktionen der diskretisierten Ortsableitung, die eine Störung mit Wellenvektor k beschreibt. Der erste Faktor ξ^n ist eine Eigenfunktion von Differenzengleichungen in der Zeit n mit einer zu bestimmenden k -abhängigen Zahl ξ , die als “Verstärkungsfaktor” bezeichnet wird: Wenn $|\xi(k)| > 1$, wächst eine Störung mit Wellenzahl k exponentiell an im Laufe der Zeit, was eine **Instabilität** bedeutet.

¹ Das Verfahren wurde von John von Neumann in Los Alamos im Rahmen des Manhattan-Projekts entwickelt. Während des Krieges wurde die Methode unter Verschluss gehalten und erst 1947 von Crank und Nicolson publiziert.

Wir setzen also den Ansatz (6.25) in das explizite FTCS-Schema (6.24) ein und können damit den k -abhängigen Verstärkungsfaktor bestimmen:

$$\xi(k) = 1 - i \frac{v\Delta t}{\Delta x} \sin(k\Delta x)$$

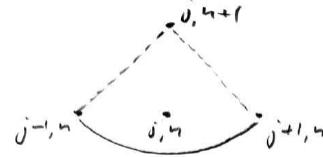
Es gilt offensichtlich *immer* $|\xi(k)| > 1$ und damit ist das explizite FTCS-Schema **immer instabil** und sollte nicht verwendet werden.

Weitaus bessere Stabilitätseigenschaften hat das **Lax-Schema**. Im Lax-Schema ersetzen wir in (6.24) einfach $u_j^n \rightarrow \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)$, was auf

$$u_j^{n+1} = \frac{1}{2} (u_{j+1}^n + u_{j-1}^n) - \frac{v\Delta t}{2\Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (6.26)$$

führt. Auch das Lax-Schema ist ein **explizites Schema** und wie das FTCS-Schema in jedem Zeitschritt von erster Ordnung in der Zeit und zweiter Ordnung in der Ortskoordinate.

Das explizite Lax-Schema kann wie auf der Abbildung rechts grafisch dargestellt werden.



Es hat jedoch bessere Stabilitätseigenschaften, wie die von Neumann Analyse zeigt. Einsetzen des Störungsansatzes (6.25) ergibt für das Lax-Schema (6.26) einen Verstärkungsfaktor

$$\xi(k) = \cos(k\Delta x) - i \frac{v\Delta t}{\Delta x} \sin(k\Delta x),$$

und wir bekommen stabiles Verhalten mit $|\xi(k)| < 1$, solange

$$\frac{v\Delta t}{\Delta x} < 1 \quad (6.27)$$

gilt. Dies ist das **Courant-Friedrichs-Lowy Kriterium**. Es besagt anschaulich, dass die “Informationsausbreitungsgeschwindigkeit” $\Delta x/\Delta t$ durch den numerischen Algorithmus mindestens so groß sein muss, wie die physikalische Ausbreitungsgeschwindigkeit v der Lösung. Das Lax-Schema bleibt damit **stabil** für genügend **kleine Δt** .

Für die Praxis kann man sich merken, dass das Lax-Schema zu stabilen Verfahren für alle Kontinuitätsgleichungen (6.22) führt bei genügend kleinem Δt .

6.3 Diffusionsgleichung

Die Diffusionsgleichung ist eine Kontinuitätsgleichung, bei der bereits das explizite FTCS-Diskretisierungsschema ein für kleine Zeitschritte stabiles Verfahren ergibt. Das implizite Crank-Nicolson-Schema bleibt auch für größere Zeitschritte stabil.

Die **Diffusionsgleichung**

$$\partial_t u = D \partial_x^2 u$$

für eine Funktion $u(x, t)$ ist eine **parabolische partielle DGL 2. Ordnung**. Auch sie ist von der Form einer **Kontinuitätsgleichung** mit einem diffusiven Strom $j(u) = -D\partial_x u$.

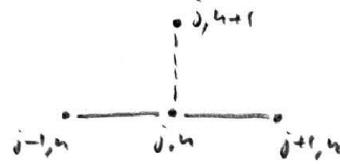
Dennoch betrachten wir die Diffusionsgleichung noch einmal gesondert. Wir wollen auch hier ein einfaches **explizites FTCS-Schema** zur Diskretisierung verwenden, indem wir die Zeit als Euler-artige Vorwärts-Differenz diskretisieren und die zweite Ableitung gemäß (3.4):

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}$$

Dies führt zu

$$u_j^{n+1} = u_j^n + D \frac{\Delta t}{\Delta x^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \quad (6.28)$$

Das explizite FTCS-Schema für die Diffusionsgleichung kann wie auf der Abbildung rechts grafisch dargestellt werden.



Wir führen wieder eine von Neumann Stabilitätsanalyse durch. Einsetzen des Störungsansatzes (6.25) ergibt für das FTCS-Schema (6.28) einen Verstärkungsfaktor

$$\xi(k) = 1 - \frac{4D\Delta t}{\Delta x^2} \sin^2 \left(\frac{k\Delta x}{2} \right).$$

Also ist $|\xi(k)| < 1$, wenn

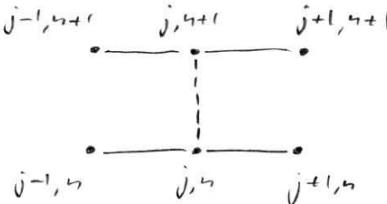
$$\frac{2D\Delta t}{\Delta x^2} < 1 \quad (6.29)$$

und das FTCS-Verfahren bleibt **stabil** für **kleine** Δt . Das Kriterium (6.29) besagt, dass die "Informations-Diffusionskonstante" $\Delta x^2/\Delta t$ des numerischen Algorithmus mindestens so groß sein muss, wie die physikalische Diffusionskonstante D .

Kann man auch größere Zeitschritte Δt stabil realisieren? Dies funktioniert tatsächlich, und zwar mit **impliziten Schemata**, d.h. wir nehmen die rechte Seite in (6.28) (teilweise) bei der Zeit $n+1$. Ein Schema, das in der Zeit sogar exakt bis zur zweiten Ordnung in Δt ist, ist das **Crank-Nicolson-Schema**

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D \frac{1}{2} \frac{(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + (u_{j+1}^n - 2u_j^n + u_{j-1}^n)}{\Delta x^2} \quad (6.30)$$

Das implizite Crank-Nicolson-Schema für die Diffusionsgleichung kann wie auf der Abbildung rechts grafisch dargestellt werden.



Die von Neumann Stabilitätsanalyse ergibt für das Crank-Nicolson-Schema einen Verstärkungsfaktor

$$\xi(k) = \frac{1 - 2 \frac{D\Delta t}{\Delta x^2} \sin^2 \left(\frac{k\Delta x}{2} \right)}{1 + 2 \frac{D\Delta t}{\Delta x^2} \sin^2 \left(\frac{k\Delta x}{2} \right)},$$

also gilt immer $|\xi(k)| < 1$ und damit ist das Crank-Nicolson-Schema **stabil für alle** Zeitschritte Δt .

Ein Nachteil impliziter Verfahren wie dem Schema (6.30) ist, dass dort in **jedem** Zeitschritt ein tridiagonales lineares Gleichungssystem für den Vektor \vec{u}^{n+1} (also u_j^{n+1} für $j = 0, \dots, K$ bei festem $n+1$) gelöst werden, was numerischen Mehraufwand bedeutet.



Abbildung 6.3: Links: John Crank (1916-2006), englischer mathematischer Physiker. Mitte: Phyllis Nicolson (1917-1968), englische Mathematikerin. Rechts: Erwin Schrödinger (1887-1961), Nobelpreis 1933. (Quelle: Wikipedia).

6.4 Schrödingergleichung

Bei der numerischen Lösung der Schrödingergleichung sollte auch unitäre Zeitentwicklung gewährleistet sein. Das Crank-Nicolson-Schema ist sowohl stabil als auch unitär.

In der Ortsdarstellung lautet die (eindimensionale) Schrödingergleichung für eine Wellenfunktion $\psi(x, t) = \langle x | \psi(t) \rangle$

$$i\hbar\partial_t\psi = \hat{H}\psi = -\frac{\hbar^2}{2m}\partial_x^2\psi + V(x)\psi$$

Gegeben ist dabei ein Anfangszustand $\psi(x, 0)$ bei $t = 0$, z.B. ein Wellenpaket, und gesucht ist die Zeitentwicklung.

Hier gibt es eine wichtige Eigenschaft quantenmechanischer Zustände und ihrer Zeitentwicklung zu berücksichtigen:

- 1) Quantenmechanische Zustände sind **normiert**

$$\int dx |\psi(x, t)|^2 = 1$$

wobei $|\psi(x, t)|^2 dx$ die Wahrscheinlichkeit ist, das Teilchen zur Zeit t in $[x, x + dx]$ zu finden.

- 2) Diese Normierung bleibt **erhalten** unter der Zeitentwicklung mit der Schrödingergleichung (Teilchenzahlerhaltung):

$$\begin{aligned} i\hbar\partial_t|\psi\rangle &= \hat{H}|\psi\rangle \\ -i\hbar\partial_t\langle\psi| &= \langle\psi|\hat{H} \\ \Rightarrow \partial_t\langle\psi|\psi\rangle &= \frac{1}{i\hbar} (\langle\psi|\hat{H}|\psi\rangle - \langle\psi|\hat{H}|\psi\rangle) \end{aligned}$$

Diese beiden Eigenschaften sind äquivalent zur Feststellung, dass der Zeitentwicklungsoperator $\hat{S}_H(t, 0)$ mit

$$|\psi(t)\rangle = \hat{S}_H(t, 0)|\psi(0)\rangle$$

unitär ist. Für einen zeitunabhängigen Hamiltonoperator \hat{H} gilt

$$\hat{S}_H(t, 0) = \exp\left(-\frac{i}{\hbar}\hat{H}t\right) \quad (6.31)$$

Auch die numerische Zeitentwicklung sollte deshalb unitär sein, also die Norm erhalten.

Für ein zeitunabhängigen Hamiltonoperator \hat{H} gibt das **explizite FTCS-Schema**

$$i\hbar \frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = \sum_k H_{jk} \psi_k^n \quad (6.32)$$

wobei H_{jk} der diskretisierte Hamiltonoperator (in Ortsdarstellung) ist:

$$\sum_k H_{jk} \psi_k^n = -\frac{\hbar^2}{2m\Delta x^2} (\psi_{j+1}^n - 2\psi_j^n - \psi_{j-1}^n) + V(x_j) \psi_j^n$$

Das FTCS-Schema (6.32) kann auch als

$$\vec{\psi}^{n+1} = \left(\underline{\underline{1}} - i \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \cdot \vec{\psi}^n \quad (6.33)$$

geschrieben werden. Die von Neumann Stabilitätsanalyse zeigt, dass das FTCS-Schema (6.32) **instabil** ist. Außerdem ist es **nicht unitär** wegen

$$\left(\underline{\underline{1}} - i \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^+ \cdot \left(\underline{\underline{1}} - i \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) = \underline{\underline{1}} + \frac{1}{\hbar^2} \underline{\underline{H}}^2 \Delta t^2 \neq \underline{\underline{1}}$$

Daher ist das explizite FTCS-Schema schlecht geeignet zur numerischen Lösung der Schrödinger-Gleichung.

Das **Crank-Nicolson-Schema** dagegen führt auf

$$i\hbar \frac{\vec{\psi}^{n+1} - \vec{\psi}^n}{\Delta t} = \frac{1}{2} \underline{\underline{H}} \cdot \vec{\psi}^n + \frac{1}{2} \underline{\underline{H}} \cdot \vec{\psi}^{n+1} \quad (6.34)$$

oder

$$\left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \cdot \vec{\psi}^{n+1} = \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \cdot \vec{\psi}^n.$$

Dies führt zur Darstellung

$$\vec{\psi}^{n+1} = \left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^{-1} \cdot \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \cdot \vec{\psi}^n \quad (6.35)$$

Das Crank-Nicolson-Schema (6.34) ist **stabil**, wie eine von Neumann Analyse zeigt, und **unitär**:

$$\begin{aligned} & \left[\left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^{-1} \cdot \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \right]^+ \cdot \left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^{-1} \cdot \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \\ &= \left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \cdot \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^{-1} \cdot \left(\underline{\underline{1}} + \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right)^{-1} \cdot \left(\underline{\underline{1}} - \frac{i}{2} \frac{\Delta t}{\hbar} \underline{\underline{H}} \right) \\ &= \underline{\underline{1}} \end{aligned}$$

Daher ist das Crank-Nicolson-Schema (6.34) das Verfahren der Wahl zur numerischen Lösung der Schrödinger-Gleichung.

6.5 Literaturverzeichnis Kapitel 6

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] R. Fitzpatrick. *Computational Physics (Skript)*. Austin, Texas: The University of Texas at Austin, 2012.
- [4] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.
- [5] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

6.6 Übungen Kapitel 6

1. Zeitabhängige Schrödingergleichung

Wir simulieren die Bewegung eines quantenmechanischen Teilchens in einer Dimension mit der Wellenfunktion $\psi(x, t)$ im harmonischen Oszillatorenpotential. Dazu integrieren wir numerisch die zeitabhängige Schrödingergleichung

$$i\hbar\partial_t\psi = -\frac{\hbar^2}{2m}\partial_x^2\psi + \frac{1}{2}m\omega^2x^2\psi = \hat{H}\psi. \quad (6.36)$$

Der Anfangszustand soll ein normiertes Gaußpaket sein (siehe unten Gleichung (6.40)).

- a) Zunächst machen wir die Schrödingergleichung (6.36) einheitenlos, indem wir Zeit- und Ortskoordinaten umskalieren. Wir messen Zeit in Einheiten von $2/\omega$ und reskalieren $\tau \equiv \omega t/2$. Mit welchem Faktor α muss die Ortskoordinate reskaliert werden, $\xi \equiv \alpha x$, um die Schrödingergleichung (6.36) in die Form

$$i\partial_\tau\psi = -\partial_\xi^2\psi + \xi^2\psi = \hat{H}\psi \quad (6.37)$$

zu bringen? Mit welchem Faktor β haben wir dann den Hamiltonoperator $\hat{H} = \beta\hat{H}$ reskaliert?

- b) Wir verwenden den Crank-Nicolson Algorithmus und lösen die einheitenlose Schrödingergleichung (6.37) auf einem Gitter $\xi_n = n\Delta\xi$.

Der diskretisierte Hamiltonoperator ist dann durch die Matrix

$$H_{nm} = -\frac{1}{(\Delta\xi)^2} (\delta_{n,m-1} + \delta_{n,m+1} - 2\delta_{nm}) + (\Delta\xi)^2 n^2 \delta_{nm} \quad (6.38)$$

gegeben. Der diskretisierte Zeitentwicklungsoperator für einen Zeitschritt der Länge $\Delta\tau$ nach Crank und Nicolson lautet

$$\underline{\underline{S}}_H = \left(\underline{\underline{I}} + \frac{i}{2} \underline{\underline{H}} \Delta\tau \right)^{-1} \left(\underline{\underline{I}} - \frac{i}{2} \underline{\underline{H}} \Delta\tau \right) \quad (6.39)$$

Berechnen Sie diese Matrix für $\Delta\tau = 0.05$ für ein System der Größe $\xi \in [-10, 10]$, das mit $\Delta\xi = 0.1$ diskretisiert wird. Welche Dimension haben dann die Matrizen $\underline{\underline{H}}$ und $\underline{\underline{S}}_H$? Zur Berechnung der Inversen in (6.39) können Sie einen Algorithmus Ihrer Wahl (z.B. Gauß-Elimination, LU-Zerlegung, Jacobi-Gauß-Seidel-Iteration) selbst programmieren oder ein entsprechendes fertiges Unterprogramm, z.B. aus Eigen, LAPACK oder NumRec verwenden.

- c) Der Anfangszustand soll ein normiertes Gaußpaket sein mit $\langle \xi \rangle = \int d\xi \xi |\psi(\xi, \tau)|^2 = \xi_0$ und $\langle \xi^2 \rangle - \langle \xi \rangle^2 = \sigma^2$:

$$\psi(\xi, 0) = \left(\frac{1}{2\pi\sigma} \right)^{1/4} e^{-(\xi - \xi_0)^2 / 4\sigma^2}. \quad (6.40)$$

Wie lautet der diskretisierte Anfangszustandsvektor $\psi_n(0)$, der diesem Anfangszustand entspricht? Welche Dimension hat er? Normieren Sie den diskretisierten Anfangszustand $\psi_n(0)$ numerisch in Ihrem Programm.

- d) Berechnen Sie für einen solchen Anfangszustand mit $\xi_0 = \sigma = 1$ den Zustand $\psi_n(\tau)$ nach einer Zeit $\tau = 10$ durch fortgesetzte Matrixmultiplikation mit dem in b) berechneten Crank-Nicolson Zeitentwicklungsoperator $\underline{\underline{S}}_H$. Prüfen Sie, ob der Zustand $\psi_n(\tau)$ während der Zeitentwicklung normiert bleibt.

- e) Versuchen Sie, den Zeitverlauf der Wellenfunktion zu visualisieren/animieren, indem Sie mindestens 4 Plots der Wahrscheinlichkeitsverteilung innerhalb einer Schwingungsperiode 2π anfertigen.

- f) Berechnen Sie den Mittelwert $\langle \xi \rangle(\tau) = \sum_n (\Delta\xi) \xi_n |\psi_n(\tau)|^2$ und entsprechend die Schwankung $\langle \xi^2 \rangle(\tau) - \langle \xi \rangle^2(\tau)$ während der Bewegung $0 < \tau < 10$. Erstellen Sie Plots vom zeitlichen Verlauf

dieser Größen. Berechnen Sie außerdem Mittelwert und Schwankung des zu $\hat{\xi}$ gehörigen ‘‘Impulsoperators’’ $\hat{p}_\xi \equiv -i\partial_\xi$ und plotten Sie auch deren zeitlichen Verlauf. Diskutieren Sie die Ergebnisse vor dem Hintergrund der klassischen Bewegung im Oszillatorenpotential und der Heisenbergschen Unschärferelation.

g) Verwenden Sie ein einfaches explizites Schema statt des Crank-Nicolson-Schemas (6.39) und vergleichen Sie die Ergebnisse, besonders bezgl. der Normierung.

h) Fügen Sie noch eine kleine Anharmonizität

$$V_{nm} = +\epsilon(\Delta\xi)^4 n^4 \delta_{nm} \quad (6.41)$$

zum Hamiltonian H hinzu und vergleichen Sie das Verhalten des Wellenpaketes.

2. Poisson-Gleichung

Lösen Sie die 2D Poisson-Gleichung

$$\partial_x^2 \phi + \partial_y^2 \phi = -\rho(x, y) \quad (6.42)$$

(also $\varepsilon_0 = 1$) mit Hilfe der Jacobi- oder der Gauß-Seidel-Iteration für folgendes System:

- Ein Quadrat $Q = [0, 1] \times [0, 1]$
- Dirichlet-Randbedingungen mit vorgegebenem Potential ϕ auf den Quadraträndern.
- Als Quellen positionieren Sie im Inneren diskrete Ladungen q_i an Orten \vec{r}_i , also $\rho(\vec{r}) = \sum_i q_i \delta(\vec{r} - \vec{r}_i)$ (Vorsicht bei der korrekten Diskretisierung der δ -Funktion).

a) Diskretisieren Sie das System mit $\Delta = 0.05$ und implementieren Sie die Jacobi- und/oder Gauß-Seidel-Iteration. Bei jeder Iteration sollte der Algorithmus einmal jeden Gitterplatz im Inneren updaten (ohne die Ränder zu verändern). Wählen Sie als Anfangsbedingung $\phi = 0$ und testen Sie den Algorithmus für $\rho = 0$ (keine Quellen) für Randbedingungen $\phi = \text{const} = 0$. Schreiben Sie eine Ausgaberoutine für $\phi(\vec{r})$ und das elektrische Feld $\vec{E} = -\vec{\nabla}\phi$.

b) Lösen Sie nun die Poissongleichung für $\rho = 0$ im Inneren und mit Randbedingungen $\phi = 0$ auf den 3 Rändern $x = 0$, $x = 1$ und $y = 0$, aber $\phi(x, 1) = 1$ auf dem Rand $y = 1$. Leiten Sie auch die analytische Lösung für $\phi(x, y)$ her (Fourierzerlegung) und vergleichen Sie das Resultat.

c) Wählen Sie wieder $\phi = \text{const} = 0$ auf allen Rändern und setzen nun eine Ladung $q_1 = +1$ ins Innere. Berechnen Sie $\phi(\vec{r})$ im Inneren durch Iteration, bis zu einer Genauigkeit 10^{-5} . Plotten Sie die Potentialverteilung $\phi(\vec{r})$ und den Betrag der Feldstärke $|\vec{E}|(\vec{r})$.

d) Überzeugen Sie sich, dass das elektrische Feld am Rand keine Tangentialkomponente besitzt (warum?). Berechnen Sie numerisch die auf dem Rand influenzierte Ladungsdichte σ über die Normalkomponente des Feldes $\sigma = -\vec{n} \cdot \vec{\nabla}\phi = E_n$. Berechnen Sie numerisch das Linienintegral $\int_{\partial Q} dl \sigma$, also das 2D-Analogon zur Oberflächenladung $\int_{\partial Q} df \sigma$ in 3D. Wie lautet das theoretische Ergebnis für diese influenzierte Oberflächenladung? (Sie können diese Frage auch in 3 Dimensionen beantworten)

e) Wählen Sie sich eine andere neutrale Ladungskonfiguration mit mindestens 2 Ladungen ($\sum_i q_i = 0$) und Randbedingungen $\phi = \text{const} = 0$ auf allen Rändern. Führen Sie wieder die Aufgabenstellungen aus **c)** und **d)** durch.

7 Iterationsverfahren

Literatur zu diesem Teil:

Iterationsverfahren zur Lösung nichtlinearer Gleichungen werden in den Numerical Recipes [1, 2] ausführlich besprochen. Mean-Field Theorien wie das Hartree-Fock-Verfahren findet man in Thijssen [3]. Iterationen, Bifurkationen und der Weg ins Chaos sind ein klassisches Computerphysikthema, siehe z.B. Korsch und Jodl [4] bzw. Kinzel und Reents [5]. Die mathematischen Grundlagen sind dem Buch von Strogatz [6] entnommen.

7.1 Iterationen, Banachscher Fixpunktsatz

Viele numerische Probleme werden iterativ gelöst, d.h. von einem Verfahren, welches aus vielen Schritte gleicher Art besteht. Die Frage, in welchen Fällen diese Verfahren konvergieren, wird durch den Banachschen Fixpunktsatz beantwortet, welchen wir in diesem Kapitel vorstellen werden.

Iterationen, bei welchen eine Funktionsvorschrift f wieder und wieder angewendet wird,

$$x_{n+1} = f(x_n) \quad \text{bzw.} \quad x_n = f^n(x_0) \equiv \underbrace{(f \circ f \dots \circ f)}_{n \text{ mal}}(x_0),$$

sind ideale Aufgaben für Computer. Sie sollen so lange ausgeführt werden, bis sie gegen einen gesuchten **Fixpunkt** x^* mit

$$x^* = f(x^*)$$

konvergiert sind.

In Kapitel [6] haben wir bereits zwei Beispiele für Iterationsverfahren kennen gelernt, nämlich das Jacobi- und das Gauß-Seidel-Verfahren zur Lösung der Poisson-Gleichung. Die entscheidenden Fragen liegen auf der Hand: Ist die Konvergenz gesichert? Falls ja, wie schnell konvergiert das Verfahren?

Die Konvergenzfrage kann oft mit Hilfe des **Banachschen Fixpunktsatzes** beantwortet werden. Dazu müssen einige Begriffe eingeführt werden.

Ein **Banachraum** ist ein normierter Vektorraum (mit Norm $\|\cdot\|$, die sich dadurch definiert, dass sie positiv ist und die Dreiecksungleichung erfüllt), in welchem jede Cauchy-Folge ($\forall \varepsilon \exists N : \|a_n - a_m\| < \varepsilon \quad \forall n, m > N$) konvergiert.

Eine Abbildung F eines Banachraums auf sich selbst heißt **Lipschitz-stetig**, falls

$$\|F(x) - F(y)\| < L\|x - y\| \tag{7.1}$$

mit einer festen Lipschitz-Konstanten L . Falls diese Lipschitz-Konstante $L < 1$ ist, heißt F eine **Kontraktion**.

Für eine Kontraktion $F : A \rightarrow A$, wobei A eine abgeschlossene Teilmenge eines Banachraums ist, gilt der **Banachsche Fixpunktsatz**, der eine (i) Existenz-, (ii) Konstruktions- und (iii) Fehleraussage

macht:

- | | |
|--|-------|
| <ul style="list-style-type: none"> (i) F besitzt genau einen Fixpunkt x^* (ii) Für jeden Startwert $x_0 \in A$ konvergiert die Folge $x_{n+1} = F(x_n)$ gegen $x^* \in A$ | (7.2) |
| (iii) Es gilt die Fehlerabschätzung $\ x_n - x^*\ \leq \frac{L^{n-l}}{1-L} \ x_{l+1} - x_l\ $
bzw. mit $l = 0$: $\ x_n - x^*\ \leq \frac{L^n}{1-L} \ x_{l+1} - x_0\ $ | |

(Beweis siehe Mathematik-Vorlesung).

Falls $L < 1$ aus (7.1) bekannt ist, lässt sich mit (7.2)ii) die Iterationszahl bis zur gewünschten Genauigkeit abschätzen.

Die Konvergenz-“Geschwindigkeit” einer Iteration kann über die **Konvergenzordnung** p klassifiziert werden: Gilt

$$\limsup_{n \rightarrow \infty} \frac{\|x_{n+1} - x^*\|}{\|x_n - x^*\|^p} = k < \infty \quad (7.3)$$

heißt die Iteration **konvergent vom Grade p** .

In vielen Iterationen, die in der Praxis genutzt werden, gelten nicht alle Voraussetzungen des Banachschen Fixpunktsatzes, weshalb solche Verfahren nicht für beliebige Startwerte konvergieren. Gilt in solchen Verfahren Eigenschaft (7.2)i) in einer Umgebung eines Fixpunktes, so nennt man diesen Fixpunkt **stabil**.



Abbildung 7.1: Links: Stefan Banach (1892-1945), polnischer Mathematiker. Rechts: Carl Jacobi (1804-1851), deutscher Mathematiker. (Quelle: Wikipedia).

Beispiele

1) Das einfachste Beispiel ist der Banachraum \mathbb{R} mit einer Abbildung, die durch eine Funktion $f(x)$ gegeben ist. Es handelt sich um eine Kontraktion, falls $|f'(x)| < 1 \quad \forall x$. Dann konvergiert die Folge $x_{n+1} = f(x_n)$ gegen die Lösung der Fixpunktgleichung $x^* = f(x^*)$.

Ist f keine Kontraktion, kann man ersatzweise die Funktion $g(x) \equiv \rho f(x) + (1 - \rho)x$ betrachten. Diese Funktion hat immer noch den selben Fixpunkt wie f , da $x^* = f(x^*) \Leftrightarrow x^* = g(x^*)$. Die Kontraktionseigenschaft $|g'(x)| < 1$ kann durch die Wahl eines geeigneten $\rho < 0$ erreicht werden, da $g'(x) = \rho f'(x) + (1 - \rho) = \rho(f'(x) - 1) + 1$.

2) Ein weiteres Beispiel sind die Jacobi- oder Gauß-Seidel Iterationen, die wir in Kap. 6.1 zur Lösung der Poisson-Gleichung kennen gelernt haben. Man kann diese Verfahren auch allgemeiner

zur Lösung beliebiger linearer Gleichungssysteme

$$\underline{\underline{A}} \vec{\phi} = \vec{r} \quad \text{mit } \vec{\phi}, \vec{r} \in \mathbb{R}^N. \quad (7.4)$$

verwenden. Bei der **Jacobi-Iteration** wird $\underline{\underline{A}}$ in einen Diagonalanteil $\underline{\underline{D}}$ (mit $D_{ij} = 0$ für $i \neq j$) und einen Rest zerlegt: $\underline{\underline{A}} = \underline{\underline{D}} + (\underline{\underline{A}} - \underline{\underline{D}})$. Die Lösung des Gleichungssystems (7.4) kann dann in Form einer Fixpunktgleichung geschrieben werden:

$$\begin{aligned} \underline{\underline{A}} \cdot \vec{\phi} = \vec{r} &\Leftrightarrow \underline{\underline{D}} \cdot \vec{\phi} = \vec{r} + (\underline{\underline{D}} - \underline{\underline{A}}) \cdot \vec{\phi} \\ &\Leftrightarrow \vec{\phi} = \underline{\underline{D}}^{-1} \cdot \vec{r} + \underbrace{(\underline{\underline{I}} - \underline{\underline{D}}^{-1} \cdot \underline{\underline{A}}) \cdot \vec{\phi}}_{\equiv \underline{\underline{T}}_J} \equiv \vec{F}(\vec{\phi}), \end{aligned}$$

Die gesuchte Lösung $\vec{\phi}$ ist also ein Fixpunkt der linearen Abbildung $\vec{F}(\vec{\phi})$.

Die Jacobi-Iteration besteht dann in der Vorschrift

$$\vec{\phi}^{(n+1)} = \vec{F}(\vec{\phi}^{(n)}).$$

Die Konvergenz dieser Fixpunktiteration lässt sich mit dem Banachschen Fixpunktsatz untersuchen. Der Banachraum ist hierbei \mathbb{R}^N . Zur Untersuchung der Lipschitz-Stetigkeit der linearen Abbildung \vec{F} benutzen wir

$$|\vec{F}(\vec{x}) - \vec{F}(\vec{y})| = |\underline{\underline{T}}_J \cdot (\vec{x} - \vec{y})| < |\lambda_{max}| |\vec{x} - \vec{y}|,$$

wobei λ_{max} der betragsmäßig größte Eigenwert der NxN Matrix $\underline{\underline{T}}_J$ ist. Die lineare Abbildung \vec{F} ist also genau dann eine Kontraktion, wenn $|\lambda_{max}| < 1$ für den diesen Eigenwert gilt.

Mit Hilfe dieses Ergebnisses wollen wir nun die Jacobi-Iteration (6.12) für die eindimensionale Poisson-Gleichung auf Konvergenz untersuchen. Die Matrizen $\underline{\underline{A}}$, $\underline{\underline{D}}$ und $\underline{\underline{T}}_J$ für die Jacobi-Iteration (6.12) lauten

$$\begin{aligned} \underline{\underline{D}} &= -2 \cdot \underline{\underline{I}}, \quad \underline{\underline{A}} = \begin{pmatrix} -2 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & -2 \end{pmatrix}, \\ \underline{\underline{T}}_J &= \underline{\underline{I}} - \underline{\underline{D}}^{-1} \cdot \underline{\underline{A}} = \frac{1}{2} \begin{pmatrix} 0 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 0 \end{pmatrix}. \end{aligned}$$

Die Eigenwertgleichung für $\underline{\underline{T}}_J$ lässt sich in Indexschreibweise als

$$\frac{1}{2}(\phi_{n-1} + \phi_{n+1}) = \lambda \phi_n$$

darstellen. Mit dem Ansatz $\phi_n = \sin(kn)$ (auch von Eigenschwingungen der linearen Kette bekannt) folgt durch Additionstheoreme $\lambda = \cos(k)$ für den Eigenwert. Mit den Randbedingungen des physikalischen Problems liegen $\phi_0 = \phi_{N+1} = 0$ fest, d.h. wir müssen $k = m \frac{\pi}{N+1}$ wählen mit $m = 1, \dots, N$. Damit sind die Eigenwerte

$$\lambda_m = \cos\left(m \frac{\pi}{N+1}\right) \quad \text{und} \quad |\lambda_{max}| = \cos\left(\frac{\pi}{N+1}\right) < 1.$$

Damit ist die durch (7.5) definierte lineare Abbildung \vec{F} bei der Jacobi-Iteration (6.12) für die eindimensionale Poisson-Gleichung eine Kontraktion und die Jacobi-Iteration damit konvergent. Wir bemerken aber, dass der größte Eigenwertvertrag $|\lambda_{\max}|$ für große N nur wenig kleiner als 1 ist: $|\lambda_{\max}| \approx 1 - \frac{\pi^2}{2(N+1)^2}$. Daher ist die Konvergenz recht langsam.

7.2 Nullstellen, Nichtlineare Gleichungen

Im folgenden Kapitel werden iterative Verfahren vorgestellt, mit denen sich nichtlineare Gleichungen lösen, bzw. äquivalent dazu Nullstellen nichtlinearer Funktionen finden lassen. Bei allen Verfahren ist Vorsicht geboten, da die Wahl der Startwerte beeinflusst, ob überhaupt und wenn ja gegen welche Nullstelle sie konvergieren.

Gegeben sei eine nichtlineare Gleichung

$$g(x) = h(x) \iff f(x) \equiv g(x) - h(x) = 0$$

deren Lösung x_N auch eine Nullstelle der nichtlinearen Funktion $f(x)$ ist. Wir überlegen uns zunächst in einer Dimension $x \in \mathbb{R}$ wie man diese Nullstelle finden kann und betrachten dazu verschiedene iterative Methoden.

7.2.1 Intervallhalbierung

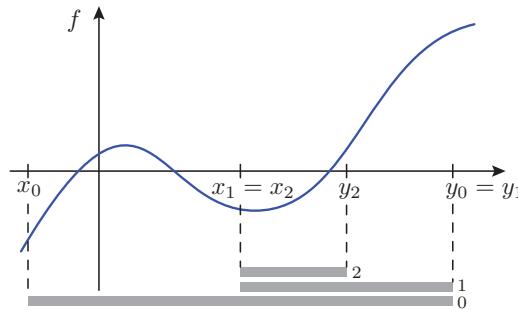


Abbildung 7.2: Prinzip der Intervallhalbierung.

Das Verfahren des Intervallhalbierungs geht folgendermaßen vor (siehe Abb. 7.2):

- 1) Start: Nullstelle(n) sind „eingeklammert“, d.h. x_0 und y_0 ($> x_0$) liegen so, dass $f(x)$ auf $[x_0, y_0]$ sein Vorzeichen wechselt, also $f(x_0) \cdot f(y_0) < 0$.
- 2) Berechne

$$z_{n+1} = \frac{x_n + y_n}{2}$$

(7.5)
- 3) Wenn $f(z_{n+1}) \cdot f(x_n) < 0$, dann setze $x_{n+1} = x_n$, $y_{n+1} = z_{n+1}$,
sonst setze $x_{n+1} = z_{n+1}$, $y_{n+1} = y_n$.
- 4) Wenn $\varepsilon_{n+1} = y_{n+1} - x_{n+1} <$ Genauigkeitsziel ε , Abbruch, sonst wieder 2).

Dieser Algorithmus hat folgende Eigenschaften

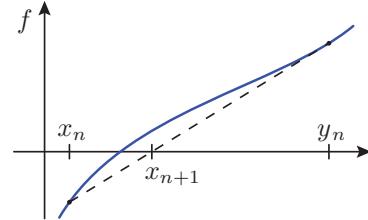
- Er findet garantiert eine Nullstelle.

- Er muss aber auf jeden Fall eine “Klammer” als Startwerte übergeben bekommen. Diese anfängliche Einklammerung einer Nullstelle muss bekannt sein oder mittels eines anderen Algorithmus gefunden werden. Ein weiteres Problem stellt sich, wenn mehrere Nullstellen in der Klammer sind, es wird dann nämlich immer nur eine dieser Nullstelle letztendlich gefunden.
- Das Verfahren ist **konvergent vom Grad $p = 1$** , weil $\varepsilon_{n+1} = \frac{1}{2}\varepsilon_n$ und $|x_n - x_N| < \varepsilon_n$.

7.2.2 Regula Falsi

Die Wahl des Intervallmittelpunkts als neuen Zwischenpunkt war die naivst mögliche. Unter der Annahme, dass die Funktion in der Nähe der Nullstelle ungefähr linear ist, lässt sich diese Wahl durch **lineare Interpolation** verbessern:

$$\tilde{f}(x) = \frac{x - x_n}{y_n - x_n} f(y_n) + \frac{x - y_n}{x_n - y_n} f(x_n)$$



und wähle die neue Zwischenstelle z_{n+1} als Nullstelle von \tilde{f} , d.h.

$$z_{n+1} = \frac{x_n f(y_n) - y_n f(x_n)}{f(y_n) - f(x_n)}. \quad (7.6)$$

Die weiteren Schritte laufen wie bei der Intervallhalbierung ab, die Nullstelle bleibt immer eingeklammert.

Ein Problem bei der Regula Falsi tut sich auf, wenn sich die Funktion $f(x)$ stark ändert in der Nähe der Nullstelle. In diesem Fall findet die Änderung immer auf der gleichen Seite von x_N statt (siehe Abb. 7.3), und das führt zu sehr langsamer (manchmal $p < 1$) Konvergenz. Dieses Problem kann durch eine kleine Modifikation leicht gelöst werden:

Wenn zum (≥ 2)-ten Mal hintereinander $\begin{Bmatrix} x_n \\ y_n \end{Bmatrix}$ geändert wird, setze nächstes Mal $\begin{Bmatrix} f(y_{n+1}) = f(y_n)/2 \\ f(x_{n+1}) = f(x_n)/2 \end{Bmatrix}$.

In der Regel konvergiert Regula Falsi mit $p = 1.618 (= \frac{1+\sqrt{5}}{2}$, goldener Schnitt) und somit schneller als die Intervallhalbierung. Mit der Modifikation gilt sogar immer $p > 1$.

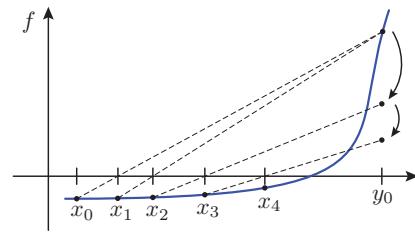


Abbildung 7.3: Modifikation d. Regula Falsi

7.2.3 Newton-Raphson-Methode

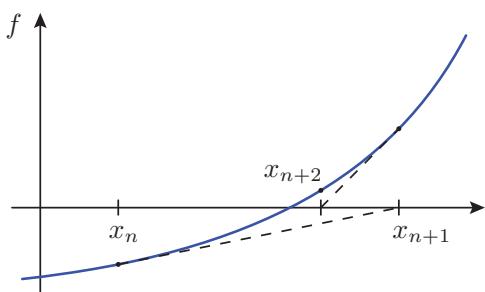


Abbildung 7.4: Beim Newton-Raphson-Verfahren wird die Funktion an der aktuellen Position x_n durch ihre Tangente approximiert. Die nächste Position x_{n+1} ergibt sich dann als Nullstelle der linearen Tangentengleichung.

In den bisherigen Verfahren wurde noch keine Information über die Ableitung der Funktion verwendet, immer nur die Funktionswerte an sich. Das Newton-Raphson-Verfahren benutzt $f'(x)$ für eine Taylorentwicklung um die aktuelle Position x_n :

$$\tilde{f}(x) = f(x_n) + (x - x_n)f'(x_n).$$

Die nächste Position x_{n+1} wird so gewählt, dass $\tilde{f}(x_{n+1}) = 0$ ist (siehe Abb. 7.4), wodurch die Iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (7.7)$$

definiert wird. Es handelt sich um eine Fixpunkt-Iteration zur Funktion

$$F(x) \equiv x - \frac{f(x)}{f'(x)} \quad \text{also mit} \quad F'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

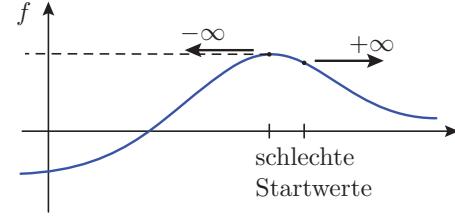
Ein Fixpunkt $F(x^*) = x^*$ ist äquivalent zu $f(x^*) = 0$, so dass $x^* = x_N$ eine Nullstelle ist.

Da an der Nullstelle x_N aber $F'(x_N) = 0$ ist, nennt man x_N einen **superstabilen Fixpunkt** der Iteration. In der Nähe der Nullstelle reduziert sich die Abweichung von der Nullstelle in jedem Schritt quadratisch,

$$\begin{aligned} \Delta x_n &\equiv x_n - x_N \\ \Delta x_{n+1} &= F(x_n) - F(x_N) = F'(x_N)\Delta x_n + \mathcal{O}(\Delta x_n^2) = \mathcal{O}(\Delta x_n^2), \end{aligned}$$

die **Konvergenzordnung** beträgt also $p = 2$ (in der Nähe der Nullstelle).

Probleme können entstehen, wenn man in zu weiter Entfernung zur Nullstelle mit der Iteration beginnt (siehe Abb. rechts). In diesen Fällen ist nicht gesichert, ob das Newton-Verfahren überhaupt konvergiert. Insbesondere Stellen mit $f'(x_n) \approx 0$ haben katastrophale Auswirkungen auf die Iteration. Eine Lösung dieses Problems ist, die Newton-Raphson Methode mit einer "sicheren" einklammernden Intervallhalbierung oder Regula Falsi zu kombinieren.



Damit die hohe Konvergenzordnung auch in der Praxis hohe Effizienz bedeutet, sollte die Ableitung $f'(x)$ **analytisch** bekannt sein. Eine numerische Berechnung über finite Differenzen lohnt sich oft nicht, da man pro Schritt f zweimal auswerten muss. Dann ist die (modifizierte) Regula Falsi normalerweise schneller.

7.2.4 Nullstellen in höheren Dimensionen

Für nichtlineare Gleichungssysteme

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned} \quad \text{oder} \quad \vec{F}(\vec{x}) = 0$$

wird als Nullstelle die Schnittmenge der 0-Konturen der einzelnen Funktionen gesucht. Zur vollständigen Lösung des Problems wäre eine vollständige Kenntnis dieser Konturen $f^{-1}(0)$ und $g^{-1}(0)$ nötig. Da ein Analogon zum sicheren "Einklammern" in einer Dimension in höheren Dimensionen nicht existiert, fehlen "sichere" Algorithmen wie Intervallhalbierung oder Regula falsi.

Zum Finden einer der Nullstellen kann aber das Newton-Raphson-Verfahren verallgemeinert werden.

Mit der Jacobimatrix $J_{i,j}(\vec{x}) = \left. \frac{\partial F_i}{\partial x_j} \right|_{\vec{x}}$ lautet die Iteration

$$\vec{x}_{n+1} = \vec{x} - \underline{J}^{-1}(\vec{x}_n) \cdot \vec{F}(\vec{x}_n) \quad (7.8)$$

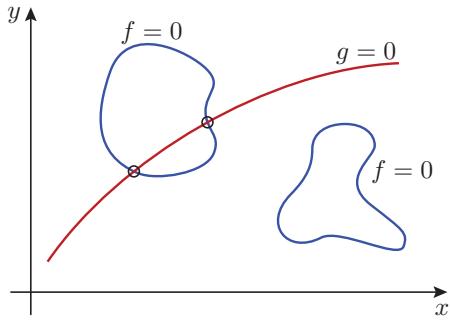


Abbildung 7.5: Die Nullstellen für ein Gleichungssystem ergeben sich aus den 0-Konturen der einzelnen Funktionen.

In jedem Schritt ist also eine Matrix-Inversion nötig, was für mehr als 2 Dimensionen bereits aufwendig wird. Außerdem besteht eine noch größere Gefahr von Katastrophen als in einer Dimension.

Als **Vorsichtsmaßnahme** sollte man erzwingen, dass das Betragsquadrat $\vec{F} \cdot \vec{F}$ der Funktion in jedem Schritt kleiner wird. Für hinreichend kleine Schritte $\delta \vec{x} = -\underline{\underline{J}}^{-1}(\vec{x}) \cdot \vec{F}(\vec{x})$ ist dies automatisch erfüllt, da

$$\begin{aligned}\vec{\nabla}(\vec{F} \cdot \vec{F}) \cdot \delta \vec{x} &= 2\vec{F}(\vec{x})^T \cdot \underline{\underline{J}}(\vec{x}) \cdot \left(-\underline{\underline{J}}^{-1}(\vec{x}) \cdot \vec{F}(\vec{x}) \right) \\ &= -2\vec{F} \cdot \vec{F} < 0.\end{aligned}$$

Falls gemäß (7.8) $\vec{F} \cdot \vec{F}$ ein einem Schritt mal größer werden sollte, benutzt man statt (7.8)

$$\vec{x}_{n+1} = \vec{x}_n - \lambda \left(\underline{\underline{J}}^{-1}(\vec{x}_n) \cdot \vec{F}(\vec{x}_n) \right)$$

mit $0 < \lambda \leq 1$ (“backtracing”).

Details zur Wahl von λ findet man in den Numerical Recipes [2], Kap. 9.7.

7.3 Mean-Field Theorien und selbstkonsistente Gleichungen

Iterierbare selbstkonsistente Gleichungen oder Verfahren sind typisch für Mean-Field-Theorien. Wir betrachten als Beispiele die Hartree-Fock-Näherung für Atome mit mehreren Elektronen und die Mean-Field-Theorie des Ising-Modells, die selbstkonsistent mit Fixpunktiterationen gelöst werden können.

Iterierbare selbstkonsistente Gleichungen $x = f(x)$ sind typisch für Mean-Field (MF) Theorien. In diesem Kapitel lernen wir einige MF Theorien kennen, deren Selbstkonsistenzgleichungen mit den oben beschriebenen Fixpunktiterationen gelöst werden können.

Mean-Field Theorien für Vielteilchenprobleme vernachlässigen bestimmte Korrelationen zwischen Teilchen, z.B. durch Mittelung über räumliche Nachbarn. Ergebnis ist eine Einteilchentheorie mit einem **selbstkonsistent** (iterativ) zu bestimmenden **mittleren Feld**. Wir werden zwei in der Physik außerordentlich wichtige Beispiele genauer diskutieren: Die **Hartree-Fock-Näherung** und die **Molekularfeldtheorie des Ising-Modells**.

7.3.1 Hartree-Fock-Näherung

Wir betrachten ein Atom mit $N > 1$ Elektronen, und interessieren uns für deren Energieniveaus und Eigenzustände. Der Hamiltonoperator lautet

$$\hat{H} = \sum_{i=1}^N \left(\frac{\hat{p}_i^2}{2m} - \frac{Ze^2}{|\vec{r}_i|} \right) + \sum_{i>j} \frac{e^2}{|\vec{r}_i - \vec{r}_j|}$$

und setzt sich zusammen aus kinetischer Energie, Coulomb-Wechselwirkung jedes Elektrons mit dem Kern (Kernladung $Z = N|e|$) sowie der Coulomb-Wechselwirkung der Elektronen untereinander (damit die Energie nicht doppelt gezählt wird, ist die Summe als $\sum_{i>j} = \sum_{j=1}^N \sum_{i=j+1}^N$ zu verstehen; Coulomb-Wechselwirkungen in Gauß Einheiten).

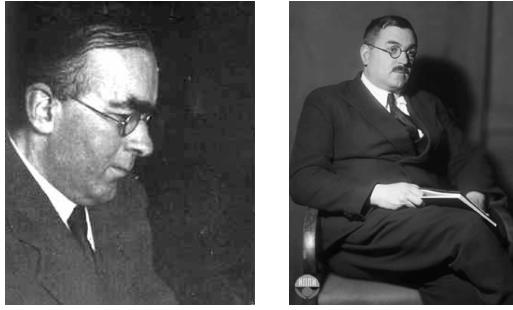


Abbildung 7.6: Links: Douglas Hartree (1897-1958), englischer Mathematiker und Physiker. Rechts: Vladimir Fock (1898-1927), russischer Physiker.

Die Eigenzustände $\hat{H} |\psi\rangle = E |\psi\rangle$ sind aufgrund der Elektron-Elektron-Wechselwirkung nicht exakt berechenbar. In der **Hartree-Näherung** wird das Ritzsche Variationsverfahren mit einem **Produktansatz**

$$\psi(\vec{r}_1 \dots, \vec{r}_N) = \prod_{i=1}^N \phi_i(\vec{r}_i)$$

durchgeführt. Der Produktansatz vernachlässigt Korrelationen zwischen den Elektronen:

$$\langle O_1(\vec{r}_1) \cdot O_2(\vec{r}_2) \rangle = \langle O_1(\vec{r}_1) \rangle \cdot \langle O_2(\vec{r}_2) \rangle.$$

Die Variation des Funktionals $E[\phi_1, \dots, \phi_N] = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle}$ nach den Funktionen $\phi_i(\vec{r}_i)$ ergibt schließlich die **Hartree-Gleichungen**

$$\left(\frac{-\hbar^2}{2m} \vec{\nabla}_i^2 - \frac{Ze^2}{r_i} + V_i(\vec{r}_i) \right) \phi_i(\vec{r}_i) = \varepsilon_i \phi_i(\vec{r}_i) \quad \text{für } i = 1, \dots, N.$$

(7.9)

Die Energien ε_i sind bei der Variation Lagrange-Multiplikatoren für die Normierungen $\langle \phi_i | \phi_i \rangle_i = 1$ der Funktionen $\phi_i(\vec{r}_i)$. Die Hartree-Gleichung (7.9) hat dann die Form einer 1-Teilchen Schrödinger-Gleichung mit einem selbstkonsistent zu bestimmenden mittleren Potential $V_i(\vec{r}_i)$, das von allen anderen Elektronen erzeugt wird:

$$V_i(\vec{r}_i) = \sum_{j(\neq i)} \int d^3 \vec{r}_j \frac{e^2}{|\vec{r}_i - \vec{r}_j|} |\phi_j(\vec{r}_j)|^2.$$

(7.10)

Die Energien ε_i in (7.9) haben auch eine physikalische Interpretation: Sie entsprechen der Ionisierungsenergie, wenn Elektron i entfernt wird (und die anderen Zustände ϕ_j dadurch nicht geändert werden). Die Gesamtenergie des Systems beträgt

$$E = \sum_i \varepsilon_i - \frac{1}{2} \sum_i \int d^3 \vec{r}_i V_i(\vec{r}_i) |\phi_i(\vec{r}_i)|^2.$$

Die Gleichungen (7.9) und (7.10) können durch **Iteration** gelöst werden:

- 1) Starte mit Ansatz $\phi_i^{(0)}$
- 2) Berechne $V_i^{(0)}$ nach (7.10) mit $\phi_i^{(0)}$
- 3) Berechne $\phi_i^{(1)}$ nach (7.9) durch Lösen des Eigenwertproblems mit $V_i = V_i^{(0)}$
- 2₁) Berechne $V_i^{(1)}$ nach (7.10) aus $\phi_i^{(1)}$
- 3₁) ...

Das Eigenwertproblem (7.9) kann dabei mit Schussmethoden oder Diskretisierung (siehe 6) gelöst werden, am besten in Kugelkoordinaten.

Die **Hartree-Fock-Näherung** läuft analog zur Hartree-Näherung, aber mit der **Slater-Determinante** als Ansatz,

$$\psi(\vec{r}_1, s_1, \dots, \vec{r}_N, s_N) = \begin{vmatrix} \phi_1(\vec{r}_1, s_1) & \cdots & \phi_1(\vec{r}_N, s_N) \\ \vdots & & \vdots \\ \phi_N(\vec{r}_1, s_1) & \cdots & \phi_N(\vec{r}_N, s_N) \end{vmatrix}, \quad (7.11)$$

um eine antisymmetrische Wellenfunktion zu garantieren für Teilchen mit Spin $s_i = \pm 1$ (Eigenwerte des zugehörigen quantenmechanischen Operators lauten $\hat{s}_{i,z} = \frac{\hbar}{2}s_i$).

Die aus der Variation resultierenden **Hartree-Fock-Gleichungen** enthalten dann zusätzlich einen Austauschterm

$$V_i(\vec{r}_i)\phi_i(\vec{r}_i) \rightarrow \sum_{j(\neq i)} \sum_{s_j} \int d^3 \vec{r}_j \frac{e^2}{|\vec{r}_i - \vec{r}_j|} \underbrace{\left[|\phi_j(\vec{r}_j, s_j)|^2 \phi_i(\vec{r}_i, s_i) - \phi_j^*(\vec{r}_j, s_j) \phi_j(\vec{r}_i, s_i) \phi_i(\vec{r}_j, s_j) \delta_{s_i s_j} \right]}_{\text{Austauschterm}}. \quad (7.12)$$

Bei diesem Term handelt es sich um ein **nicht-lokales Potential**. Es ist wiederum selbstkonsistent zu bestimmen, numerisch wieder iterativ wie beim Hartree-Verfahren.

Bemerkungen:

- Die Hartree-Fock-Gleichungen für $s_i = \pm 1$ sind *gleich* auf Grund der Symmetrie $s_i \leftrightarrow -s_i$.
- Der Austauschterm wirkt nur zwischen Elektronen mit *gleichem* Spin $s_i = s_j$, da nur dann das Pauli-Prinzip relevant ist.
- Der Austauschterm *senkt* die Energie (negatives Vorzeichen); das Pauli-Verbot spart also Coulomb-Energie.

7.3.2 Mean-Field-Theorie des Ising-Modells

Das Ising-Modell ist ein einfaches Modell um magnetische Ordnung und **Phasenübergänge** zu untersuchen. Wir betrachten ein Gittermodell mit N Gitterplätzen. Auf jedem Gitterplatz $i = 1 \dots N$ sei eine Spinvariable s_i lokalisiert, welche die beiden Zustände $s_i = \pm 1$ annehmen kann. Die Eigenwerte des zugehörigen quantenmechanischen Operators lauten $\hat{s}_{i,z} = \frac{\hbar}{2}s_i$. Zusätzlich wirkt ein äußeres Magnetfeld mit der Feldstärke H . Die Hamiltonfunktion des **Ising-Modells** (1925) lautet:

$$\mathcal{H}_{\text{Ising}} = -\frac{1}{2} \sum_i \sum_{j \neq i} J_{ij} s_i s_j - H \sum_i s_i. \quad (7.13)$$

Wir betrachten das Ising-Modell auf einem kubischen Gitter in D Dimensionen. Im hier als homogen angenommenen Magnetfeld H entspricht der zweite Term in Gleichung (7.13) der **Zeeman-Energie**. J_{ij} bezeichnet die **magnetische Kopplung**, die z.B. durch die Austauschwechselwirkung vermittelt wird. J_{ij} sei isotrop, d.h. $J_{ij} = J_{ji}$. Für die Doppelsumme in Gleichung (7.13) gilt: $1/2 \sum_{i \neq j} = \sum_{\text{Bonds}(ij)}$. Dies entspricht einer Summation über alle Bonds zwischen Gitterplatz i und j , wobei jeder Bond nur einmal gezählt wird. Im Fall $J_{ij} > 0$ spricht man von einem **Ferromagneten**. In diesem wird eine parallele Ausrichtung der Spins bevorzugt. Wir betrachten speziell

$$J_{ij} = \begin{cases} J > 0 & i, j \text{ nächste Nachb.} \\ J = 0 & \text{sonst} \end{cases}$$

Dieses ist ein wichtiges Modell in der statistischen Physik der Phasenübergänge:

- Es zeigt einen Phasenübergang für $D \geq 2$ und ist exakt lösbar für $D = 1$. Für $D = 2$ ist es nur bei $H = 0$ exakt lösbar (**Onsager** 1948).
- Bei $H = 0$ besitzt das Modell einen magnetischen Phasenübergang. Bei hohen Temperaturen liegt eine paramagnetische Phase mit $m = \langle s_i \rangle = 0$ vor, während sich für tiefe Temperaturen spontan eine ferromagnetische Phase mit $m = \langle s_i \rangle \neq 0$ ausbildet.
- Das Ising-Modell ist *das* Gittermodell zur Computersimulation von Phasenübergängen. Zur Simulation kann z.B. ein **Monte-Carlo** Algorithmus genutzt werden, dazu später in Kapitel 11.3 mehr.
- Es gehört zu einer wichtigen **Universalitätsklasse**, die viele äquivalente Probleme beschreibt, wie z.B. die Kondensation eines **Gittergases**, bei der jedem Gitterplatz eine Besetzungszahl $n_i = 0; 1$ zugeordnet ist: $n_i = 0$ entspricht einem leeren und $n_i = 1$ einem besetzten Gitterplatz. Ein weiteres äquivalentes Problem ist die Entmischung einer **binären Legierung** mit den Konstituenten A und B . Hierbei kann ein Gitterplatz entweder von A oder von B belegt sein.
- Der **Ordnungsparameter**, mit dem sich der Phasenübergang charakterisieren lässt, ist die Magnetisierung $m = \langle s_i \rangle$. In der paramagnetischen Phase ist $m = 0$, während in der ferromagnetischen Phase $m \neq 0$.

Die Berechnung der Zustandssumme $Z = \sum_{\{s_i=\pm 1\}} e^{-\beta H_{\text{Ising}}[\{s_i\}]}$ ist schwierig bzw. i.Allg. nicht möglich, wegen der Kopplung benachbarter Spins. Zur Berechnung muss deshalb auf Näherungen zurückgegriffen werden. Die einfachste approximative Theorie die Phasenübergänge erklären kann, ist die **Mean-Field Theorie**. In dieser Näherung wirkt auf den Spin s_i nur das **mittlere Feld** (oder auch **Molekularfeld**) der anderen Spins

$$H_i^{\text{MF}} = H + \sum_{j \neq i} J_{ij} \langle s_j \rangle.$$

Damit kann $\mathcal{H}_{\text{Ising}}$ durch

$$\mathcal{H}_{\text{MF}} = - \sum_i s_i H_i^{\text{MF}} = - \sum_{i \neq j} J_{ij} s_i \langle s_j \rangle - H \sum_i s_i$$

approximiert werden. Durch diese Näherung werden die beiden Spins s_i und s_j entkoppelt. Die MF-Approximation vernachlässigt Korrelationen zwischen den Spins. Nun wird $\langle s_j \rangle$ durch $\langle s_j \rangle_{\text{MF}}$ ersetzt, das **selbstkonsistent** mit \mathcal{H} berechnet wird.

$$\langle s_j \rangle_{\text{MF}} = \frac{1}{Z_{\text{MF}}} \sum_{\{s_j\}} s_j e^{-\beta \mathcal{H}_{\text{MF}}}. \quad (7.14)$$

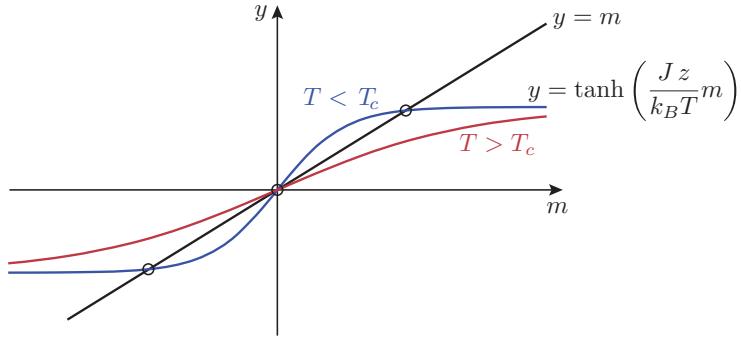


Abbildung 7.7: Grafische Lösung der Selbstkonsistenzgleichung (7.17) für $H = 0$. Die linke Seite $y = m$ und die rechte Seite $y = \tanh\left(\frac{1}{k_B T}(zJm)\right)$ der Gleichung werden in das gleiche Koordinatensystem eingetragen, Schnittpunkte der beiden Funktionen sind Lösungen der Gleichung.

Hierbei ist Z_{MF} die Zustandsumme in MF-Näherung.

$$Z_{\text{MF}} = \sum_{\{s_i=\pm 1\}} e^{-\beta \sum_i H_i^{\text{MF}}} = \prod_{i=1}^N \left(e^{\beta H_i^{\text{MF}}} + e^{-\beta H_i^{\text{MF}}} \right).$$

$$\langle s_j \rangle_{\text{MF}} = \left(e^{\beta H_j^{\text{MF}}} + e^{-\beta H_j^{\text{MF}}} \right)^{-1} \left(\sum_{j=\pm 1} s_j e^{\beta \sum_j s_j H_j^{\text{MF}}} \right)$$

[alle Faktoren $i \neq j$ kürzen sich]

$$= \frac{e^{\beta H_j^{\text{MF}}} - e^{-\beta H_j^{\text{MF}}}}{e^{\beta H_j^{\text{MF}}} + e^{-\beta H_j^{\text{MF}}}} = \tanh(\beta H_j^{\text{MF}}). \quad (7.15)$$

Wenn H homogen ist, dann ist auch $m = \langle s_i \rangle$ homogen. Daraus folgt, dass auch das mittlere Feld

$$\boxed{\mathcal{H}_{\text{MF}} = H + \sum_{i \neq j} J_{ij} m = H + zJm} \quad (7.16)$$

homogen ist. Hier ist z die **Koordinationszahl**, die die Anzahl der nächsten Nachbarn angibt. Für ein kubisches Gitter gilt $z = 2D$. Es folgt

$$\boxed{m = \tanh\left(\frac{1}{k_B T}(H + zJm)\right)}. \quad (7.17)$$

Gleichung (7.17) ist eine **selbstkonsistente MF-Gleichung** für die Magnetisierung m . Sie hat die Form einer Fixpunktgleichung, $m = f(m)$, und kann durch numerisches iterieren gelöst werden.

Wie betrachten den Fall $H = 0$:

- 1) Gleichung (7.17) hat einen trivialen Fixpunkt $m = 0$. Dieser ist stabil und der einzige Fixpunkt nach Banachschem Fixpunktsatz, wenn $|f'(m = 0)| < 1$. Da $f'(0) = \frac{Jz}{k_B T}$, ist dies für hohe Temperaturen $k_B T > Jz \equiv k_B T_c$ der Fall, wobei T_c die **kritische Temperatur** bezeichnet. Abb. 7.7 zeigt eine grafische Lösung der Gleichung (7.17).

Für $T > T_c$ ist $m = 0$ einziger und stabiler Fixpunkt.

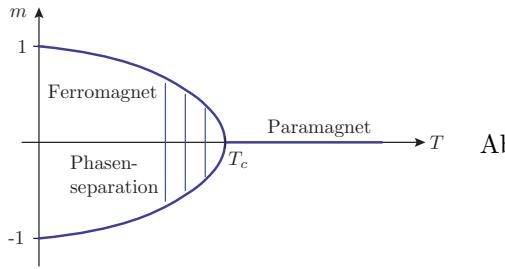


Abbildung 7.8: Phasendiagramm für $H=0$ in der m - T -Ebene. Am Punkt T_c steigt die Funktion $m(T)$ nach links mit dem kritischen Exponenten $\beta = 1/2$ an, also wie eine Wurzelfunktion.

- 2) Bei $T = T_c$ tritt eine **Bifurkation** auf. Der Fixpunkt $m = 0$ wird instabil und zwei weitere stabile Fixpunkte $\pm m(T)$ erscheinen.

$$\frac{J_z m}{k_B T} = \arctan(m) \approx m + \frac{m^3}{3} + \dots$$

Es gilt

$$m^2(T) = \begin{cases} T > T_c : & 0 \\ T < T_c : & 3\left(\frac{T_c}{T} - 1\right), \end{cases}$$

also

$$m(T) \propto \left| \frac{T - T_c}{T_c} \right|^\beta$$

mit einem **kritischen Exponenten**

$$\beta_{MF} = \frac{1}{2}$$

(mehr dazu in Kapitel 11.5).

Wenn die numerische Iteration der Gleichung (7.17) einen stabilen Fixpunkt findet, entspricht dieser normalerweise auch einer thermodynamisch stabilen Phase.

7.4 Iterationen, Bifurkationen und Chaos

Während die vorherigen Iterationen alle das Ziel hatten, gegen den Fixpunkt als Lösung eines bestimmten Problems zu konvergieren, betrachten wir nun die Frage, wie sich Iterationen verhalten können, wenn ein Fixpunkt instabil wird. Wir werden am Beispiel der logistischen Abbildung feststellen, dass sich die Periode in einer Folge von Bifurkationen verdoppelt und eine einfache Iteration auf diese Art in eine chaotische Dynamik übergehen kann. Durch die Verwendung von Computern kann dies leicht untersucht und visualisiert werden.

7.4.1 Iteration der Logistischen Abbildung

Wir betrachten die **logistische Abbildung**

$$f(x) = r x(1 - x), \quad (7.18)$$

welche für $r < 4$ das Intervall $[0, 1] \in \mathbb{R}$ auf sich selbst abbildet (siehe Abb. 7.10). Die Iteration

$$x_{n+1} = r x_n(1 - x_n) \quad (7.19)$$



Abbildung 7.9: Links: Mitchell Feigenbaum (geb. 1944). Rechts: Taschenrechner HP-65 von Hewlett-Packard (erster programmierbarer Taschenrechner, 1974), mit dessen Hilfe Feigenbaum die Konstante (7.22) fand.

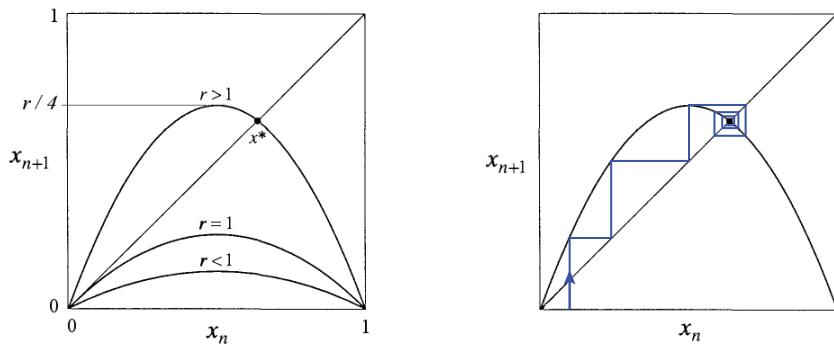


Abbildung 7.10: Links: Verlauf der logistischen Abbildung für verschiedene Werte von r . Rechts: Grafische Iteration (“Spinnennetz-Konstruktion”) der logistischen Abbildung. Von einem Startwert x_n erhält man den nächsten Wert x_{n+1} indem man nach oben bis zum Funktionsgraphen läuft. Um diesen Wert als neuen Startwert zu benutzen, muss er auf die x -Achse übertragen werden, indem man horizontal bis zur Winkelhalbierenden läuft. Dann wieder hoch zum Funktionsgraphen... Sie verstehen es schon. (Quelle [6].)

wurde um 1976 von May und Feigenbaum ausführlich untersucht.

Sie kann als diskrete Version der logistischen Differentialgleichung $\dot{N} = r N(1 - N/K)$ (Verhüst 1838) aufgefasst werden, welches das **einfachste Modell für Populationswachstum** ist: Mit der Wachstumsrate r führt der erste Term $\dot{N} = r N$ zu unbegrenztem exponentiellen Wachstum. Um diesen unrealistischen Effekt zu vermeiden, nimmt man an, dass die Wachstumsrate mit N abnimmt (Überbevölkerung, begrenzte Ressourcen), so dass man eine effektive Wachstumsrate $r_{\text{eff}} = r(1 - N/K)$ definieren kann. Die Größe K ist dabei die Tragkapazität; für $N > K$ nimmt die Bevölkerung nämlich wieder ab.

Die Iteration (7.19) kann sehr anschaulich auf grafische Weise durchgeführt werden (siehe Abb. 7.10). Die Frage ist nun: Für welche Werte von r konvergiert die Iteration gegen einen Fixpunkt, wie stabil ist er, und was passiert wenn er seine Stabilität durch eine Änderung von r verliert?

Generell trifft man diskrete Iterationen der Form

$$x_{n+1} = f(x_n) \quad (7.20)$$

sehr oft an in der Physik. Dies ist z.B. der Fall, wenn man eine echte **diskrete Dynamik** hat wie

bei der getakteter digitaler Elektronik oder bei gepulst angetriebenen mechanischen Systemen. In der nichtlinearen Dynamik oder Chaostheorie stellen **Poincaré-Schnitte** einen sehr fruchtbaren Zugang dar, bei dem der Zustand eines dynamischen Systems "stroboskopisch" zu bestimmten diskreten Zeiten untersucht wird (siehe unten, Kapitel 7.5). Dadurch gewinnt man aus der ursprünglich kontinuierlichen Dynamik ebenfalls eine diskrete Dynamik, die ähnliche Phänomene wie die Iterationen (7.20) zeigt.

Iterationen der Form (7.20) zeigen interessantes Verhalten immer dann, wenn die Abbildung $f(x)$ **nichtlinear** ist, wie das bei der logistischen Abbildung (7.18) der Fall ist. Dann treten Phänomene wie **Bifurkationen**, **Chaos** und **Intermittenz** auf, wie wir im Folgenden am Beispiel der logistischen Abbildung diskutieren werden.

7.4.2 Fixpunkte, Bifurkationen und Chaos

Trivialer Fixpunkt: $0 < r < 1$

Das einfachste (und langweiligste) dynamische Verhalten stellt sich ein, wenn die Iteration gemäß des **Banachschen Fixpunktsatzes** gegen den einzigen vorhandenen Fixpunkt konvergiert. Der Banachraum $[0, 1] \subset \mathbb{R}$ wird durch die logistische Gleichung (7.18) auf sich selbst abgebildet. Sie ist eine Kontraktion falls $|f'(x)| < 1$ auf dem gesamten Intervall, also für $0 \leq r < 1$ (siehe Abb. 7.10). Der Banachsche Fixpunktsatz sagt aus, dass es genau einen Fixpunkt x^* gibt, und die Iteration für jeden Startwert gegen ihn konvergiert. Aus der Funktionsvorschrift und Abbildung 7.10 (links) ist ersichtlich, dass

$$x^* = 0$$

dieser einzige Fixpunkt ist.

Weiterer Fixpunkt: $1 < r < 3$

Für den Physiker wird es interessant, wenn der Mathematiker seinen Fixpunktsatz nicht mehr anwenden kann. Das ist bereits für $r > 1$ der Fall, da $f(x)$ dann keine Kontraktion mehr ist. Tatsächlich ist aus Abb. 7.10 (links) ersichtlich, dass es nun zwei Fixpunkte gibt.

Um die Frage der **Stabilität der Fixpunkte** zu beantworten, betrachten wir eine kleine Störung $x_n = x^* + \varepsilon_n$. Sie beträgt im nächsten Schritt

$$x_{n+1} = f(x^* + \varepsilon_n) \approx f(x^*) + \underbrace{f'(x^*) \varepsilon_n}_{\equiv \varepsilon_{n+1}}.$$

Verkleinert sich die Störung in jedem Schritt, gilt also $|\varepsilon_{n+1}| < |\varepsilon_n|$, ist der Fixpunkt stabil, andernfalls instabil:

Fixpunkt stabil	\iff	$ f'(x^*) < 1$	(7.21)
Fixpunkt instabil	\iff	$ f'(x^*) > 1$	

In unserem Fall sind die beiden Lösungen der Fixpunktgleichung $x = rx(1-x)$ von Hand berechenbar, und Auswertung von $f'(x^*)$ an diesen Stellen zeigt

$$x^* = 0 \text{ (instabil)} \quad \text{und} \quad x^* = 1 - 1/r \text{ (stabil für } 1 < r < 3).$$

Unsere Stabilitätsanalyse gilt eigentlich nur in unmittelbarer Umgebung des Fixpunktes, da die Taylorentwicklung bereits nach dem 1. Glied abgebrochen wurde. Die grafische Iteration (Abb. 7.10 rechts) lässt jedoch erkennen, dass die Iteration für jeden Startwert außer 0 und 1 gegen den Fixpunkt bei $1 - 1/r$ konvergiert.

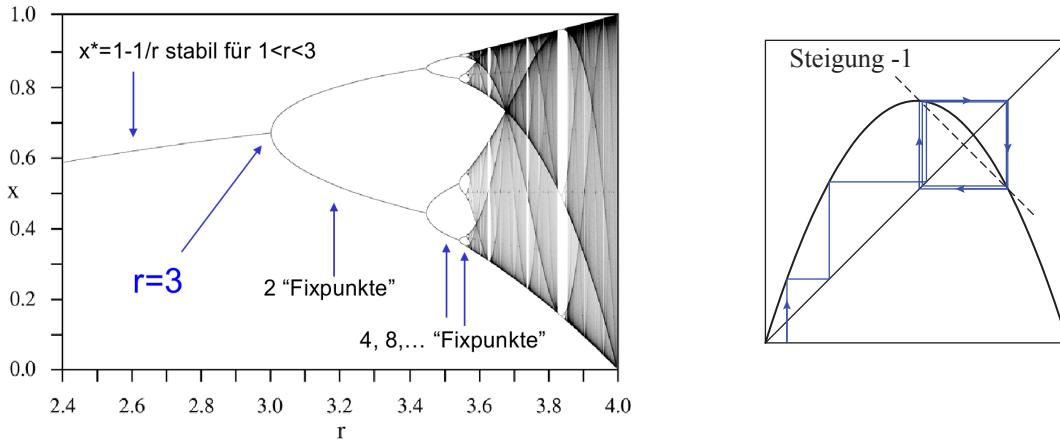


Abbildung 7.11: Links: Bifurkationsdiagramm der logistischen Abbildung. Rechts: Zustandekommen eines Orbits der Periode 2 in der grafischen Iteration. Das “Spinnennetz” kann sich nicht mehr zusammenziehen, wenn die Steigung am Fixpunkt $|f'(x)| > 1$ ist.

Bifurkationen und Periodenverdopplungen: $3 < r < 3.5699\dots$

Für $r > 3$ wird auch der Fixpunkt bei $x^* = 1 - 1/r$ instabil, da $f'(x^*) < -1$. Das Verhalten der Iteration ist jedoch immer noch sehr geordnet, was sich in Computerexperimenten überprüfen lässt (Abb. 7.11). Dieses Diagramm kommt zustande, wenn man für viele Werte von r die Iteration mit einem zufälligen Wert startet, ein paar Schritte “warmläuft” lässt (so dass es Zeit hat zu einem “Fixpunkt” zu konvergieren) und in den anschließenden Schritten alle erreichten x -Werte speichert und in das Diagramm einträgt.

Für $r < 3$ erkennen wir den einen stabilen Fixpunkt (s. Abb. 7.11). Für $r > 3$ scheint es zwei “Fixpunkte” zu geben. Diese entstehen nicht etwas aus zwei verschiedenen Startwerten der Iteration, sondern werden abwechselnd nacheinander von der Iteration angenommen. Diese Oszillation der Iteration zwischen zwei Werten nennt man einen **Orbit der Periode 2**. In der grafischen Veranschaulichung der Iteration (Abb. 7.11 rechts) wird klar, wie er zustande kommt: Das “Spinnennetz” zieht sich nicht mehr zusammen.

Da jeder Punkt des Orbits genau nach zwei Iterationsschritten wieder erreicht wird, ist er ein Fixpunkt der **zweifach iterierten Abbildung**

$$f^2(x) = f(f(x)) = r[r x(1-x)](1-[r x(1-x)]).$$

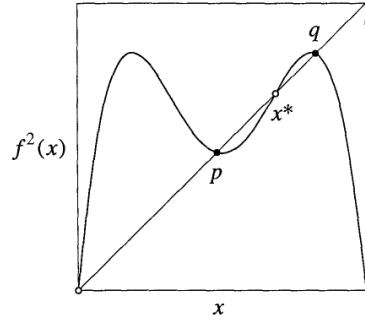
Sie ist ein Polynom vierten Grades (siehe Skizze) und hat die Fixpunkte

$$\begin{aligned} x^* = 0, \quad x^* = 1 - 1/r &\quad (\text{instabil für } r > 3) \\ p, q = \frac{r+1 \pm \sqrt{(r-3)(r+1)}}{2r} &\quad (\text{nur für } r > 3, \text{ dann stabil bis } r \approx 3.449\dots). \end{aligned}$$

Denkt man von vornherein in der Sprache der zweifach iterierten Abbildung f^2 , so hat sie für $1 < r < 3$ nur die Fixpunkte $x^* = 0$ und $1 - 1/r$, welche auch die einfache Abbildung schon hat. Letzterer wird bei $r = 3$ instabil, während gleichzeitig unmittelbar links und rechts von ihm je ein neuer, stabiler Fixpunkt (p und q) entsteht. Diesen Vorgang, ein stabiler Fixpunkt wird instabil und “spaltet” dabei in zwei stabile Fixpunkte auf, nennt man **Bifurkation**. Die beiden stabilen Fixpunkte von $f^2(x)$ werden vom Iterationsverfahren $x_{n+1} = f^2(x_n)$ gefunden; welcher genau, hängt vom gewählten Startwert ab. Nur wenn man einen Schritt zurück geht und die Iteration der einfachen Abbildung betrachtet, $x_{n+1} = f(x_n)$, wechseln sich diese beiden Punkte ab und man erhält einen Orbit. Die Fixpunkte p und q von $f^2(x)$ erfüllen dann $q = f(p)$ und $p = f^2(p) = f(q)$.

Es gibt weitere Bifurkationen für größere Werte von r , wie in Abbildung 7.11 zu sehen. Nennen wir den Wert von r , bei welchem ein Orbit der Periode 2^n stabil wird r_n . Analog zum Fall der Periode 2 entspricht ein Orbit der Periode 2^n dabei einem Paar ($n = 1$), Quartett ($n = 2$), Oktett ($n = 3$)... von stabilen Fixpunkten der Abbildung f^{2^n} . Eine Übersicht über die Stellen, an denen die Bifurkationen auftreten ist nachfolgend gegeben:

$r_1 = 3$	(Periode 2 wird geboren)
$r_2 = 3.449\dots$	4
$r_3 = 3.54409\dots$	8
$r_4 = 3.5644\dots$	16
$r_5 = 3.568759\dots$	32
\vdots	\vdots
$r_\infty = 3.569946\dots$	∞



Die Annäherung an den Grenzwert r_∞ geschieht dabei ungefähr wie in einer geometrischen Reihe¹ mit der **universellen Feigenbaum-Konstante**

$$\delta = \lim_{n \rightarrow \infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} = 4.669\dots \quad (7.22)$$

Chaos und Intermittenz: $r > 3.5699\dots$

Feigenbaum hat gezeigt, dass die Periodenverdopplungskaskade bei $r_\infty = 3.569946\dots$ ins **Chaos** übergeht. Im Chaos gibt es allgemein keine periodischen Orbits mehr, d.h. keine endliche Sequenz wiederholt sich bei der Iteration. Im Bifurkationsdiagramm (Abb. 7.11) liegen die geplotteten Punkte dicht auf der x -Achse, in gewissen Bändern offensichtlich “dichter” als im übrigen Bereich.

Doch auch im Chaos gibt es noch einige Stellen, an denen man Ordnung erkennen kann, sogenannte **periodische Fenster**. Am prominentesten ist das Fenster der Periode 3 im Bereich $3.8284 < r < 3.8415$. Man beachte, dass es vorher immer nur Orbits der Periode 2^n gab, dieser Orbit der Periode 3 passt nicht in dieses Schema.

Die Entstehung dieses Orbits aus dem Chaos heraus wird durch **Intermittenz** geprägt, in denen das System sich lange Zeit schon fast wie in einem Periode-3-Orbit verhält, doch zwischenzeitlich wieder ins Chaos ausbricht. Der Verlauf der Iteration für r knapp unterhalb des periodischen Fensters ist in Abbildung 7.12 oben gezeigt. Graphisch kann das Verhalten erklärt werden, wenn man die Spinnennetzkonstruktion für die dreifach iterierte Abbildung $f^3(x)$ durchführt (siehe 7.12 unten). Sie besitzt drei Stellen, welche kurz davor sind die Winkelhalbierende zu schneiden und somit zu Fixpunkten zu werden. Da sie aber noch einen kleinen Abstand zur Winkelhalbierenden haben, bilden sie mit ihr einen schmalen “Kanal”, welcher vom Spinnennetz durchlaufen werden muss. Innerhalb des Kanals schreitet die Iteration von $f^3(x)$ nur in sehr kleinen Schritten voran. In diesen Schritten sieht es so aus, als hätte man einen Fixpunkt von $f^3(x)$ erreicht, welcher in der Iteration zur einfachen Abbildung $f(x)$ als Orbit der Periode 3 erscheint. Nach dem Kanaldurchgang bricht das System ins Chaos aus, bis es irgendwann wieder in die Mündung eines Kanals läuft.

¹Feigenbaum fand dies heraus, weil er einen ziemlich langsamen programmierbaren Taschenrechner (ein HP-65, siehe Abb. 7.9) für seine numerischen Experimente benutzte. Während der Taschenrechner beschäftigt war, hatte er genügend Zeit zu raten, welches das nächste r_n sein wird. Er fand, dass sich der Abstand aufeinanderfolgender r_n bei jedem Mal um den Faktor $1/4.669\dots$ reduziert.

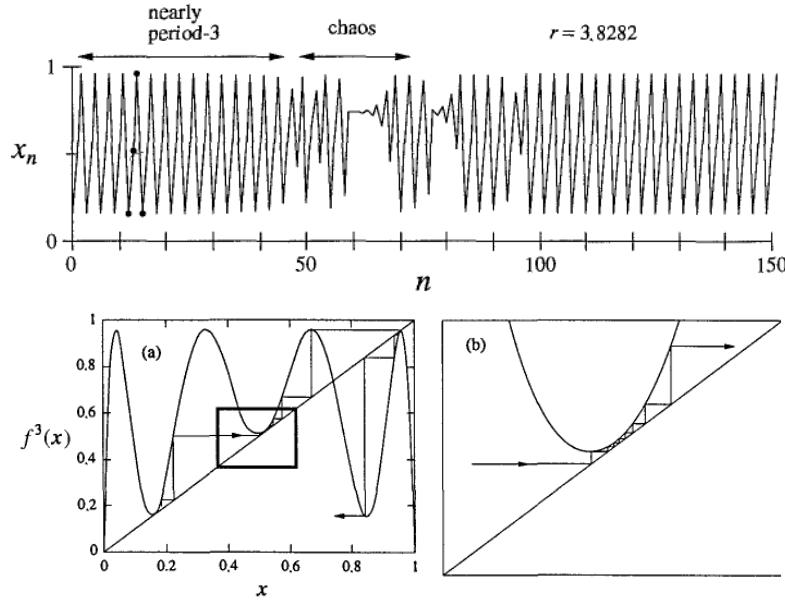


Abbildung 7.12: Intermittenz. Oben: Verlauf der Iteration knapp unterhalb des periodischen Fensters. Unten: Grafische Veranschaulichung der Intermittenz. (Quelle [6].)

Der Ausgang des Systems aus dem Fenster der Periode 3 heraus geschieht mit verblüffender Ähnlichkeit zum anfänglichen Periodenverdopplungsszenario (siehe Abb. 7.13). Es gibt wieder zunächst eine Periodenverdopplungskaskade, gefolgt von Chaos welches zwischenzeitlich wieder von periodischen Fenstern (deren Ordnung und relative Position zueinander wie im großen Diagramm sind) unterbrochen wird.

7.4.3 Selbstähnlichkeit und Universalität

Das Bifurkationsdiagramm der logistischen Abbildung (Abb. 7.11 (links)) hat eine bemerkenswerte Eigenschaft, die in Abb. 7.13 verdeutlicht ist. Wird ein Teil des Diagramms herausgeschnitten und vergrößert ähnelt er wieder dem gesamten Diagramm. Wir werden im nächsten Abschnitt versuchen, diese **Selbstähnlichkeit** des Bifurkationsdiagramms mit Hilfe der Renormierungsgruppe zu verstehen.

Eine weitere erstaunliche Eigenschaft ist die **Universalität** des Bifurkationsverhaltens. Die gleiche Feigenbaumkonstante, die für die logistische Abbildung gefunden wurde, beschreibt die Bifurkationsdiagramme für **alle unimodalen Abbildungen** auf $[0, 1]$, d.h. glatte konkave Abbildungen mit einem Maximum. Dies verdeutlicht Abb. 7.14. Auch diese Universalität wollen wir im nächsten Abschnitt mit Hilfe der Renormierungsgruppe verstehen.

Noch erstaunlicher ist, dass die Universalität der Feigenbaumkonstante noch weiter reicht und sie auch in realen physikalischen Systemen auftaucht, zum Beispiel in chaotischen Systemen der Fluidodynamik. Erwärmst man einen mit Quecksilber gefüllten Behälter von unten, so muss das Quecksilber Wärme von unten nach oben transportieren. Bis zu einer kritischen Rayleigh-Zahl R (eine dimensionslose Kennzahl proportional zum Temperaturgradienten) geschieht dies ohne makroskopische Bewegung. Ab R_c bilden sich **Konvektionsrollen**, in denen das heiße Quecksilber auf der einen Seite nach oben strömt, seine Temperatur dort abgibt und auf der anderen Seite wieder absinkt. Misst man in solchen Zuständen an einem festen Ort den zeitlichen Temperaturverlauf, so erhält man eine periodische Funktion (siehe Abb. 7.15), deren Periode sich bei gewissen Werten von R/R_c verdoppelt. Aus diesen Werten lässt sich wieder die Feigenbaumkonstante berechnen.

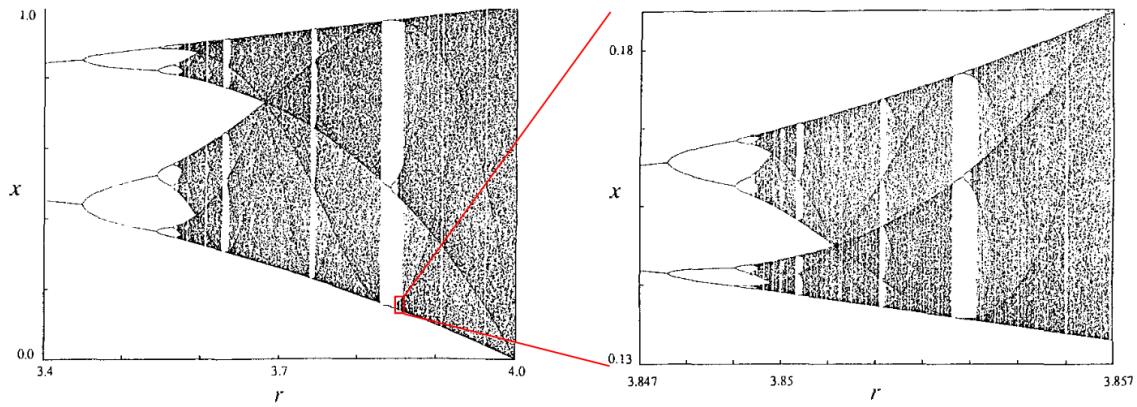


Abbildung 7.13: Selbstähnlichkeit des Bifurkationsdiagramms der logistischen Abbildung auf verschiedenen Skalen. (Quelle [6].)

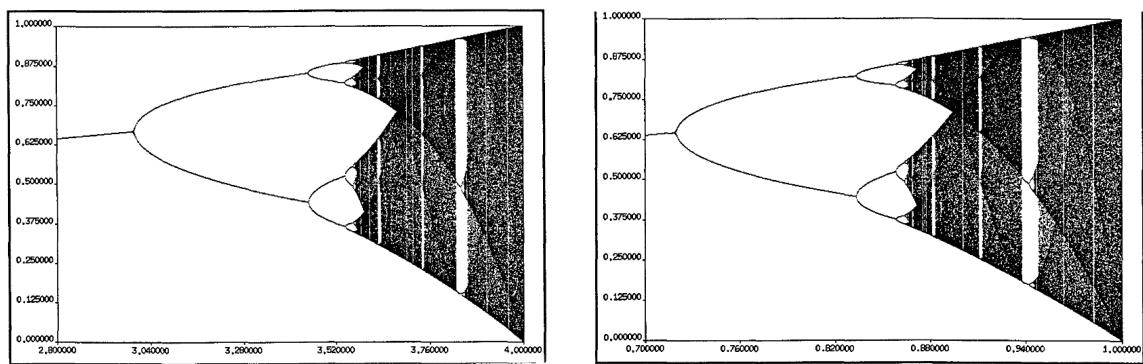


Abbildung 7.14: Universalität des Bifurkationsdiagramms für unimodale Abbildungen. Gezeigt sind Diagramme für die Logistische Abbildung $f(x) = r x(1 - x)$ (links) und die Sinusfunktion $f(x) = r \sin(\pi x)$ (rechts). (Quelle [6].)

Dieses Vorgehen lässt sich in vielen chaotischen Systemen anwenden, und führt immer auf Konstanten die in der Nähe von Feigenbaums $\delta = 4.669\dots$ liegen:

	Experiment	Anzahl Verdopplungen	δ
<i>Hydrodynamik</i>			
Wasser	4		4.3 ± 0.8
Quecksilber	4		4.4 ± 0.1
<i>Elektronik</i>			
Diode	4		4.5 ± 0.6
andere Diode	5		4.3 ± 0.1
Transistor	4		4.7 ± 0.3
Josephson Simul.	3		4.5 ± 0.3

Man kann daher sagen, dass die Feigenbaumkonstante tatsächlich eine Art “Naturkonstante” für Bifurkationen darstellt.

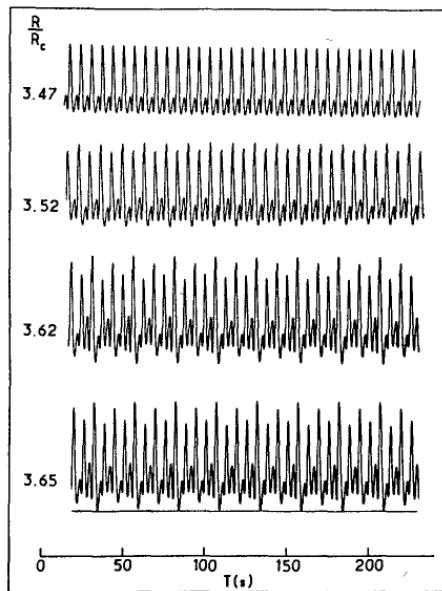


Figure 10.6.5 Libchaber et al. (1982), p. 213

Abbildung 7.15: Periodenverdopplung im zeitlichen Temperaturverlauf in einem Quecksilber-System mit Konvektionsrollen (Quelle [6]). Der oberste Zustand mit gleich hohen Peaks kann als Periode 1 gedeutet werden. Ab einem zu hohen Wert R/R_c zerfallen die Peaks in zwei Sorten; eine ist etwas höher als die andere. Weiteres Aufspalten in mehrere Sorten von Peaks kann bei noch höheren R/R_c -Werten beobachtet werden.

7.4.4 Renormierungsgruppe

Die zunächst überraschende Universalität und Selbstähnlichkeit der Bifurkationsdiagramme für unimodale Abbildungen lässt sich durch eine **Renormierungsgruppentransformation** verstehen [7], welche wir im Folgenden auf einem recht intuitiven Niveau behandeln werden [6].

Die Idee dabei ist, dass bei einer Periodenverdopplung die

- Iteration der Abbildung $f^{2^n}(x) \rightarrow f^{2^{n+1}}(x)$
- und anschließende Reskalierung von x - und f -Achse

zu einer sehr ähnlichen Funktion führt. Diese Renormierungsgruppentransformation wird wiederholt ausgeführt und ist somit eine Iteration auf dem Raum der unimodalen Funktionen auf $[0, 1]$. Diese Iteration in einem Funktionenraum führt auch auf einen Fix“punkt”, besser gesagt eine “Fixfunktion”.

Ein wesentlicher Bestandteil der Renormierungsgruppentransformation sind **superstabile Fixpunkte**. Bei unserer Definition stabiler und instabiler Fixpunkte (7.21) haben wir gesehen, dass die Konvergenz gegen einen Fixpunkt x^* umso schneller ist, je kleiner $|f'(x^*)|$ an dieser Stelle ist. Im Bestfall $f'(x^*) = 0$ bezeichnen wir den Fixpunkt als **superstabil**. Die Konvergenz der Iteration ist dann mindestens vom Grad 2. Geometrisch bedeutet $f'(x^*) = 0$, dass der **Fixpunkt zugleich ein Maximum oder Minimum** der Funktion ist.

Präzisieren wir unseren Renormierungsschritt anhand Abb. 7.16. Gegeben sei eine unimodale Abbildung $f(x, r)$ mit Parameter r und Maximum bei x_m . Das Koordinatensystem wird um x_m zentriert, so dass die nachfolgenden Reskalierungen einfacher werden. Wir wählen $r = R_0$ so, dass x_m ein superstabilen Fixpunkt ist, siehe Abb. 7.16 (a). Der Renormierungsschritt besteht nun darin, die Funktion zweimal anzuwenden (f^2), $r = R_1$ so zu wählen dass x_m wieder ein superstabilen Fixpunkt (nunmehr der Abbildung f^2) ist, und die x - und f -Achse mit einem Faktor α zu reskalieren (siehe Abb. 7.16 (b)). Die so erhaltene Funktion ist der ursprünglichen Funktion sehr ähnlich:

$$f(x, R_0) \approx \alpha f^2\left(\frac{x}{\alpha}, R_1\right).$$

Wiederholt man diese Konstruktion, erhält man wieder eine sehr ähnliche Funktion, welche sich

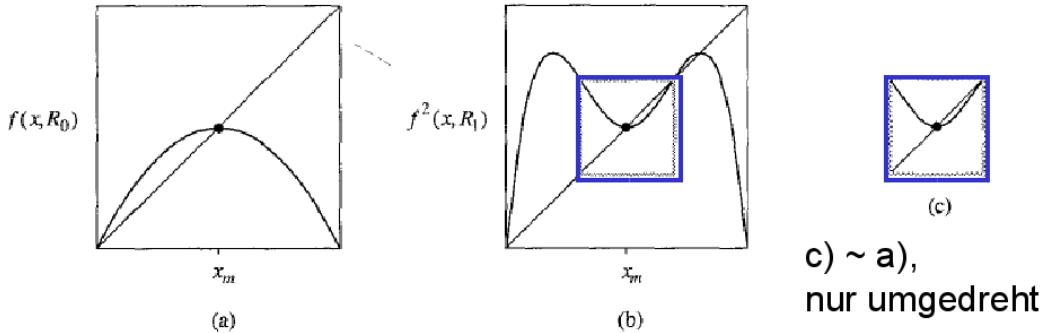


Abbildung 7.16: Veranschaulichung des Renormierungsschrittes

nun noch weniger von ihrem Vorgänger unterscheidet als diese ersten beiden. Diese Iteration läuft demnach gegen eine **universelle Fixpunkt-Funktion** mit superstabilem Fixpunkt,

$$g_0(x) \equiv \lim_{n \rightarrow \infty} \alpha^n f^{2^n} \left(\frac{x}{\alpha^n}, R_n \right).$$

Alternativ kann man auch mit einem anderen Parameter $r = R_i$ für die erste Funktion starten, so dass die Funktion $f(x, R_i)$ bei x_m keinen superstabilen Fixpunkt hat, sondern x_m auf einem Orbit der Periode 2^i durchläuft. Dann laufen die Renormierungsschritte gegen

$$g_i(x) \equiv \lim_{n \rightarrow \infty} \alpha^n f^{2^n} \left(\frac{x}{\alpha^n}, R_{n+i} \right).$$

Wie können wir den Grenz-“Wert”, bzw. die Grenz-Funktion dieser Folgen berechnen? Indem wir ausnutzen, dass die gesuchte Grenz-Funktion sich nicht ändert, wenn wir den Renormierungsschritt noch einmal auf sie anwenden. Dies funktioniert wenn man gleichzeitig $n \rightarrow \infty$ und $i \rightarrow \infty$ schickt, und man erhält die Funktionalgleichung

$$g_\infty(x) = \alpha g_\infty \left(\frac{x}{\alpha} \right). \quad (7.23)$$

mit den Randbedingungen $g'(0) = 0$ (weil bei $x_m = 0$ ein superstabilen Fixpunkt sein soll) und $g(0) = 1$ (beliebig).

Der Wert für α ist durch diese Funktionalgleichung und deren Randbedingung ebenfalls bestimmt, $1 = g(0) = \alpha g(g(0)) = \alpha g(1)$. Feigenbaum berechnete eine Näherungslösung der Funktionalgleichung durch einen Potenzreihenansatz $g(x) = 1 + c_2 x^2 + c_4 x^4$ mit dem Ergebnis

$$\alpha = 1/g(1) = -2.5029\dots \quad (7.24)$$

Die Konstante α ist eine ähnliche universelle Konstante wie die Feigenbaum-Konstante δ . Während die Feigenbaum-Konstante die Verhältnisse von Differenzen $\Delta_n = r_n - r_{n-1}$ aufeinanderfolgender Bifurkationsparameter r_n beschreibt, $\delta = \lim_{n \rightarrow \infty} \Delta_n / \Delta_{n+1}$ (siehe (7.22)), beschreibt α das Verhältnis von Differenzen aufeinanderfolgender Abstände d_n zwischen dem superstabilen Fixpunkt x_m und dem benachbarten Fixpunkt von $f^{2^n}(x, R_n)$ bei $r = R_n$ (siehe Abb. 7.17), also $\alpha = \lim_{n \rightarrow \infty} d_n / d_{n+1}$. Dementsprechend kann auch die universelle Feigenbaum-Konstante $\delta = 4.669\dots$ auf ähnliche Weise aus der Renormierung bestimmt werden. Die Rechnung ist allerdings komplizierter als für α . Auch δ wird vollständig durch die Fixpunkt-funktion $g_\infty(x)$ bestimmt.

Viele verschiedene Ausgangsfunktionen $f(x, r)$ konvergieren unter der Renormierungsgruppentransformation gegen die gleiche Fixpunkt-Funktion $g_\infty(x)$, deren Gleichung (7.23) keine Information über $f(x, r)$ mehr enthält. Dies erklärt die **Universalität der Feigenbaumkonstante**: Sie ist

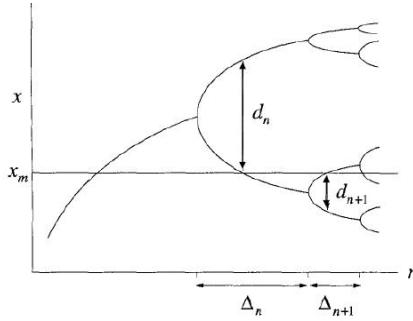


Abbildung 7.17: Veranschaulichung der universellen Konstanten α und δ .

nicht abhängig von der genauen Gestalt der unimodalen Funktion f , welche iteriert wird, sondern nur von der Fixpunkt-Funktion $g_\infty(x)$ welche durch (7.23) definiert ist.

Wir sehen auch, dass wir unabhängig von der anfänglichen Periodenlänge 2^i des superstabilen Fixpunktes $r = R_i$ immer bei der gleichen Fixpunkt-Funktion landen. Dies erklärt die **Selbstähnlichkeit** des Bifurkationsdiagramms: Egal, bei welcher Verzweigungs-“Generation” i wir beginnen, nähert sich die Renormierungsgruppe der gleichen Fixpunkt-Funktion $g_\infty(x)$.

7.5 Poincaré-Schnitte in chaotischen Systemen

Poincaré-Schnitte bilden die kontinuierliche Dynamik eines mechanischen Systems auf eine diskrete iterative Dynamik ab. Das chaotische Verhalten lässt sich so anschaulich visualisieren. Für eindimensionale Systeme gibt es eine Analogie zu Bifurkationen und Chaos bei einfachen Iterationen.

7.5.1 Integrable Systeme

Integrable Systeme der klassischen Mechanik sind nicht chaotisch, sondern im Gegenteil exakt lösbar weil “genügend” Erhaltungsgrößen bekannt sind.

Integrable Systeme werden durch folgenden wichtigen Satz von Liouville aus der klassischen Mechanik klassifiziert:

Sind in einem mechanischen System mit f Freiheitsgraden ($2f$ -dimensionaler Phasenraum) f **unabhängige Erhaltungsgrößen in Involution** bekannt, so ist das System durch **Winkel- und Wirkungsvariable** integrierbar.

Wir beweisen diesen Satz nicht (Beweis siehe z.B. Arnold [8]), sondern machen uns lediglich seinen Inhalt etwas präziser klar.

Zunächst erläutern wir einige Begriffe in obigem Satz. Wir betrachten ein autonomes System, d.h. eine zeitunabhängige Hamiltonfunktion $\partial H/\partial t = 0$, also $H = H(\vec{q}, \vec{p})$. Dann ist eine Observable $F_i = F_i(\vec{q}, \vec{p})$ eine **Erhaltungsgröße**, wenn die Poissonklammer mit H verschwindet

$$\{F_i, H\} = \sum_{n=1}^f \left(\frac{\partial F_i}{\partial q_n} \frac{\partial H}{\partial p_n} - \frac{\partial F_i}{\partial p_n} \frac{\partial H}{\partial q_n} \right) = 0$$

Zwei Größen F_i und F_j heißen **unabhängig**, wenn ihre ($2f$ -dimensionalen) Gradienten im Phasenraum linear unabhängig sind. Zwei Größen F_i und F_j heißen **in Involution** (“miteinander



Abbildung 7.18: Links: Joseph Liouville (1809-1882), französischer Mathematiker. Rechts: Henri Poincaré (1854-1912), französischer Mathematiker und Physiker. (Quelle: Wikipedia).

verträglich”), wenn ihre Poissonklammer verschwindet:

$$\{F_i, F_j\} = 0$$

Winkel- und Wirkungsvariablen folgen später.

Damit können wir den **Satz über integrale Systeme** von Liouville etwas präziser fassen [8]:

Für ein autonomes System mit f Freiheitsgraden und Hamiltonfunktion $H(\vec{q}, \vec{p})$ seien neben $F_1 = H$ noch $f - 1$ weitere unabhängige Erhaltungsgrößen $F_i = F_i(\vec{q}, \vec{p})$ in Involution bekannt, so dass

$$\{F_i, F_j\} = 0 \quad \forall i = 1, \dots, f$$

- 1) Dann bewegt sich das System im $2f$ -dimensionalen Phasenraum \mathbb{P} auf der f -dimensionalen Niveaumenge

$$M_{\vec{f}} = \{\vec{x} \in \mathbb{P} : F_i(\vec{x}) = f_i, i = 1, \dots, f\}$$

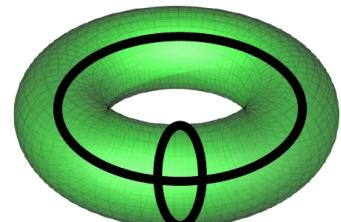
- 2) Ist diese kompakt (also “begrenzt”) und zusammenhängend, dann ist sie diffeomorph (eindeutig und differenzierbar abbildbar) zu einem **f -dimensionalen Torus im $2f$ -dimensionalen Phasenraum**

- 3) Die Bewegung auf diesem Torus kann durch kanonische **Winkel- und Wirkungsvariablen** beschrieben werden.

Der f -dimensionale Torus, der durch die f Erhaltungsgrößen F_i festgelegt wird und auf dem die Bewegung stattfindet, heißt auch **invarianter Torus**. Man beachte, dass die Bewegung eines integrablen Systems damit sehr stark eingeschränkt ist: Im $2f$ -dimensionalen Phasenraum sind der Bewegung f Dimensionen genommen.

Ein f -dimensionaler Torus ist das Produkt von f 1-dimensionalen Kreisen S_1 : $T_f = S_1 \times \dots \times S_1$ (f -mal). Die Abb. rechts zeigt einen 2-dimensionalen Torus eingebettet in 3 Raumdimensionen.

Jeden der f Kreise kann man mit einem **Winkel** φ_i und einem Radius F_i parametrisieren. Winkel und Radius sind normalerweise keine kanonischen Variablen.



blen wie q_i und p_i , d.h. für ihrer Poissonklammern gilt i.Allg. $\{\varphi_i, F_i\} \neq 1$ (d.h. sie gehen *nicht* durch eine kanonische Transformation aus den \vec{q} und \vec{p} hervor). Man kann aber zum Winkel φ_i eine **Wirkungsvariable** $I_i = I_i(F_1, \dots, F_f)$ finden, so dass die Variablen φ_i und I_i **kanonisch** sind (Beweis siehe z.B. Arnold [8]). Dies sind dann die **Winkel- und Wirkungsvariablen** mit folgenden Eigenschaften:

- 1) Die Wirkungsvariablen $I_i = I_i(F_1, \dots, F_f)$ sind wieder Erhaltungsgrößen.
- 2) Die Winkelvariablen φ_i sind zyklisch, also $H = H(I_1, \dots, I_f)$ hängt nicht von den φ_i ab und 2π -periodisch. Man kann die Bewegungsgleichungen dann explizit lösen

$$\begin{aligned}\dot{I}_i &= -\frac{\partial H}{\partial \varphi_i} = 0 \\ \dot{\varphi}_i &= \frac{\partial H}{\partial I_i} = \omega_i(I_1, \dots, I_f) = \text{const} \\ \Rightarrow \varphi_i(t) &= \varphi_i(0) + \omega_i(I_1, \dots, I_f)t\end{aligned}$$

Ein (triviales) **Beispiel** für einen invarianten Torus stellen zwei ungekoppelte Oszillatoren dar mit $H = p_1^2/2m_1 + p_2^2/2m_2 + \omega_1^2 q_1^2/2 + \omega_2^2 q_2^2/2$. Die Energien

$$E_i = \frac{p_i^2}{2m_i} + \frac{1}{2}\omega_i^2 q_i^2 \quad (i = 1, 2) \quad (7.25)$$

stellen dann 2 Erhaltungsgrößen dar für ein autonomes System mit $f = 2$ Freiheitsgraden. Damit ist das System integrabel. Die Gleichungen (7.25) beschreiben jeweils eine Ellipse, d.h. das System bewegt sich im 4-dimensionalen Phasenraum auf dem Produkt zweier Ellipsen, was offensichtlich diffeomorph ist zu einem 2-dimensionalen invarianten Torus im 4-dimensionalen Phasenraum.

Die explizite kanonische Transformation auf Winkel- und Wirkungsvariablen sieht dann so aus:

$$I_i = \frac{1}{\omega_i} E_i(q_i, p_i) \quad \text{und} \quad \varphi_i = \arctan\left(\frac{\omega_i q_i}{p_i}\right)$$

Damit wird dann $H = H(I_1, I_2) = \omega_1 I_1 + \omega_2 I_2$. Die kanonische Transformation $q_i \rightarrow Q_i \equiv \varphi_i$ und $p_i \rightarrow P_i \equiv I_i$ wird durch eine Erzeugende $M = M(\vec{q}, \vec{Q}) = \sum_i \frac{1}{2} \omega_i q_i^2 \cot Q_i$ generiert, wie man nachrechnen kann.

Folgende **integrable Systeme** sind uns aus den Grundvorlesungen bekannt:

- 1) **Autonome Systeme mit $f = 1$** , also eindimensionale Bewegungen mit Energieerhaltung:
Dann ist die Energie als Erhaltungsgröße ausreichend, um das System integrabel zu machen nach obigem Satz von Liouville. Die Lösung solcher eindimensionalen Bewegungen gelingt tatsächlich immer über den *Energieerhaltungssatz und Trennung der Variablen*.
- 2) **Lineare Systeme**, also Systeme, wo die Bewegungsgleichungen linear sind (gekoppelte Oszillatoren, lineare Ketten, Wellen):
Diese Systeme lassen sich durch Einführung von *Normal- oder Eigenschwingungen* auf entkoppelte Oszillatoren zurückführen. Dies funktioniert prinzipiell für beliebig große Zahlen f von Freiheitsgraden.
- 3) Systeme mit genügend Erhaltungsgrößen, wie das **Teilchen im Zentralfeld (Kepler-Problem)** in $f = 3$ Raumdimensionen:
Hier ergeben H , der Drehimpulsbetrag $|\vec{L}|$ und eine Komponente L_i des Drehimpulses 3 Erhaltungsgrößen in Involution (die drei Drehimpulskomponenten sind allerdings *nicht* in Involution).

Typische Beispiele **nicht-integrabler Systeme** sind:

- 1) Systeme mit $f > 1$ Freiheitsgraden, in denen **nichtlineare Terme** in den Bewegungsgleichungen auftreten. Daher wird auch oft die Bezeichnung **nichtlineare Dynamik** für das ganze Feld der Physik verwendet, dass sich mit chaotischen Bewegungen befasst.
- 2) Dissipative (Systeme *ohne* Energieerhaltung) nichtlineare Systeme können auch bereits für $f = 1$ chaotisches Verhalten zeigen, wie wir am Beispiel des gedämpften getriebenen nichtlinearen Oszillators unten sehen werden.
- 3) Auch mit Zentralkräften sind Systeme mit mehr als 2 Teilchen (das berühmte 3-Körper-Problem) nicht mehr integrabel

Solche Systeme können dann im Normalfall nur noch numerisch untersucht werden durch numerische Integration der Bewegungsgleichungen mit Hilfe der Methoden, die wir in Kapitel 4 kennengelernt haben.

Nicht-integrable Systeme kann man im Regelfall nur noch numerisch untersuchen. Es gibt allerdings einige äußerst wichtige mathematische Sätze, die das “Abgleiten” eines integrablen Systems in chaotisches Verhalten qualitativ charakterisieren, wenn es nicht-integrabel gestört wird. Diese werden wir später noch kurz andiskutieren.

7.5.2 Poincaré-Schnitt

Die Dynamik eines mechanischen Systems im vollen $2f$ -dimensionalen Phasenraum ist nur schwer zu visualisieren (für $f > 1$). Ein hilfreiches Werkzeug dabei sind **Poincaré-Schnitte**. Dies gilt besonders für $f = 2$. Poincaré-Schnitte sind auch sehr aufschlussreich, um das Abgleiten eines gestörten integrablen Systems in chaotisches Verhalten zu untersuchen und zu visualisieren, wie wir noch sehen werden.

Dazu definieren wir eine $(2f - 1)$ -dimensionale **Hyperfläche** S im Phasenraum zusammen mit einem Richtungssinn, die durch die Richtung eines Normalenvektors auf der Fläche angegeben werden kann. Dann wird der Punkt auf der Hyperfläche markiert, in dem die Trajektorie im Phasenraum die Hyperfläche in gewünschter Richtung schneidet. Dies ergibt eine diskrete Punktfolge P_0, P_1, \dots und führt zu einer **diskreten Dynamik**: Beim Poincaré-Schnitt wird das System “stroboskopisch” betrachtet, immer wenn es die Hyperfläche S schneidet.

Die entstehende Abbildung

$$P_n \rightarrow T(P_n) = P_{n+1} \quad (7.26)$$

der Schnitthyperfläche auf sich selbst heißt **Poincaré-Abbildung**. Sie kann in der Regel nicht analytisch angegeben werden, allerdings kann der Poincaré-Schnitt numerisch gut durchgeführt werden.

Mit Hilfe von Poincaré-Schnitten lassen sich integrable Systeme von chaotischen unterscheiden. Bei einem integrablen System bewegt sich das System periodisch auf einem invarianten Torus. Die Hyperfläche S sollte diesen Torus schneiden und die Durchstoßpunkte liegen dann auf dem kreisartigen Schnitt von Torus und Hyperebene. Eine kreisartige Topologie der Poincaré-Schnitte sind ein Kennzeichen integrabler Systeme. Für chaotische Systeme gibt es dagegen keine invarianten Tori und Poincaré-Schnitte geben typischerweise eine ungeordnete Ansammlung von Punkten (evtl. mit fraktalen Eigenschaften), die die Hyperfläche ausfüllt.

Wir betrachten zwei Beispiele.

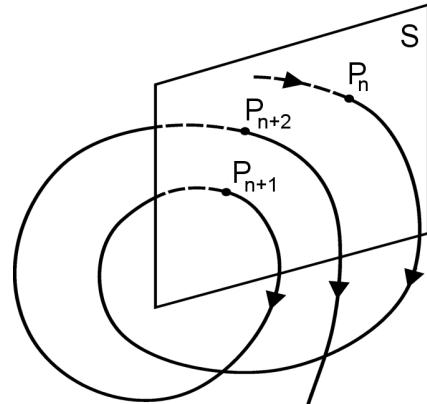


Abbildung 7.19: Poincaré-Schnitt und Poincaré-Abbildung.

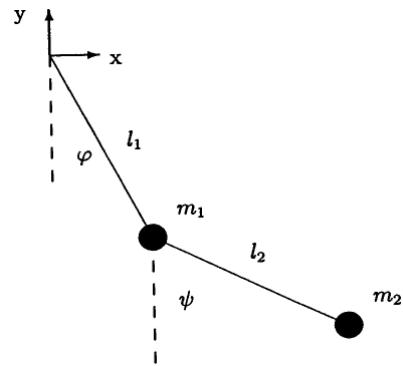
Doppelpendel

Wir betrachten ein **ebenes Doppelpendel** mit zwei Massen m_1 und m_2 und Stablängen l_1 und l_2 . Dieses Beispiel ist aus Korsch/Jodl [4]. Das Doppelpendel wird durch die Lagrangefunktion

$$L = \frac{1}{2} M l_1^2 \dot{\varphi}^2 + \frac{1}{2} m_2 l_2^2 \dot{\psi}^2 + m_2 l_1 l_2 \dot{\varphi} \dot{\psi} \cos(\psi - \varphi) - M g l_1 (1 - \cos \varphi) - m_2 g l_2 (1 - \cos \psi), \quad (7.27)$$

beschrieben, wobei $M = m_1 + m_2$ die Gesamtmasse ist. Die resultierenden Bewegungsgleichungen lauten

$$\begin{aligned} \ddot{\varphi} &= \{1 - \mu \cos^2(\psi - \varphi)\}^{-1} [\mu g_1 \sin \psi \cos(\psi - \varphi) \\ &\quad + \mu \dot{\varphi}^2 \sin(\psi - \varphi) \cos(\psi - \varphi) - g_1 \sin \varphi + \frac{\mu}{\lambda} \dot{\psi}^2 \sin(\psi - \varphi)] \\ \ddot{\psi} &= \{1 - \mu \cos^2(\psi - \varphi)\}^{-1} [g_2 \sin \varphi \cos(\psi - \varphi) \\ &\quad - \mu \dot{\psi}^2 \sin(\psi - \varphi) \cos(\psi - \varphi) - g_2 \sin \psi - \lambda \dot{\varphi}^2 \sin(\psi - \varphi)] \end{aligned} \quad (7.28)$$



mit $\lambda \equiv l_1/l_2$, $g_i \equiv g/l_i$ ($i = 1, 2$) und $\mu \equiv m_2/M$. Die Bewegungsgleichungen sind offensichtlich *nichtlinear*, zum einen auf Grund der intrinsischen Nichtlinearität eines einfachen mathematischen Pendels, zum anderen auf Grund der Kopplung beider Pendel.

Die Bewegungsgleichungen können z.B. mit dem Runge-Kutta-Verfahren 4. Ordnung aus Kapitel 4.3 sehr genau gelöst werden. Ein **Poincaré-Schnitt** kann durch die Bedingung

$$\psi = 0 \quad (\text{und } \dot{\psi} + \lambda \dot{\varphi} \cos \varphi > 0) \quad (7.29)$$

definiert werden, wobei der zweite Teil die Richtung festlegt, in der die Trajektorie die Hyperebene $\psi = 0$ schneidet. Da die numerische Lösung der Bewegungsgleichungen nur zu diskreten Zeiten vorliegt, geht man so vor, dass bei einem Vorzeichenwechsel von ψ (und korrekter Richtung) nochmal (linear) interpoliert wird, um den Poincaré-Schnittpunkt möglichst genau zu bestimmen.

Poincaré-Schnitte zu ansteigender Energie sind in Abb. 7.20 gezeigt: Bei kleinen Energien sind die Auslenkungen der Schwingungen klein und damit auch die Nichtlinearitäten in den Bewegungsgleichungen. Zunächst verhält sich das System daher wie ein integrables System mit invarianten Tori, die sich als geschlossene Konturen mit Kreistopologie darstellen. Mit zunehmender Energie gleitet das System ins Chaos ab und die invarianten Tori lösen sich in flächenfüllende Punkte auf.

Gedämpfter getriebener nichtlinearer Oszillatator

Das zweite Beispiel ist der **gedämpfte getriebene nichtlineare Oszillatator** aus Aufgabe 6 aus Kapitel 4 (siehe auch Fitzpatrick [9]). Die Bewegungsgleichung lautet

$$\ddot{\theta} = -\frac{\dot{\theta}}{Q} - \sin \theta + A \cos(\omega t) \quad (7.30)$$

wobei die lineare Eigenfrequenz $\omega_0 = 1$ beträgt, die Dämpfung ist proportional zu $1/Q$ (wobei Q auch Qualität genannt wird), die Antriebsfrequenz ist ω , die Antriebsamplitude A . Obwohl dieses System nur $f = 1$ Freiheitsgrad hat, kann es chaotisches Verhalten zeigen, da auf Grund von Dämpfung und Antrieb die Energie nicht erhalten ist. Auch diese Bewegungsgleichung kann numerisch z.B. mit dem Runge-Kutta-Verfahren 4. Ordnung aus Kapitel 4.3 sehr genau gelöst werden.

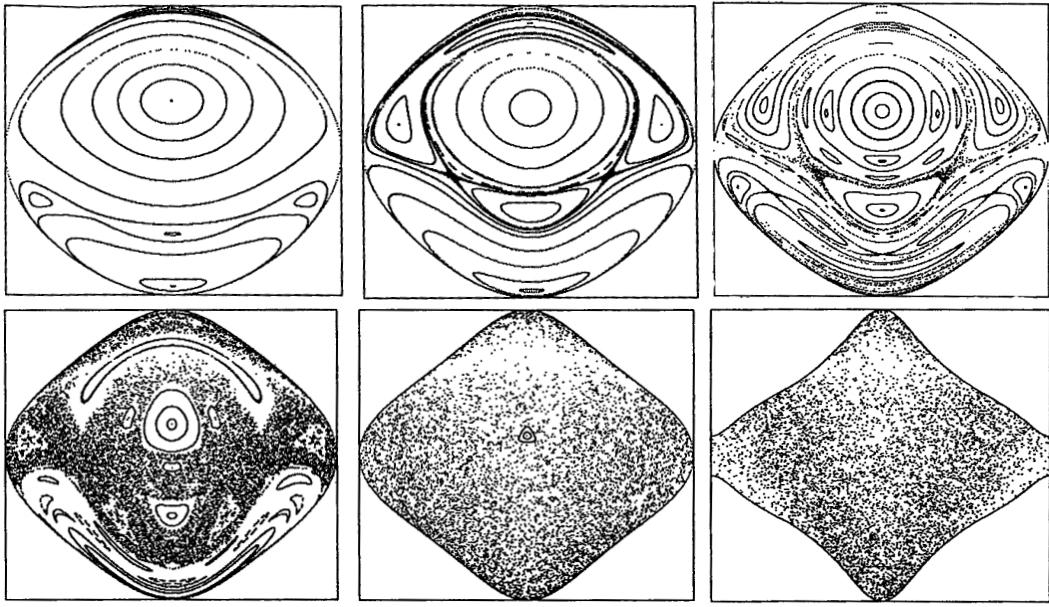


Abbildung 7.20: Poincaré-Schnitte (7.29) bei Gesamtenergien $E = 4, 8, 10, 15, 25, 40$ (von links nach rechts). (Quelle: Korsch/Jodl [4]).

Ein **Poincaré-Schnitt** ist hier einfach durchzuführen, da das System nach einer Einschwingphase die treibende Frequenz ω annimmt. Die Winkelvariable ist dann die Phase ωt und die Poincaré-Schnittfläche kann durch eine konstante Phase definiert werden und wird dann zu *äquidistanten* Zeiten

$$t_n = nT = 2\pi n/\omega \quad (7.31)$$

durchstoßen. Da hier nur $f = 1$ Freiheitsgrad vorliegt, besteht der Poincaré-Schnitt nur aus **einzelnen Punkten**, entweder $\theta(t_n)$ oder $\dot{\theta}(t_n)$.

In der Übung soll $\dot{\theta}(t_n)$ als Funktion von Q betrachtet werden für $A = 3/2$ und $\omega = 2/3$. Das Ergebnis ist in Abb. 7.21 gezeigt: Links in einem relativ kleinen Wertebereich für Q erkennen wir wieder das **universelle Feigenbaum Verzweigungsszenario** für Iterationen. In diesem Fall definiert die Poincaré-Abbildung eine Abbildung im Raum der reellen Zahlen $\dot{\theta}$, die iteriert wird. Wir können sogar verifizieren, dass die Feigenbaumkonstante

$$\delta = \lim_{n \rightarrow \infty} \frac{Q_n - Q_{n-1}}{Q_{n+1} - Q_n} = 4.669\dots$$

wieder ihren universellen Wert (7.22) annimmt. Die rechte Ansicht in Abb. 7.21 zeigt eine gröbere Übersicht über einen größeren Wertebereich für Q , und wir erkennen auch die anderen Eigenschaften nichtlinearer Iterationen wie chaotisches Verhalten und periodische Fenster.

7.5.3 Weg ins Chaos: KAM-Theorem, Poincaré-Birkhoff-Theorem

Nicht-integrable Systeme kann man im Regelfall nur noch numerisch untersuchen. Es gibt allerdings einige äußerst wichtige mathematische Sätze, die das “Abgleiten” eines integrablen Systems in chaotisches Verhalten qualitativ charakterisieren, wenn es nicht-integrabel gestört wird.

Wird ein integrables System mit einer nicht-integrablen Störung versehen, werden auch die invarianten Tori gestört. Das KAM-Theorem (Kolmogorov, Arnold, Moser) beantwortet, unter welchen Bedingungen solch eine Störung zu chaotischem Verhalten führen kann.

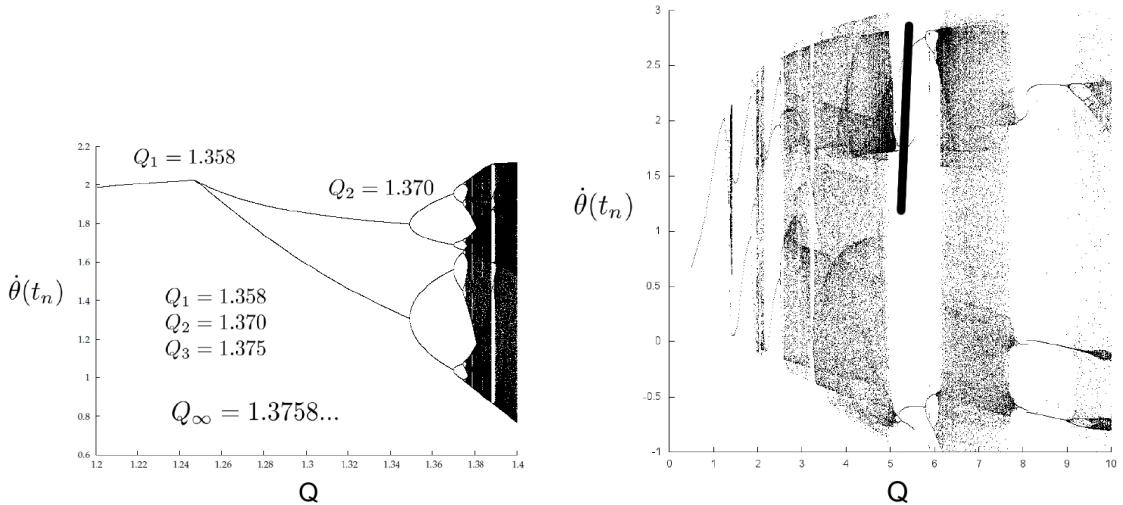


Abbildung 7.21: Poincaré-Schnitte (7.31) bestehend aus einem Punkt $\dot{\theta}(t_n)$ als Funktion von Q für $A = 3/2$ und $\omega = 2/3$. (Quelle: Boltz/Kampmann).

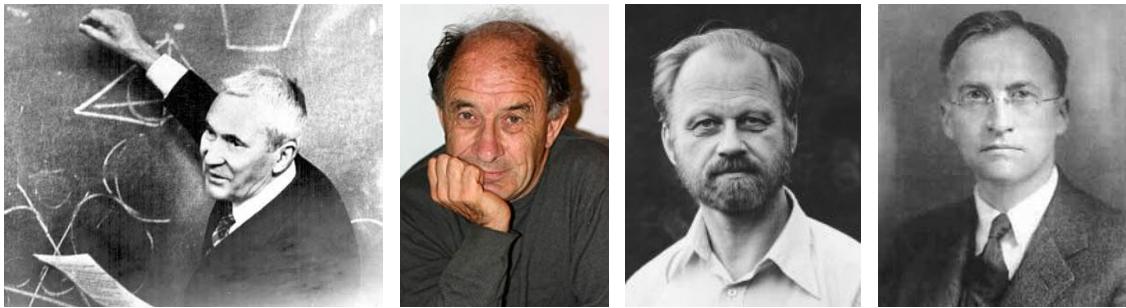


Abbildung 7.22: Von links nach rechts: Andrey Kolmogorov (1903-1987), russischer Mathematiker. Vladimir Igorevich Arnold (1937-2010), russischer Mathematiker, Jürgen Moser (1928-1999), deutsch-amerikanischer Mathematiker. George Birkhoff (1884-1944), amerikanischer Mathematiker. (Quelle: Wikipedia).

Dabei ist es zunächst einmal wichtig, den Begriff die invarianten Tori integrabler Systeme weiter zu klassifizieren in

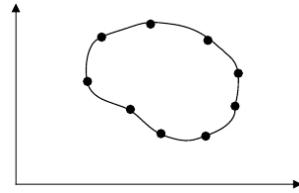
- **Resonante Tori:**

Sind die (konstanten) Winkelgeschwindigkeiten ω_i der Winkelvariablen φ_i **kommensurabel**, d.h.

$$\exists m_1, \dots, m_f \in \mathbb{Z} : m_1\omega_1 + \dots + m_f\omega_f = 0,$$

heißt der invariante Torus resonant. Die Bewegung auf einem resonanten Torus ist periodisch (siehe Abb. 7.23, rechts) und die Poincaré-Schnitte bestehen aus diskreten Punkten auf einem Kreis (für eine feste Anfangsbedingung).

- **Nicht-resonante Tori:**



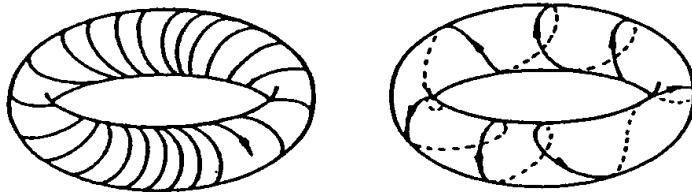
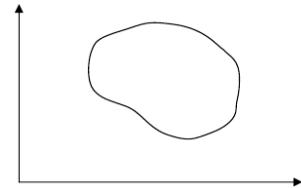


Abbildung 7.23: Dynamik auf einem invarianten Torus. Links: Eine Bewegung auf einem nicht-resonanten Torus füllt den ganzen Torus aus. Rechts: Die Bewegung auf einem resonanten Torus ist periodisch. (Quelle [4]).

Sind die Winkelgeschwindigkeiten ω_i **inkommensurabel**, d.h.

$$m_1\omega_1 + \dots + m_f\omega_f = 0 \iff m_1 = \dots = m_f = 0$$

$(\forall m_1, \dots, m_f \in \mathbb{Z})$, ist der invariante Torus nicht-resonant. Die Bewegung auf einem nicht-resonanten Torus füllt den Torus aus (siehe Abb. 7.23, links) und die Poincaré-Schnitte füllen den Kreis aus (selbst für eine feste Anfangsbedingung).



Bei einer nicht-integrablen Störung kann man zunächst **kanonische Störungstheorie** verwenden. Diese versucht eine kanonische Transformation auf "gestörte" Winkel- und Wirkungsvariablen. Dabei stellt sich heraus, dass eine solche Störungstheorie für *resonante* Tori zusammenbricht. Ein anschauliches Argument dafür ist, dass die ungestörte Bewegung dann periodisch war und sich die Störung dadurch "aufschaukeln" kann.

Das **KAM-Theorem** macht eine präzise Aussage, welche invarianten Tori auch bei einer nicht-integrablen Störung "fast invariant" bleiben:

- Nach Anschalten einer kleinen nicht-integrablen Störung bestehen fast alle invarianten Tori eines integrablen Systems fort (evtl. leicht deformiert).
- Nur in der Nähe *resonanter* (kommensurabler) Tori werden die invarianten Tori *instabil*.
- Das Maß der Menge der instabilen Tori kann beliebig klein gemacht werden für schwache Störungen.

(Für einen Beweis siehe zum Beispiel das Buch von Arnold [8]. Entscheidend ist, dass die fortbestehenden Tori *nicht* Tori zu gleichen Anfangsbedingungen sind, sondern man sucht Tori mit gleicher Frequenz durch Ändern der Anfangsbedingungen.)

Das KAM-Theorem hat wichtige Konsequenzen: Es zeigt beispielsweise, dass der Einfluss eines schweren Planeten wie des Jupiter auf die Erdumlaufbahn nicht zu einer chaotischen Erdtrajektorie führt, solange die Jupiterumlaufzeit gemessen in Jahren ausreichend "irrational" ist.

Während das KAM-Theorem die Aussage macht, dass die nicht-resonanten Tori fast alle "überleben", wenn eine nicht-integrable Störung eingeschaltet wird, macht das **Poincaré-Birkhoff-Theorem** eine Aussage darüber, was mit den resonanten Tori passiert. Es besagt:

Nach Anschalten der Störung verbleiben aus einem resonanten Torus eine gerade Anzahl von Fixpunkten, eine Hälfte ist elliptisch, eine Hälfte hyperbolisch (instabil).

Das Fazit beider Theoreme ist folgendes: Während nicht-resonante Tori nur deformiert werden, zerbricht ein resonanter Torus in eine Folge von elliptischen und hyperbolischen Fixpunkten. In der Nähe der hyperbolischen Fixpunkte wird die Bewegung chaotisch. Ein schematischer Poincaré-Schnitt wie in Abb. 7.24 veranschaulicht dies.

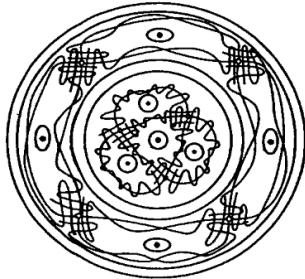


Fig. 2.5. Intact invariant curves and destroyed zones with chains of alternating elliptic (\bullet) and hyperbolic (\circ) fixed points.

Abbildung 7.24: Ein resonanter Torus zerbricht in elliptische und hyperbolische Fixpunkte. (Quelle [4]).

7.6 Literaturverzeichnis Kapitel 7

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [4] H. Korsch und H. Jodl. *Chaos: a program collection for the PC*. Bd. 1. Springer, 1999.
- [5] W. Kinzel und G. Reents. *Physics by Computer*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
- [6] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Studies in nonlinearity. Westview Press, 2008.
- [7] M. J. Feigenbaum. *Quantitative Universality for a Class of Nonlinear Transformations*. J. Stat. Phys. **19** (1978), 25–52.
- [8] V. Arnol'd. *Mathematical Methods of Classical Mechanics*. Graduate Texts in Mathematics. Springer, New York, 1997.
- [9] R. Fitzpatrick. *Computational Physics (Skript)*. Austin, Texas: The University of Texas at Austin, 2012.

7.7 Übungen Kapitel 7

1. Mean-Field Gleichung Ising-Modell

Lösen Sie die Mean-Field Gleichung für die Magnetisierung m im Ising-Modell,

$$m = \tanh [(H + Jzm)/k_B T] = \tanh [(H + k_B T_c m)/k_B T]$$

mit $k_B T_c = zJ$, numerisch mittels Newton-Raphson-Verfahren.

a) Berechnen Sie numerisch für $H/k_B T_c = 0, 0.1, 0.5$ die Magnetisierung $m = m(T)$ als Funktion von T . Plotten Sie dazu die Kurve $m(T)$ im Bereich $0 < T/T_c < 3$.

b) Berechnen Sie für $T/T_c = 0.5, 1, 1.5$ numerisch die Magnetisierung $m = m(H)$ als Funktion von H . Plotten Sie die Kurve $m(H)$ im Bereich $-3 < H/k_B T_c < 3$. Verwenden Sie sowohl $m_0 = +1$ als auch $m_0 = -1$ als Startwerte für das Newton-Verfahren. Was passiert für $T < T_c$?

2. Bifurkationsdiagramme

Berechnen und plotten Sie die Bifurkationsdiagramme für die Abbildungen

- (a) $x_{n+1} = rx_n(1 - x_n)$, $x_n \in [0, 1]$, (logistische Abbildung)
- (b) $x_{n+1} = rx_n - x_n^3$, $x_n \in [-\sqrt{1+r}, \sqrt{1+r}]$, (kubische Abbildung)

durch numerische Iteration. Vergrößern Sie dafür den Parameter r in kleinen Schritten in einem Bereich $0 < r < r_{max}$. Was passiert, wenn Sie r zu groß wählen und welche r_{max} ergeben sich? Iterieren Sie dann für jedes r lange genug, bis sich ein Fixpunkt oder Orbit einstellt. Jeder Punkt des Orbits ergibt einen Punkt im Bifurkationsdiagramm.

3. Feigenbaum-Konstante

Die Fixpunkt-Gleichung für die 2^n -fach iterierte logistische Abbildung $f(r, x) = rx(1 - x)$,

$$x = f^{2^n}(r, x) = f(f(\dots f(r, x))),$$

gibt für $r < r_\infty = 3.5699\dots$ die Werte eines Orbits der Länge 2^n im Periodenverdopplungsszenario nach Feigenbaum.

Bestimmen Sie für $n = 1, 2, 3$ numerisch die Werte $r = R_n < r_\infty$, für die superstabile Fixpunkte existieren. Diese Werte erfüllen die Gleichung

$$\frac{1}{2} = f^{2^n}\left(r, \frac{1}{2}\right).$$

a) Plotten Sie zunächst $g_n(r) \equiv 1/2 - f^{2^n}(r, 1/2)$ als Funktion von r für $n = 0, 1, 2, 3$ im Bereich $0 < r < r_\infty$. Wird n um 1 vergrößert, kommt jeweils eine Nullstelle von $g_n(r)$ bei $r = R_n$ hinzu. Machen Sie die Schrittweite in Ihrem Plot so klein, dass sie Schranken für die Nullstellen angeben können.

b) Bestimmen Sie numerisch mit Hilfe von Intervallhalbierung oder Regula falsi ausgehend von den Schranken aus a) die Nullstellen R_n von $g_n(r)$ für $n = 1, 2, 3$.

c) Gewinnen Sie aus Ihren Ergebnissen eine erste Schätzung der Feigenbaum-Konstante

$$\delta = \lim_{n \rightarrow \infty} \frac{R_{n-1} - R_{n-2}}{R_n - R_{n-1}}$$

4. Poincaré-Schnitte

Werfen Sie nochmal einen Blick auf Aufgabe 6 (speziell Teil b)) aus Kapitel 4!

8 Matrixdiagonalisierung, Eigenwertprobleme

Literatur zu diesem Teil:

Es gibt umfangreiche Literatur zur numerischen Behandlung von Matrix-Eigenwertproblemen, z.B. Numerical Recipes [1, 2] oder das Standardwerk von Golub/van Loan [3].

Das klassische **Eigenwertproblem** sieht folgendermaßen aus: Gegeben ist eine $N \times N$ Matrix $\underline{\underline{A}}$. Gesucht wird ein **Eigenwert (EW)** λ mit einem **Eigenvektor (EV)** \vec{x} , die die **Eigenwertgleichung**

$$\underline{\underline{A}} \cdot \vec{x} = \lambda \vec{x} \quad (8.1)$$

erfüllen. Aus der EW-Gleichung folgt unmittelbar

$$0 = \det(\lambda \underline{\underline{1}} - \underline{\underline{A}}) = P(\lambda) \quad (8.2)$$

mit dem **charakteristischen Polynom** $P(\lambda)$. Ein EW λ ist also ein Nullstelle des charakteristischen Polynoms vom Grad N . Die Gesamtheit der Nullstellen von $P(\lambda)$ ergeben das **Spektrum** $\lambda(\underline{\underline{A}}) = \{\lambda_1, \dots, \lambda_N\}$ der EW.

Eine naive numerische Methode, das Eigenwertproblem zu lösen, wäre daher:

- 1) Bestimme die Nullstellen λ_i des charakteristischen Polynoms $P(\lambda)$
- 2) Löse zu jedem EW λ_i das lineare Gleichungssystem

$$\underline{\underline{A}} \cdot \vec{x}_i = \lambda_i \vec{x}_i,$$

um ein Eigensystem $\{\vec{x}_1, \dots, \vec{x}_N\}$ aus EV \vec{x}_i zu erhalten

Dieses naive Verfahren ist allerdings für die Numerik zu uneffektiv und zu ungenau.

Alle effektiven numerischen Verfahren beruhen auf **Ähnlichkeitstransformationen**

$$\underline{\underline{A}} \longrightarrow \underline{\underline{A}}' = \underline{\underline{Z}}^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{Z}} \quad (8.3)$$

die die EW erhalten wegen

$$\det(\lambda \underline{\underline{1}} - \underline{\underline{Z}}^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{Z}}) = \det \underline{\underline{Z}}^{-1} \det(\lambda \underline{\underline{1}} - \underline{\underline{A}}) \det \underline{\underline{Z}} = \det(\lambda \underline{\underline{1}} - \underline{\underline{A}}).$$

Die Ähnlichkeitstransformation ändert aber die EV:

$$\begin{aligned} & \vec{x}' \text{ EV von } \underline{\underline{A}}' \text{ zu EW } \lambda \\ \iff & \underline{\underline{Z}}^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{Z}} \cdot \vec{x}' = \lambda \vec{x}' \iff \underline{\underline{A}} \cdot (\underline{\underline{Z}} \cdot \vec{x}') = \lambda (\underline{\underline{Z}} \cdot \vec{x}') \\ \iff & \underline{\underline{Z}} \vec{x}' \text{ EV von } \underline{\underline{A}} \text{ zu selben EW } \lambda. \end{aligned}$$

Ist eine Matrix **diagonalisierbar**, heißt das

$$\exists \underline{\underline{Z}} : \underline{\underline{Z}}^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{Z}} = \underline{\underline{D}} \quad (8.4)$$

wobei $\underline{\underline{D}}$ eine Diagonalmatrix mit Elementen $d_{ij} = \lambda_i \delta_{ij}$. Das heißt dann aber auch, dass die Matrix $\underline{\underline{Z}}$ in (8.4) die EV von $\underline{\underline{A}}$ als Spalten enthält:

$$\underline{\underline{A}} \cdot \underline{\underline{Z}} = \underline{\underline{Z}} \cdot \underline{\underline{D}} \iff \sum_j a_{ij} z_{jk} = \sum_j z_{ij} d_{jk} = \lambda_k z_{ik}.$$

Nicht jede Matrix ist diagonalisierbar; das gilt sowohl über dem Körper der reellen Zahlen (dort hat bereits das charakteristische Polynom nicht unbedingt nur reelle Nullstellen) als auch über dem Körper der komplexen Zahlen (dort kann es bei entarteten, also vielfachen EW, Probleme geben, obwohl sich das charakteristische Polynom immer in Linearfaktoren zerlegen lässt). Für die Physik ist der folgende **Spektralsatz** sehr wichtig, der Diagonalisierbarkeit reeller symmetrischer bzw. hermitescher komplexer Matrizen garantiert:

- 1) Jede **symmetrische reelle** Matrix $\underline{\underline{A}} = \underline{\underline{A}}^t$ kann mit **orthogonalem** $\underline{\underline{Z}}$ ($\underline{\underline{Z}}^{-1} = \underline{\underline{Z}}^t$) diagonalisiert werden mit **reellen** EW λ_i .
- 2) Jede **hermitesche** Matrix $\underline{\underline{A}} = (\underline{\underline{A}}^t)^* = \underline{\underline{A}}^+$ kann mit **unitärem** $\underline{\underline{Z}}$ ($\underline{\underline{Z}}^{-1} = \underline{\underline{Z}}^+$) diagonalisiert werden mit **reellen** EW λ_i .
- 3) Die Spalten von $\underline{\underline{Z}}$ bilden eine **orthonormale Eigenbasis**.

Die **Idee** aller üblichen numerischen Verfahren zur Diagonalisierung, die wir im Folgenden diskutieren, ist immer folgende: Wir suchen **sukzessive Ähnlichkeitstransformationen**

$$\underline{\underline{A}} \rightarrow \underline{\underline{P}}_1^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_1 \rightarrow \underline{\underline{P}}_2^{-1} \cdot \underline{\underline{P}}_1^{-1} \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_1 \cdot \underline{\underline{P}}_2 \rightarrow \dots,$$

bis die Diagonalgestalt erreicht ist. Dann können wir die EW auf der Diagonale ablesen und die Spalten des Produktes $\underline{\underline{P}}_1 \cdot \underline{\underline{P}}_2 \cdot \dots$ sind die EV. Bei reellen symmetrischen Matrizen werden wir immer mit *orthogonalen* Transformationsmatrizen $\underline{\underline{P}}_i$ ansetzen nach Spektralsatz.

Es gibt eine Vielzahl von numerischen Verfahren beruhend auf diesem Prinzip, die oft auf spezielle Matrixtypen (tridiagonal, Bandmatrizen, symmetrisch, hermitesch, usw.) zugeschnitten sind.

8.1 Jacobi-Rotation

Der Jacobi-Algorithmus beruhend auf Jacobi-Rotationen als Ähnlichkeitstransformationen für reelle symmetrische Matrizen ist nicht das effektivste, aber ein relativ einfach zu implementierendes Verfahren.

Der Jacobi-Algorithmus diagonalisiert reelle symmetrische Matrizen $\underline{\underline{A}}$. Zur Ähnlichkeitstransformation benutzen wir bei diesem Verfahren in jedem Schritt eine **Jacobi-** oder **Givens-Rotation** $\underline{\underline{P}}_{pq}$ in der pq -Ebene der Matrix:

$$\underline{\underline{P}}_{pq} \equiv \begin{pmatrix} 1 & & & & & & & 0 & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & \cos \phi & \dots & \sin \phi & & & \\ & & & & \ddots & & & & \\ & & & -\sin \phi & \dots & \cos \phi & & & \\ & & & & & & 1 & & \\ 0 & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix}. \quad (8.5)$$

Dabei stehen die 4 Einträge $\cos \phi$, $\sin \phi$, $-\sin \phi$ und $\cos \phi$ in den zwei Spalten p und q und den zwei Zeilen p und q der Matrix $\underline{\underline{P}}_{pq}$. Die so definierte Matrix $\underline{\underline{P}}_{pq}$ ist **orthogonal** (daher der Name Jacobi-Rotation) und eine Ähnlichkeitstransformation kann dann auch als

$$\boxed{\underline{\underline{A}} \longrightarrow \underline{\underline{A}}' = \underline{\underline{P}}_{pq}^t \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_{pq}} \quad (8.6)$$

geschrieben werden.

Die **Idee** ist nun folgende:

Wir versuchen in jedem Transformationsschritt der Art (8.6), das nicht-diagonale Element a_{pq} der Matrix $\underline{\underline{A}}$ nach Null zu transformieren, so dass $a'_{pq} = 0$. Daraus folgt für $c \equiv \cos \phi$, $s \equiv \sin \phi$

$$\begin{aligned} a'_{pq} &= (c^2 - s^2)a_{pq} + sc(a_{pp} - a_{qq}) \stackrel{!}{=} 0 \\ \iff \Theta &\equiv \frac{c^2 - s^2}{2sc} = \frac{a_{qq} - a_{pp}}{2a_{pq}}. \end{aligned} \quad (8.7)$$

Θ wird also aus der ursprünglichen Matrix $\underline{\underline{A}}$ nach (8.7) berechnet. Dann bestimmen sich c und s für die Jacobi-Rotationsmatrix gemäß

$$\begin{aligned} t &= \frac{s}{c} \Rightarrow 2\Theta = \frac{1}{t} - t \\ t &= \frac{\operatorname{sgn}(\Theta)}{|\Theta| + \sqrt{\Theta^2 + 1}} \\ c &= \frac{1}{\sqrt{1+t^2}}, \quad s = tc. \end{aligned} \quad (8.8)$$

Die Beziehungen (8.7) für Θ und (8.8) für t , s und c bestimmen die Jacobi-Rotation $\underline{\underline{P}}_{pq}$, die das Matrixelement a_{pq} von $\underline{\underline{A}}$ nach Null transformiert.

Dieser Schritt muss nun prinzipiell nacheinander auf *alle* nicht-diagonalen Matrixelemente von $\underline{\underline{A}}$ angewandt werden. Dabei sind aufeinanderfolgende Transformationen aber nicht unabhängig und man läuft Gefahr, bei nachfolgenden Transformationen bereits nach Null transformierte nicht-diagonalen Elemente wieder zu verlieren. Dies ist aber beherrschbar, denn es gilt für

$$\operatorname{off}(\underline{\underline{A}}) \equiv \sum_{r \neq s} |a_{rs}|^2 = \sum \text{ nicht-diagonale Quadrate}$$

bei einer Ähnlichkeitstransformation mit einer Jacobi-Rotation mit $\underline{\underline{P}}_{pq}$:

$$\boxed{\operatorname{off}(\underline{\underline{A}}') = \operatorname{off}(\underline{\underline{A}}) - 2|a_{pq}|^2.} \quad (8.9)$$

Damit kann sichergestellt werden, dass die transformierte Matrix $\underline{\underline{A}}'$ auf jeden Fall “diagonaler” ist als die Ausgangsmatrix $\underline{\underline{A}}$.

Der **iterative Jacobi-Algorithmus** reiht nun Ähnlichkeitstransformationen mit Jacobi-Rotationen aneinander, bis $\operatorname{off}(\underline{\underline{A}})$ unter einem Genauigkeitsziel ε liegt:

- 1) Wähle Zielschranke ε für $\text{off}(\underline{\underline{A}})$.
- 2) Wähle pq so, dass $|a_{pq}|$ maximal, $|a_{pq}| = \max_{i \neq j} |a_{ij}|$.
- 3) Berechne Θ, t, c, s nach (8.7) und (8.8).
- 4) Ähnlichkeitstransformation $\underline{\underline{A}}' = \underline{\underline{P}}_{pq}^t \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_{pq}$.
- 5) Wenn $\text{off}(\underline{\underline{A}}') < \varepsilon$, dann Ende; sonst $\underline{\underline{A}}' = \underline{\underline{A}}$ und wieder 2).

Der Jacobi-Algorithmus hat folgende Eigenschaften:

- Er funktioniert immer, allerdings relativ langsam für große N .
- Für große N ist die Max-suche in 2) langsam. Daher arbeitet man dann die nicht-diagonalen Elemente besser in einer festen systematischen Reihenfolge ab, z.B.

$$pq = 12, 13, \dots, 1N; 23, 24, \dots, 2N; 34, \dots$$

Nach einem vollen “sweep” beginnt man wieder von vorne.

- Typischerweise sind dann 6-10 sweeps notwendig, d.h. $3N^2$ bis $5N^2$ Ähnlichkeitstransformationen mit Jacobi-Rotationen, jede davon braucht $\sim 8N$ floating-point Operationen:
 $a'_{pq} = 0, a'_{pp} = a_{pp} - ta_{pq}, a'_{qq} = a_{qq} + ta_{pq}$ (4 Operationen) mit $t = s/c$.
Dann (N-2) Berechnungen $a'_{rp} = a_{rp} - s(a_{rq} + \tau a_{rp})$ (mit jeweils 4 floating-point Operationen) und nochmal
(N-2) Berechnungen $a'_{rq} = a_{rq} + s(a_{rp} - \tau a_{rq})$ (mit jeweils 4 floating-point Operationen) mit $\tau \equiv s/(1+c)$.
Damit sind insgesamt $24N^3$ bis $40N^3$ Operationen nötig, um die EW zu bestimmen.
- Für die EV muss zusätzlich noch $\underline{\underline{Z}} = \underline{\underline{P}}_1 \cdot \underline{\underline{P}}_2 \cdot \dots$ berechnet werden, d.h. bei jeder Jacobi-Rotation muss $\underline{\underline{Z}}' = \underline{\underline{Z}} \cdot \underline{\underline{P}}_{pq}$ mit berechnet werden.

8.2 Householder und QR-Iteration

Mit Hilfe des Householder-Algorithmus kann eine symmetrische Matrix auf Tridiagonalform transformiert werden. Die QR-Iteration ist ein schnelles Verfahren, um eine Tridiagonalmatrix dann auf Diagonalform zu transformieren.

Für große N ist der Jacobi-Algorithmus langsam. Eine bessere Strategie für große N ist folgende:

- 8.2.1: Zunächst eine Transformation auf **Tridiagonalform** (d.h. eine Matrix mit drei diagonalen Bändern mit Elementen ungleich 0). Für diesen Schritt benutzt man den Householder-Algorithmus.
- 8.2.2: Dann die EW und EV für die Tridiagonalmatrix berechnen. Für diesen Schritt ist die QR-Iteration ein effektives Verfahren.

8.2.1 Householder-Algorithmus

Auch der **Householder-Algorithmus zur Tridiagonalisierung** ist auf reelle und symmetrische Matrizen $\underline{\underline{A}}$ anwendbar. Im Gegensatz zur Jacobi-Iteration ist dies ein deterministischer Algorithmus der nach einer festen Zahl von Schritten beendet ist (unabhängig von Genauigkeitszielen).



Abbildung 8.1: Links: James Wallace Givens (1910-1993), amerikanischer Mathematiker. Mitte: Carl Jacobi (1804-1851), deutscher Mathematiker. Rechts: Alston Scott Householder (1904-1993), amerikanischer Mathematiker.

1) Der Algorithmus beginnt mit der 1.Zeile und 1.Spalte von $\underline{\underline{A}}$ und sucht eine orthogonale Transformationsmatrix $\underline{\underline{P}}_1$ mit der Wirkung

$$\underline{\underline{P}}_1^t \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_1 = \left(\begin{array}{c|ccc} a_{11} & k_1 & 0 & \\ \hline k_1 & a'_{22} & a'_{23} & \dots & a'_{2N} \\ & a'_{32} & & & a'_{3N} \\ 0 & \vdots & & & \vdots \\ & a'_{N2} & \dots & & a'_{NN} \end{array} \right). \quad (8.10)$$

2) Danach suchen wir ein orthogonales $\underline{\underline{P}}_2$ mit analoger Wirkung auf die Untermatrix $\underline{\underline{A}}'$, d.h. auf die 2.Zeile und 2.Spalte:

$$(\underline{\underline{P}}_1 \cdot \underline{\underline{P}}_2)^t \cdot \underline{\underline{A}} \cdot (\underline{\underline{P}}_1 \cdot \underline{\underline{P}}_2) = \left(\begin{array}{cc|cc} a_{11} & k_1 & 0 & \\ k_1 & a'_{22} & k_2 & \\ \hline & k_2 & a''_{33} & \dots \\ 0 & & \vdots & \end{array} \right) \quad (8.11)$$

usw., bis nach $N - 2$ Durchläufen die Matrix Tridiagonalgestalt hat.

Um solche orthogonalen Matrizen $\underline{\underline{P}}_n$ zu finden, machen wir den Ansatz

$$\boxed{\underline{\underline{P}}_n = \left(\begin{array}{c|c} \underline{\underline{1}}_{n \times n} & 0 \\ \hline 0 & \underline{\underline{S}}_n \end{array} \right)}, \quad (8.12)$$

wobei $\underline{\underline{S}}_n$ eine symmetrische $(N - n) \times (N - n)$ Matrix mit $\underline{\underline{S}}_n^t = \underline{\underline{S}}_n$ ist ($\iff \underline{\underline{P}}_n^t = \underline{\underline{P}}_n$). Die Orthogonalitätseigenschaft $\underline{\underline{P}}_n^t \cdot \underline{\underline{P}}_n = \underline{\underline{1}}$ ist dann äquivalent zu $\underline{\underline{S}}_n^2 = \underline{\underline{1}}$ und wird erfüllt von Matrizen der Form

$$\boxed{\underline{\underline{S}}_n = \underline{\underline{1}} - 2\vec{u}_n \otimes \vec{u}_n \quad \text{mit } \vec{u}_n^2 = 1.} \quad (8.13)$$

Die durch (8.12) und (8.13) definierte Klasse von **orthogonalen** Matrizen heißen **Householder-Matrizen**. Die Matrizen $\underline{\underline{S}}_n$ aus (8.13) haben auch eine geometrische Interpretation: Sie beschreiben eine Spiegelung im \mathbb{R}^{N-n} an einer $N - n - 1$ -dimensionalen Hyperebene mit Normalenvektor \vec{u}_n .

Es bleibt die Frage, wie der Vektor \vec{u}_n zu wählen ist, damit die Householder-Matrizen zur gewünschten Tridiagonalisierung führen. Dazu betrachten wir den Fall $n = 1$:

$$\underline{\underline{P}}_1^t \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_1 = \left(\begin{array}{c|c} a_{11} & (\underline{\underline{S}}_1 \cdot \vec{v})^t \\ \hline \underline{\underline{S}}_1 \cdot \vec{v} & \underline{\underline{A}}' \end{array} \right) \quad \text{mit} \quad \vec{v} \equiv \begin{pmatrix} a_{21} \\ \vdots \\ a_{N1} \end{pmatrix}.$$

Damit (8.10) gilt, muss

$$\underline{\underline{S}}_1 \cdot \vec{v} = \vec{v} - 2\vec{u}_1(\vec{u}_1 \cdot \vec{v}) \stackrel{!}{=} k_1 \underbrace{\vec{e}_1}_{(N-1)-\text{dim.}}.$$

Wir suchen also im \mathbb{R}^{N-1} die durch den Normalenvektor \vec{u}_1 beschriebene Hyperebene, so dass der Vektor \vec{v} nach Spiegelung parallel zum kartesischen Einheitsvektor \vec{e}_1 ist.

$$\Rightarrow k_1^2 = (\underline{\underline{S}}_1 \cdot \vec{v})(\underline{\underline{S}}_1 \cdot \vec{v}) \stackrel{\underline{\underline{S}}_1^2 = 1}{=} \vec{v}^2 = \sum_{i=2}^N a_{i1}^2.$$

Dies bestimmt k_1 bis auf das Vorzeichen:

$$k_1 = \pm |\vec{v}|. \quad (8.14)$$

Wegen $\vec{u}_1 \parallel (\vec{v} - k_1 \vec{e}_1)$ und $\vec{u}_1^2 = 1$ ist dann auch \vec{u}_1 bestimmt:

$$\vec{u}_1 = \frac{\vec{v} - k_1 \vec{e}_1}{|\vec{v} - k_1 \vec{e}_1|} \quad (8.15)$$

Wir wählen das Vorzeichen von $-k_1$ in (8.14) wie das Vorzeichen von a_{21} , um Rundungsfehler in (8.15) und in der Transformation (8.10) zu minimieren.

Der **Householder-Algorithmus** läuft nun folgendermaßen ab:

- Wir konstruieren $\underline{\underline{P}}_1$ und k_1 aus $\underline{\underline{A}}$ gemäß (8.14) und (8.15).
- Ähnlichkeitstransformation (8.10): $\underline{\underline{A}}' = \underline{\underline{P}}_1^t \cdot \underline{\underline{A}} \cdot \underline{\underline{P}}_1$.
- Analog weiter: Konstruiere $\underline{\underline{P}}_2$ und k_2 aus $\underline{\underline{A}}'$
- usw., bis nach $N - 2$ Iterationen Tridiagonalgestalt erreicht ist.

Der Householder-Algorithmus braucht $\mathcal{O}(N^3)$ Operationen wie der Jacobi-Algorithmus, allerdings ist im Normalfall der Vorfaktor von N^3 hier wesentlich kleiner.

8.2.2 Eigenwerte und Eigenvektoren tridiagonaler Matrizen

Charakteristisches Polynom

Eine Möglichkeit, die EW und EV einer tridiagonalen Matrix $\underline{\underline{A}}$ zu bestimmen, ist direkt über ihr **charakteristisches Polynom** $P(\lambda)$. Dieses kann für tridiagonale Matrizen rekursiv schnell berechnet werden, was eine effektive Nullstellensuche ermöglicht, die dann die EW liefert. Die zugehörigen EV werden dann über die Lösung des linearen Gleichungssystems $\underline{\underline{A}} \cdot \vec{x} = \lambda_i \vec{x}$ zu jedem EW λ_i bestimmt.

QR-Zerlegung

Eine noch effektivere Methode gründet auf der **QR-Zerlegung**: *Jede reelle Matrix $\underline{\underline{A}}$ kann folgendermaßen zerlegt werden:*

$$\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}} \quad (8.16)$$

mit $\underline{\underline{Q}}$ orthogonal und $\underline{\underline{R}}$ obere (rechte) Dreiecksmatrix

Z.B. können die **Householder-Matrizen** $\underline{\underline{P}}_i$ aus 8.2.1 auch zur QR-Zerlegung genutzt werden:
Wir wollen zeigen, dass wir ein $\underline{\underline{Q}}^t$ der Form $\underline{\underline{Q}}^t = \underline{\underline{P}}_{N-1} \cdot \dots \cdot \underline{\underline{P}}_0$ so konstruieren können, dass $\underline{\underline{Q}}^t \cdot \underline{\underline{A}} = \underline{\underline{R}}$.

Dafür betrachten wir zuerst die Wirkung einer Householder-Matrix $\underline{\underline{P}}_0$ der Form (8.12). $\underline{\underline{P}}_0$ soll so gewählt werden, dass

$$\underline{\underline{P}}_0 \cdot \underline{\underline{A}} = \underline{\underline{S}}_0 \cdot \underline{\underline{A}} = \begin{pmatrix} k_0 & * \\ 0 & \underline{\underline{A}}' \end{pmatrix}.$$

Dafür muss

$$\underline{\underline{S}}_0 \cdot \vec{v} = k_0 \vec{e}_1 \quad \text{mit } \vec{v} = \text{1.Spalte von } \underline{\underline{A}} \text{ und}$$

$$\underline{\underline{S}}_0 = \underline{\underline{1}} - 2\vec{u}_0 \otimes \vec{u}_0.$$

Dies kann wiederum erfüllt werden mit der Wahl

$$k_0 = \pm |\vec{v}| \text{ und } \vec{u}_0 = \frac{\vec{v} - k_0 \vec{e}_1}{|\vec{v} - k_0 \vec{e}_1|}.$$

Damit ist $\underline{\underline{P}}_0$ aus $\underline{\underline{A}}$ konstruiert.

Danach wird ein $\underline{\underline{P}}_1$ der Form (8.12) analog aus $\underline{\underline{A}}'$ konstruiert usw. so dass am Ende

$$\underline{\underline{Q}}^t \cdot \underline{\underline{A}} = \underline{\underline{P}}_{N-1} \cdot \dots \cdot \underline{\underline{P}}_0 \cdot \underline{\underline{A}} = \underline{\underline{R}}$$

eine obere Dreiecksmatrix ist. Für eine *beliebige* reelle Matrix $\underline{\underline{A}}$ braucht diese QR-Zerlegung $\mathcal{O}(N^3)$ Operationen.

Für eine **tridiagonale und symmetrische** Matrix

$$\underline{\underline{A}} = \begin{pmatrix} a_{11} & k_1 & & & & 0 \\ k_1 & a_{22} & k_2 & & & \\ & k_2 & a_{33} & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ 0 & & \ddots & \ddots & k_{N-1} & \\ & & & k_{N-1} & a_{NN} & \end{pmatrix}$$

wie sie vom Householder-Algorithmus erzeugt wird, lässt sich $\underline{\underline{Q}}^t$ einfacher als ein Produkt von **Jacobi-Rotationen** (8.5) schreiben:

$$\underline{\underline{Q}}^t = \underline{\underline{P}}_{N-1N} \cdot \dots \cdot \underline{\underline{P}}_{23} \cdot \underline{\underline{P}}_{12}$$

die jeweils ein unteres k_i "wegdrehen". Zunächst konstruieren wir $\underline{\underline{P}}_{12}$:

$$\begin{aligned} \underline{\underline{Q}}^t \cdot \underline{\underline{A}} &\stackrel{!}{=} \underline{\underline{R}} = \begin{pmatrix} r_{11} & & * & \\ & r_{22} & & \\ & & \ddots & \\ 0 & & & r_{NN} \end{pmatrix} \\ \underline{\underline{P}}_{12} \begin{pmatrix} a_{11} \\ k_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} &\stackrel{!}{=} \begin{pmatrix} r_{11} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \iff \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} a_{11} \\ k_1 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} r_{11} \\ 0 \end{pmatrix} \\ \Rightarrow t &= \frac{s}{c} = \frac{k_1}{a_{11}} \\ c &= \frac{1}{\sqrt{t^2 + 1}}, \quad s = tc. \end{aligned}$$

Die letzte Zeile bestimmt $\underline{\underline{P}}_{12}$.

Analog geht es dann weiter, für $\underline{\underline{P}}_{n,n+1}$ gilt:

$$\boxed{\begin{aligned} t &= \frac{s}{c} = \frac{k_n}{a_{nn}} \\ c &= \frac{1}{\sqrt{t^2 + 1}}, \quad s = tc. \end{aligned}} \tag{8.18}$$

Diese auf Jacobi-Rotationen beruhende QR-Zerlegung braucht nur $\mathcal{O}(N)$ Operationen.

QR-Iteration

Mit Hilfe der QR-Zerlegung lässt sich die "**QR-Transformation**" einer Tridiagonalmatrix durchführen:
Mit $\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}}$ bzw. $\underline{\underline{R}} = \underline{\underline{Q}}^t \cdot \underline{\underline{A}}$ definieren wir

$$\boxed{\underline{\underline{R}} \cdot \underline{\underline{Q}} = \underline{\underline{Q}}^t \cdot \underline{\underline{A}} \cdot \underline{\underline{Q}} = \underline{\underline{A}}' = \text{QR-Trafo von } \underline{\underline{A}}} \tag{8.19}$$

Da $\underline{\underline{Q}}$ orthogonal ist, ist auch die QR-Transformation eine Ähnlichkeitstransformation, die die EW erhält. Weiter gilt: Wenn die Matrix $\underline{\underline{A}}$ tridiagonal ist wie nach dem Householder-Algorithmus, dann ist auch die QR-Transformierte $\underline{\underline{A}}'$ tridiagonal.

Mit Hilfe der QR-Transformation wird die **QR-Iteration** für ein tridiagonales $\underline{\underline{A}}$ definiert:

- Wir starten mit $\underline{\underline{T}}_0 = \underline{\underline{A}}$.
- Mit Hilfe der QR-Zerlegung $\underline{\underline{T}}_{k-1} = \underline{\underline{Q}}_k \cdot \underline{\underline{R}}_k$ von $\underline{\underline{T}}_{k-1}$ definieren wir
- $\underline{\underline{T}}_k \equiv \underline{\underline{R}}_k \cdot \underline{\underline{Q}}_k = \underline{\underline{Q}}_k^t \cdot \underline{\underline{T}}_{k-1} \cdot \underline{\underline{Q}}_k$ als QR-Transformierte (8.19) von $\underline{\underline{T}}_{k-1}$.

Zu der so definierten QR-Iteration (die die EW in jedem Schritt erhält) gibt es folgenden mathematischen

Satz: Die Folge $\underline{\underline{T}}_k$ konvergiert fast immer gegen die gesuchte Diagonalmatrix.

Der Beweis ist nicht-trivial, siehe [3]. ‘‘Fast’’ heißt hier, dass entartete EW Probleme bereiten. Dies sieht man an dem Ergebnis (siehe [3]), dass nicht-diagonale Elemente $t_{i+1,i}^{(k)}$ der Matrizen $\underline{\underline{T}}_k$ nach Null konvergieren wie $(|\lambda_{i+1}|/|\lambda_i|)^k$, wobei λ_i die gesuchten EW sind, nach Größe sortiert: $|\lambda_1| > |\lambda_2| > \dots > |\lambda_N|$.

Typisch sind $\mathcal{O}(N)$ Iteration nötig, pro Iteration braucht man $\mathcal{O}(N)$ Operationen. Daher braucht die Diagonalisierung einer Tridiagonalmatrix nur $\mathcal{O}(N^2)$ Operationen.

8.3 Potenzmethode, Transfermatrix

Die Potenzmethode ist eine einfache und wichtige Methode, um hochdimensionale Matrizen zu diagonalisieren. Eine ‘‘physikalische Realisierung’’ stellt die Transfermatrixmethode dar. Eine praktisch sehr wichtige Anwendung ist das Google PageRank-Verfahren.

8.3.1 Potenzmethode

Der betragsmäßig *größte* EW λ_1 und der zugehörige EV \vec{x}_1 einer diagonalisierbaren Matrix $\underline{\underline{A}}$ können einfach durch wiederholtes Anwenden von $\underline{\underline{A}}$ gefunden werden.

Dies ist die **Potenzmethode**:

- 1) Wähle einen Startvektor \vec{v}_0 .
- 2) Definiere eine Iteration durch $\vec{w}_n = \underline{\underline{A}} \cdot \vec{v}_{n-1}$ und (optionaler) Normierung $\vec{v}_n = \frac{\vec{w}_n}{\|\vec{w}_n\|}$.

Dazu gibt es folgenden

Satz: Die Folge \vec{v}_n konvergiert gegen den normierten EV \vec{x}_1 und $\langle \vec{v}_n | \underline{\underline{A}} \vec{v}_n \rangle$ gegen EW λ_1 .

Beweis:

$\underline{\underline{A}}$ ist diagonalisierbar mit normierte EV-Basis $\{\vec{x}_1, \dots, \vec{x}_2\}$ und EW mit $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_N|$. Wir nehmen hier zunächst an, dass der betragsmäßig größte Eigenwert nicht entartet ist. Dann kann $\vec{v}_0 = \sum_{i=1}^N \alpha_i \vec{x}_i$ in der EV-Basis dargestellt werden und

$$\begin{aligned} \underline{\underline{A}}^n \vec{v}_0 &= \sum_{i=1}^N \alpha_i \lambda_i^n \vec{x}_i \\ &= \lambda_1^n \left(\alpha_1 \vec{x}_1 + \sum_{i=2}^N \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^n \vec{x}_i \right). \end{aligned}$$

Es gilt aber

$$\begin{aligned} \lim_{n \rightarrow \infty} \left| \frac{\lambda_i}{\lambda_1} \right|^n &= 0 \\ \Rightarrow \frac{\underline{\underline{A}}^n \vec{v}_0}{\|\underline{\underline{A}}^n \vec{v}_0\|} &= \vec{x}_1 + \mathcal{O} \left(\left| \frac{\lambda_2}{\lambda_1} \right| \right). \end{aligned}$$

Das beweist den Satz und aus der letzten Fehlerabschätzung folgt zudem, dass die Konvergenz nach \vec{x}_1 vom Grade $p = 1$ ist.

Wir schließen mit einigen Bemerkungen zur Potenzmethode:

- Die Potenzmethode ist relativ langsam; insbesondere, wenn $|\lambda_2/\lambda_1|$ nur wenig kleiner als 1 ist.
- Bei Entartung des betragsmäßig größten Eigenwerts, $\lambda_1 = \lambda_2$, konvergiert $\langle \vec{v}_n | \underline{\underline{A}} \vec{v}_n \rangle$ ebenfalls gegen λ_1 , der Vektor \vec{v}_n konvergiert gegen die normierte Linearkombination $\frac{\alpha_1 \vec{x}_1 + \alpha_2 \vec{x}_2}{|\alpha_1 \vec{x}_1 + \alpha_2 \vec{x}_2|}$ aus dem Eigenraum.
- Sie ist die einzige praktikable Methode für *sehr* große N ($N > 10^6$ oder Ähnliches).
- Hat man den betragsmäßig größten EW λ_1 gefunden, kann der nächste EW λ_2 gefunden werden, indem man das Verfahren anschließend auf

$$\underline{\underline{A}} - \lambda_1 \vec{x}_1 \otimes \vec{x}_1$$

anwendet usw. Wegen $(\underline{\underline{A}} - \lambda_1 \vec{x}_1 \otimes \vec{x}_1) \vec{x}_1 = 0$ hat diese neue $N \times N$ Matrix statt λ_1 einen neuen EW $\lambda = 0$ im Spektrum; der betragsmäßig größte EW ist nun λ_2 .

- Das **Lanczos-Verfahren zur Tridiagonalisierung** beruht auch auf der Potenzmethode, ist aber noch mächtiger, da es auch die “unterwegs” generierte Information $\underline{\underline{A}} \vec{v}_0, \underline{\underline{A}}^2 \vec{v}_0, \dots$ zur Tridiagonalisierung in nur $\mathcal{O}(N)$ Operationen verwendet, siehe z.B. das Buch [3] von Golub/v.Loa. Ausgehend von der Tridiagonalform können dann auch Eigenwerte bestimmt werden. **Modifizierte Lanczos-Verfahren** erlauben auch die direkte Bestimmung des größten oder kleinsten Eigenwertes, was wichtig bei der Bestimmung von Grundzuständen in der Quantenmechanik ist.

8.3.2 Transfermatrix des 1D Ising-Modells

Die Potenzmethode ist ein wichtiges numerisches *und* analytisches Werkzeug in der statistischen Physik in Form der **Transfermatrixmethode**, wo sie dann auch auf kontinuierliche Systeme mit Operatoren statt Matrizen oder diskrete Systeme im thermodynamischen Limes mit “unendlich”-dimensionalen Matrizen angewendet wird.

Die **Idee** der Transfermatrixmethode ist, die Zustandssumme eines Systems als Spur über eine Potenz einer **Transfermatrix** $\underline{\underline{T}}$ zu schreiben

$$Z = \text{Sp} \underline{\underline{T}}^N,$$

wobei $N \sim$ Systemgröße. Im thermodynamischen Limes $N \rightarrow \infty$ wird die Zustandssumme dann eine “physikalische Realisierung” der Potenzmethode.

Wir erläutern die Transfermatrixmethode am Beispiel des 1-dimensionalen Ising-Modells. Das Ising Modell wurde bereits in Kapitel (7.3.2) eingeführt. Das 1D-Ising Modell besteht aus N gekoppelten Spins $s_i = \pm 1$ auf einer eindimensionalen Reihe von Gitterplätzen.

Wir betrachten ferromagnetische Kopplungen zwischen nächsten Nachbarn $J_{ij} = J > 0$ für $|i-j| = 1$ und $J_{ij} = 0$ sonst in der Hamiltonfunktion aus Gl. (7.13). Die Hamiltonfunktion des **1D-Ising Modells** wird dann

$$\mathcal{H}_{\text{Ising}} = -J \sum_{i=1}^N s_i s_{i+1} - H \sum_{i=1}^N s_i. \quad (8.20)$$

Wir wollen hier **periodische Randbedingungen** betrachten, d.h. wir identifizieren $s_{N+1} \equiv s_1$. Dies entspricht dann einem Ising-Ring, siehe Abb. 8.2. Wir schreiben die negative Hamiltonfunktion (8.20) in Temperatureinheiten $-\beta \mathcal{H}$ mit Hilfe der Abkürzungen $\bar{J} \equiv \beta J$ und $\bar{H} = \beta H$ in etwas

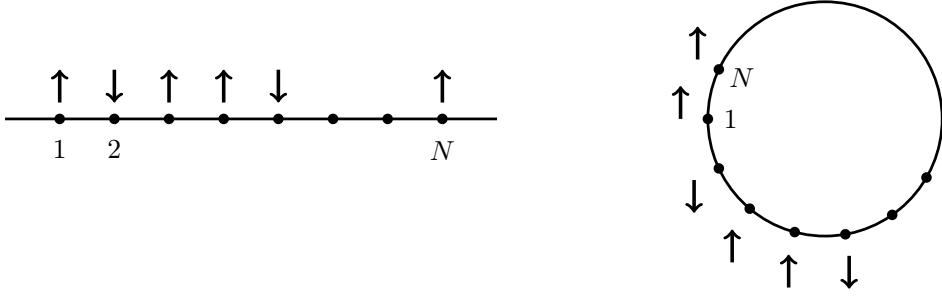


Abbildung 8.2: Links: 1D-Ising Modell, Rechts: 1D Ising-Ring bei periodischen Randbedingungen.

symmetrisierter Form als

$$-\beta \mathcal{H} = \sum_i \underbrace{\left(\bar{J} s_i s_{i+1} + \frac{1}{2} \bar{H}(s_i + s_{i+1}) \right)}_{\equiv K(s_i, s_{i+1})}.$$

Die Größe $K(s_i, s_{i+1})$ ist symmetrisch bezgl. der Vertauschung $s_i \leftrightarrow s_{i+1}$ und kann als 2x2 Matrix mit den Indizes $ij = s_i s_{i+1}$ interpretiert werden, da die $s_i = \pm 1$ jeweils 2 Werte annehmen können. Daraus gewinnen wir dann die **Transfermatrix** als die Matrix der entsprechenden Boltzmann-Gewichte

$$\underline{\underline{T}} = T_{s_i s_{i+1}} = e^{K(s_i, s_{i+1})} = \begin{pmatrix} e^{\bar{J} + \bar{H}} & e^{-\bar{J}} \\ e^{-\bar{J}} & e^{\bar{J} - \bar{H}} \end{pmatrix} \begin{array}{l} \leftarrow s_i = +1 \\ \uparrow \\ s_{i+1} = +1 \end{array} \quad \begin{array}{l} \leftarrow s_i = -1 \\ \uparrow \\ s_{i+1} = -1. \end{array} \quad (8.21)$$

Die Transfermatrix ist hier also eine 2x2 Matrix, weil 2 die Dimension des Zustandsraumes $s_i = \pm 1$ eines Spins ist. Mit Hilfe der Transfermatrix lässt sich die Zustandssumme (für periodische Randbedingungen) schreiben als:

$$\begin{aligned} Z &= \sum_{\{s_i\}} e^{-\beta \mathcal{H}} \\ &= \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{K(s_1, s_2)} e^{K(s_2, s_3)} \dots e^{K(s_{N-1}, s_N)} e^{K(s_N, s_1)} \\ &= \text{Sp} \left(\underline{\underline{T}} \cdot \underline{\underline{T}} \cdot \dots \cdot \underline{\underline{T}} \right). \end{aligned}$$

Also

$$Z = \text{Sp} \underline{\underline{T}}^N. \quad (8.22)$$

Die Transfermatrix kann normalerweise (wie auch in unserem Beispiel 1D Ising-Modell) so gewählt werden, dass sie **symmetrisch** ist und sich damit diagonalisieren lässt. Außerdem ist sie im Normalfall auf Grund ihrer Definition über Boltzmann-Gewichte auch positiv, d.h. alle EW sind $\lambda_i > 0$ (dies gilt auch im 1D Ising-Modell), was wir im Folgenden immer annehmen wollen. Dann lässt sich $\underline{\underline{T}}$ diagonalisieren

$$\underline{\underline{T}} = \underline{\underline{Z}} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \underline{\underline{Z}}^{-1} \quad \text{mit } \lambda_1 > \lambda_2 \quad \text{EW von } \underline{\underline{T}}$$

und die Zustandssumme als

$$\begin{aligned}
Z &= \text{Sp} \left(\underline{\underline{Z}} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \underline{\underline{Z}}^{-1} \right) \left(\underline{\underline{Z}} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \underline{\underline{Z}}^{-1} \right) \dots \\
&= \text{Sp} \left(\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}^N \right) = \lambda_1^N + \lambda_2^N \\
&= \lambda_1^N \left(1 + \left(\frac{\lambda_2}{\lambda_1} \right)^N \right) \\
&\stackrel{N \rightarrow \infty}{\approx} \lambda_1^N
\end{aligned} \tag{8.23}$$

umschreiben. Damit dominiert genau wie bei der Potenzmethode der größte EW den thermodynamischen Limes $N \rightarrow \infty$. Für die freie Energie ergibt sich damit

$$\begin{aligned}
F(T, H) &= -k_B T \ln Z(T, H) \\
&= -k_B T N \ln \lambda_1 - k_B T \ln \underbrace{\left(1 + \left(\frac{\lambda_2}{\lambda_1} \right)^N \right)}_{\approx 0} \\
&\stackrel{N \rightarrow \infty}{\approx} -k_b T N \ln \lambda_1.
\end{aligned} \tag{8.24}$$

Also ist auch die freie Energie durch den größten EW λ_1 bestimmt.

Im 1D Ising-Modell gilt mit $\underline{\underline{T}}$ aus (8.21) für die EW der Transfermatrix:

$$\boxed{\lambda_{\pm} = e^{\bar{J}} \cosh \bar{H} \pm \left(e^{2\bar{J}} \sinh^2 \bar{H} + e^{-2\bar{J}} \right)^{1/2}} \tag{8.25}$$

Bei $\bar{H} = 0$ ergibt das $\lambda_{\pm} = e^{\bar{J}} \pm e^{-\bar{J}}$. Also gilt immer $\lambda_+ > \lambda_-$ und damit $\lambda_1 = \lambda_+$.

Transfermatrix und Phasenübergänge

Ein **Phasenübergang** liegt vor, wenn die freie Energie $F(T, H)$ (also das zu den gegebenen intensiven Variablen T und H gehörige **thermodynamische Potential**) **nicht-analytisch** ist an einer Übergangstemperatur $T = T_c$. Die freie Energie ist aber als thermodynamisches Potential auf jeden Fall stetig an T_c . Bei einem **diskontinuierlichen Phasenübergang** (Phasenübergang 1. Ordnung) ist die 1. Ableitung der freien Energie unstetig an T_c und damit hat die freie Energie einen "Kinken", siehe Abb. 8.3 links. Bei einem **kontinuierlichen Phasenübergang** (Phasenübergang höherer Ordnung) ist die Nicht-Analytizität derart, dass erst eine höhere Ableitung der freien Energie eine Divergenz an T_c aufweist. Die freie Energie selbst sieht dann recht glatt aus am Übergang, siehe Abb. 8.3 rechts.

Wegen (8.24) gilt in der Transfermatrixmethode, dann: ein Phasenübergang liegt genau dann vor, wenn der größte EW $\lambda_1(T)$ als Funktion der Temperatur T nicht-analytisch ist. Dazu bemerken wir folgendes:

- Die EW $\lambda_i(T)$ selbst sind analytische Funktionen, wie man für das 1D Ising-Modell an (8.25) sieht.
- Allerdings ist der *größte* EW eine nicht-analytische Funktion, wenn die Funktionen $\lambda_1(T)$ und $\lambda_2(T)$ sich **schnüren**, d.h. wenn der größte EW als Funktion von T "wechselt" siehe Abb. 8.4. Am Phasenübergang muss dann $\lambda_1(T_c) = \lambda_2(T_c)$ gelten, die größten EW also **entartet** sein.

Zur Entartung des größten EW macht das **Perron-Frobenius-Theorem** eine wichtige Aussage:

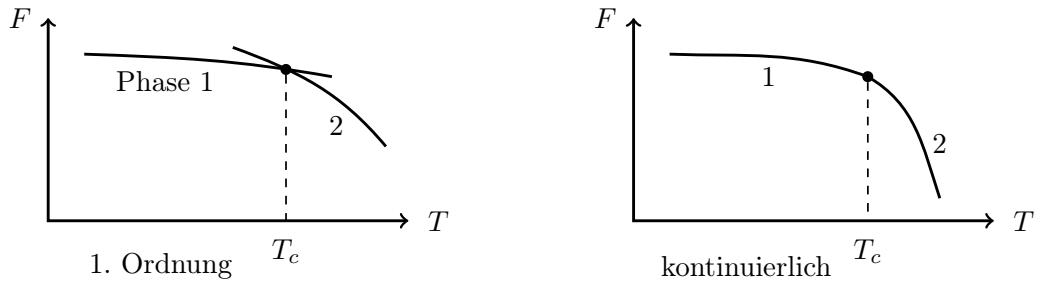


Abbildung 8.3: Links: Diskontinuierlicher Phasenübergang (1. Ordnung). Rechts: Kontinuierlicher Phasenübergang (höherer Ordnung).

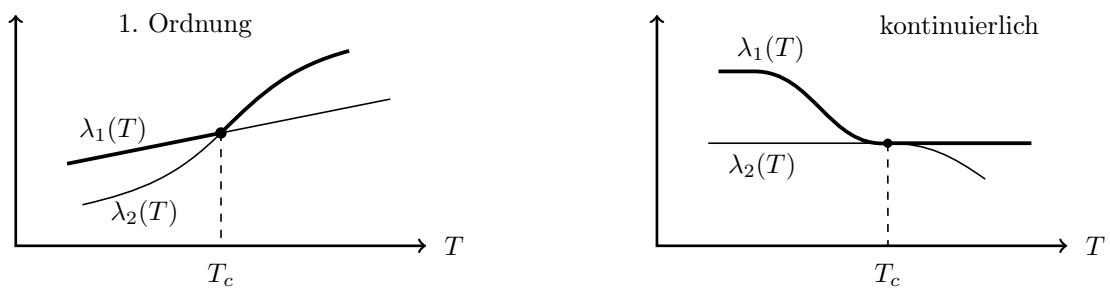


Abbildung 8.4: Links: Verhalten der EW $\lambda_i(T)$ an einem diskontinuierlichen Phasenübergang. Rechts: an einem kontinuierlichen Phasenübergang.

Für eine **endlich-dimensionale** Matrix $\underline{\underline{A}}$, bei der alle $a_{ij} > 0$, gilt

- a) \exists reeller EW $\lambda_1 > 0$: $\lambda_1 > \lambda$ für alle anderen EW λ .
- b) Dieser größte EW λ_1 ist **nicht entartet**.

Auf Grund der Aussage b) kann es *keinen* Phasenübergang in Systemen mit einer endlich-dimensionalen Transfermatrix $\underline{\underline{T}}$ geben. Insbesondere folgt

Es gibt *keinen* Phasenübergang im 1D Ising-Modell.

da wir dort ja nur eine 2×2 Transfermatrix haben. Allgemeiner gilt sogar: Es gibt keine Phasenübergänge in klassischen 1D Systemen mit kurzreichweiter Wechselwirkung.

Wir wissen allerdings, dass das 2D Ising-Modell sehr wohl einen Phasenübergang besitzt. Es war eines der ersten nicht-trivialen Modelle, wo analytisch gerade mittels der Transfermatrixmethode von Lars Onsager 1944 gezeigt werden konnte, dass für $D = 2$ und $H = 0$ ein kontinuierlicher Phasenübergang vorliegt. Dies steht nicht im Widerspruch zum Perron-Frobenius-Theorem, da die Transfermatrix $\underline{\underline{T}}$ des 2D Ising-Modells ∞ -dimensional ist. Im 2D Ising-Modell gibt es $N \times N$ Spins und die Transfermatrix nach Onsager ist eine Transfermatrix zwischen ganzen **Reihen** von jeweils N Spins, die daher jeweils 2^N Zustände haben. Damit wird dann aber auch die Transfermatrix 2^N -dimensional und $2^N \rightarrow \infty$ im thermodynamischen Limes $N \rightarrow \infty$. Daher ist hier ein Phasenübergang möglich, wie ihn die Onsager-Lösung ergibt.

Im 1D Ising-Modell ist gibt es dagegen lediglich einen "Phasenübergang" bei $H = 0$ im Limes

$T \rightarrow 0$, also bei “ $T_c = 0$ ”; es gibt also keine echte Tieftemperaturphase. Betrachten wir die EW (8.25), so gilt bei $H = 0$ nämlich

$$\lambda_1 - \lambda_2 = 2e^{-\bar{J}} T \xrightarrow{T \rightarrow 0} 0,$$

da im Limes $T \rightarrow 0$ auch $\bar{J} = J/k_B T \rightarrow \infty$. Damit haben wir lediglich im Limes $T \rightarrow 0$ eine Entartung vorliegen.

8.3.3 Google PageRank

Eine interessante Anwendung der Potenzmethode ist das **PageRank-Verfahren**, auf dem die Internet-Suchmaschine Google basiert. Das PageRank-Verfahren hat sich sein Namensgeber Larry Page, einer der Gründer von Google (neben Sergey Brin) 1998 patentieren lassen [4]. Der Name “Google” geht auf die Bezeichnung “googol” für die Zahl 10^{100} zurück. Der Name scheint passend für die Informationsfülle des WorldWideWeb: 1998 wurden 26×10^6 Seiten gezählt, im Jahr 2000 waren es ca. 10^9 und im Juli 2008 wurde die Marke von 10^{12} Seiten durchbrochen. Dies ist tatsächlich auch die Dimension des Eigenwertproblems, mit dem man es bei der Google-Matrix zu tun hat. Im PageRank Algorithmus wird jeder Seite j im Internet eine “**Wichtigkeit**” r_j zugeordnet. Der Vektor \vec{r} hat daher die Dimension der Anzahl der Seiten im Internet. Mit Hilfe des Vektors \vec{r} werden die Suchergebnisse nach ihrer Wichtigkeit vorsortiert. Die Idee dabei ist, dass Seiten, die von wichtigen Seiten aus verlinkt sind, auch selber wichtig sind.

Das Internet wird im PageRank Algorithmus als **gerichteter Graph** aus **Knoten** und **Kanten** interpretiert: Knoten sind Seiten im WWW; Kanten sind die Links im WWW, die von einer Seite auf eine andere verweisen und daher gerichtet. Zu jeder Seite j gibt es einen “In-degree” $I(j)$, der gleich der Anzahl eingehender Links ist und einen “Out-degree” $O(j)$, der gleich der Zahl ausgehender Links ist.

Aus dem Out-degree wird eine Gleichung für den “Wichtigkeits-Vektor” \vec{r} gewonnen basierend auf folgenden 2 Annahmen:

- 1) Jede Seite j verteilt ihre Wichtigkeit r_j gleichmäßig auf alle $O(j)$ ausgehenden Links.
- 2) Eine Seite ist so wichtig, wie die Summe aller Wichtigkeiten, die über eingehende Links hereinkommen.

Diese beiden Annahmen führen auf folgende Gleichung für \vec{r} :

$$r_i = c \underbrace{\sum_{j \text{ zeigt auf } i} \frac{r_j}{O(j)}}_{\text{Annahme 2}} \underbrace{.}_{\text{Annahme 1}} \quad (8.26)$$

c ist dabei eine Normierung, damit die Gesamtwichtigkeit einen festen Wert hat.

Die Gleichung (8.26) kann auch als **Eigenwertproblem** aufgefasst werden. Dazu definieren wir zuerst eine **Konnektivitätsmatrix**

$$C_{ij} \equiv \begin{cases} 1 & j \text{ zeigt auf } i \\ 0 & \text{sonst} \end{cases} \quad (8.27)$$

Damit lässt sich der Out-degree als $O(j) = \sum_i C_{ij}$ schreiben. Außerdem können wir damit Gleichung (8.26) umschreiben:

$$r_i = c \sum_j C_{ij} \frac{r_j}{O(j)} = c \sum_j S_{ij} r_j,$$

also

$\underline{\underline{S}} \cdot \vec{r} = \frac{1}{c} \vec{r} \quad \text{mit} \quad S_{ij} \equiv \frac{C_{ij}}{O(j)} = \frac{C_{ij}}{\sum_k C_{kj}}$

(8.28)

Das heißt, der Wichtigkeitsvektor \vec{r} ist ein Eigenvektor der “**Google-Matrix**” $\underline{\underline{S}}$ (zum Eigenwert $1/c$). Die Dimension dieser Google-Matrix ist die Dimension des WWW.

PageRank basiert also auf einem Eigenwertproblem, das so hochdimensional ist, dass es effektiv nur noch mit der **Potenzmethode** gelöst werden kann, d.h. die Iteration

$$\vec{r}_{n+1} = \underline{\underline{S}} \cdot \vec{r}_n \quad (8.29)$$

wird bis zur Konvergenz wiederholt. Die Potenzmethode sollte dann gegen den EV zum *größten* EW $1/c$ konvergieren. Bei der Beantwortung der Konvergenzfrage hilft hier eine Erweiterung des **Perron-Frobenius-Theorems**:

Wenn alle $S_{ij} > 0$, dann ist der größte EW λ_1 reell, positiv und nicht entartet.
Für den zugehörigen EV \vec{r} gilt $r_i > 0$ (für alle anderen EV gilt das *nicht*).
Es gilt weiter:
Wenn $\underline{\underline{S}}$ eine **stochastische Matrix**, dann ist $\lambda_1 = 1$.

Eine stochastische Matrix $\underline{\underline{S}}$ ist definiert durch

$$\sum_i S_{ij} = 1 \quad \text{und} \quad S_{ij} > 0. \quad (8.30)$$

Wenn also die Google-Matrix S_{ij} stochastisch ist, gilt:

- Die Potenzmethode konvergiert, und zwar gegen einen EV \vec{r} mit EW $1/c = 1$.
- Für den EV \vec{r} gilt $r_i > 0$. Er kann so normiert werden, dass $\sum_i r_i = 1$. Dies erlaubt folgende Interpretation: r_i ist die Wahrscheinlichkeit, bei *zufälligem* Surfen auf der Seite i zu landen.
- Man kann zeigen: Der EV zum *größten* EW $1/c = 1$ entspricht der *stationären* Wahrscheinlichkeitsverteilung r_i für die Seiten i , die sich ergibt, wenn man sehr lange und völlig zufällig surft. D.h. man folgt von einer Seite i aus zufällig und mit gleicher Wahrscheinlichkeit einem der ausgehenden Links (dann gelangt man genau mit Wahrscheinlichkeit S_{ij} zur Seite j).

Die Matrix $\underline{\underline{S}}$, so wie sie in (8.28) eingeführt wurde, erfüllt $\sum_i S_{ij} = (\sum_i C_{ij})/O(j) = 1$ nach Definition des Out-degree $O(j) = \sum_i C_{ij}$. Allerdings ist die Matrix $\underline{\underline{S}}$ trotzdem noch nicht stochastisch, weil nicht $S_{ij} > 0$ für alle Links ij gilt: Diese Bedingung heißt, dass jede Seite mit jeder anderen Seite verlinkt ist, was offensichtlich nicht der Fall ist. Dies gefährdet also die Konvergenz der Potenzmethode. Als besonders schlimm stellen sich dabei a) Schleifen und b) lose Enden (“dangling ends”) in der Konnektivität heraus, siehe Abb. 8.5.

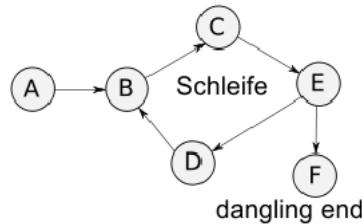


Abbildung 8.5: Links: Larry Page. Rechts: Schleifen und “dangling ends” in der WWW-Konnektivität.

Um dieses Problem zu beseitigen, wird noch eine Modifikation an der Google-Matrix vorgenommen:

$$S_{ij} \equiv p \frac{C_{ij}}{O(j)} + (1-p) \quad (8.31)$$

mit einem Parameter $p < 1$. Die so definierte Matrix S_{ij} ist nun wirklich **stochastisch**, womit die Konvergenz der Potenzmethode in einen EV \vec{r} mit $r_i > 0$ nach Perron-Frobenius-Theorem gesichert ist. Wenn der Startvektor mit $\sum_i r_i^{(1)} = 1$ normiert wird, konvergiert die Potenzmethode gegen einen Wahrscheinlichkeitsvektor \vec{r} mit ebenfalls $\sum_i r_i = 1$ und $r_i > 0$.

Die Bedeutung des **Parameters** p ist folgende: Mit einer Wahrscheinlichkeit p folgt ein Surfer zufällig einem der ausgehenden Links von einer Seite j , mit einer Wahrscheinlichkeit $(1-p)$ aber springt er zu *irgendeiner* Seite i (einschließlich von j selbst). Google benutzt einen Wert von $p = 0.85$.

Zur Matrix S_{ij} wollen wir auch bemerken, dass sie Surfen als **Markov-Prozess** im Raum der WWW-Seiten mit Übergangswahrscheinlichkeiten $M_{ij} = S_{ij}$ beschreibt, siehe später in Kapitel 11.2. Der Wichtigkeitsvektor \vec{r} zum *größten* EW ist die stationäre Verteilung dieses Markov-Prozesses bei sehr langem Surfen.

8.4 Matrixdiagonalisierung in der Quantenmechanik

Matrixdiagonalisierung spielt in der Quantenmechanik zur Lösung stationärer Schrödingergleichungen eine wichtige Rolle. Dazu muss man sich auf einen eindlich-dimensionalen Hilbertraum einschränken.

Eine überaus wichtige Anwendung der Matrixdiagonalisierung ist die **stationäre Schrödingergleichung** der Quantenmechanik:

$$\hat{H}|\psi\rangle = E|\psi\rangle, \quad |\psi\rangle \in \mathcal{H} \text{ Hilbertraum.}$$

Bei der Suche nach Energieeigenzuständen müssen wir

$$\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle \quad (8.32)$$

lösen, wobei $E_0 < E_1 < E_2 < \dots$ das Energiespektrum ergeben mit dem Grundzustand E_0 .

Das Problem (8.32) kann näherungsweise numerisch durch Matrixdiagonalisierung gelöst werden: Dazu betrachten wir den Hamiltonoperator \hat{H} nicht auf seinem i.Allg. ∞ -dimensionalen vollen Hilbertraum \mathcal{H} (in der Ortsdarstellung wäre $\mathcal{H} =$ der Raum der quadratintegrablen Funktionen), sondern auf einem **endlich-dimensionalem Unterraum** $\mathcal{U} \subset \mathcal{H}$, der natürlich geeignet und dem physikalischen Problem angepasst gewählt werden sollte.

Sei $\{|\varphi_1\rangle, \dots, |\varphi_N\rangle\}$ eine Orthonormalbasis des N -dimensionalen Unterraums \mathcal{U} . Dann ist die Einschränkung oder Projektion \hat{H}_U von \hat{H} auf $\mathcal{U} \subset \mathcal{H}$ in der Darstellung durch die $\{|\varphi_i\rangle\}$ durch die **hermitesche $N \times N$ Matrix**

$$H_{ij} = \langle \varphi_i | \hat{H}_U | \varphi_j \rangle = \langle \varphi_i | \hat{H} | \varphi_j \rangle \quad (8.33)$$

gegeben. Diese Matrix H_{ij} kann dann numerisch diagonalisiert werden und wir erhalten N reelle EW $\varepsilon_0 < \varepsilon_1 < \dots < \varepsilon_{N-1}$ mit zugehörigen EV $u_j^{(i)}$, also $|u_i\rangle = \sum_{j=1}^N u_j^{(i)} |\varphi_j\rangle$ mit $\hat{H}_U |u_i\rangle = \varepsilon_i |u_i\rangle$.

Für die numerisch erhaltenen EW ε_i gilt dann das **Hylleraas-Undheim-Theorem** [5]:

- a) $\varepsilon_i \geq E_i$ für alle $i = 0, \dots, N-1$,
d.h. die ε_i sind **obere Schranken**.
- b) Wenn $\mathcal{U} \subset \mathcal{U}' \subset \mathcal{H}$, gilt die **Schachtelung** $\varepsilon_i \geq \varepsilon'_i \geq \varepsilon_{i-1}$,
d.h. für größere Unterräume \mathcal{U} nehmen die ε_i sukzessiv ab.

Dieses Theorem spezifiziert einige Aussagen, die auch intuitiv klar sind: Je größer der Unterraum \mathcal{U} , desto besser approximieren die numerisch gefundenen EW ε_i die echten EW E_i . Den Beweis des Theorems (über Ritzsches Variationsprinzip) sparen wir uns an dieser Stelle.

8.5 Literaturverzeichnis Kapitel 8

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] G. H. Golub und C. F. Van Loan. *Matrix Computations*. 3rd. Johns Hopkins Studies in the Mathematical Sciences. Baltimore, Maryland, USA: Johns Hopkins University Press, 1996.
- [4] L. Page. *Method for node ranking in a linked database*. US Patent 6,285,999. Sep. 2001.
- [5] E. A. Hylleraas und B. Undheim. *Numerische Berechnung der 2 S-Terme von Ortho- und Par-Helium*. Z. Physik **65** (1930), 759–772.

8.6 Übungen Kapitel 8

1. Matrixdiagonalisierung

Gegeben sei die symmetrische 4x4 Matrix

$$\underline{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 2 & 1 \\ 3 & 2 & 6 & 3 \\ 4 & 1 & 3 & 4 \end{pmatrix} \quad (8.34)$$

- a) Bestimmen Sie die Eigenwerte der symmetrischen Matrix \underline{A} mit Hilfe des **Linear Algebra PACKAGE** (“Lapack”: <http://www.netlib.org/lapack/>). Informieren Sie sich über die entsprechenden driver-Routinen, z.B. dsyev in Lapack (<http://www.netlib.org/lapack/lug/node30.html> und <http://www.netlib.org/lapack/lug/node32.html#tabdriveseig>) und wie Sie diese in Ihr Programm einbinden.

Wahlweise können Sie auch Routinen aus den Numerical Recipes oder andere lineare Algebra Pakete verwenden (In den Numerical Recipes wird allerdings auch zum Gebrauch von Lapack geraten). Das Paket **Eigen** stellt beispielsweise eine gute Alternative dar.

- b) Bestimmen Sie die Eigenwerte mit der einfach selbst zu implementierenden Potenzmethode und vergleichen Sie mit dem Resultat aus a).

(Kontrollergebnis: betragsmäßig größter Eigenwert $\lambda_1 \simeq 11.7978$)

2. Anharmonischer Oszillator

In der Quantenmechanik kann die numerische Lösung der stationären Schrödinger-Gleichung auf die Diagonalisierung einer endlich-dimensionalen Matrix zurückgeführt werden, indem man das quantenmechanische System in einem geeigneten endlich-dimensionalen Unterraum seines Hilbertraumes betrachtet.

Hier sollen Sie auf diese Weise die Energieeigenwerte des anharmonischen Oszillators mit dem Hamiltonoperator

$$\hat{H} = -\frac{\hbar^2}{2m}\partial_x^2 + \frac{1}{2}m\omega^2x^2 + \lambda x^4 \quad (8.35)$$

numerisch berechnen.

- a) In welchen Einheiten müssen Sie Längen ($x = \alpha\xi$) und Energien ($E = \beta\epsilon$) messen, um die stationäre Schrödinger-Gleichung in die dimensionslose Form

$$\left[-\partial_\xi^2 + \xi^2 + \tilde{\lambda}\xi^4 \right] \psi(\xi) = \epsilon\psi(\xi) \quad (8.36)$$

zu bringen. Wie lautet die resultierende Beziehung zwischen dem dimensionslosen Störungsparameter $\tilde{\lambda}$ und dem ursprünglichen λ ?

Man kann nun eine Matrixdarstellung in einem endlich-dimensionalen Unterraum auf verschiedene Arten, d.h. in verschiedenen Darstellungen gewinnen. Wir betrachten in b) die Ortsdarstellung und in c) die Besetzungszahldarstellung (bezüglich des ungestörten Problems):

- b) Um aus der Ortsdarstellung eine endlich-dimensionale Darstellung zu bekommen, müssen wir (i) den Ortsraum diskretisieren mit $\xi_n = n\Delta\xi$ und (ii) die Koordinate ξ nur auf einem endlichen Intervall $\xi \in [-L, L]$ betrachten. Diese beiden Schritte wurden auch in Aufgabe 1 in Kapitel 6 durchgeführt. Der Hamiltonoperator nimmt dann die Matrixform

$$\begin{aligned} H_{nm} &= \langle \xi_n | \hat{H} | \xi_m \rangle \\ &= -\frac{1}{(\Delta\xi)^2} (\delta_{n,m-1} + \delta_{n,m+1} - 2\delta_{nm}) + [(\Delta\xi)^2 n^2 + \tilde{\lambda}(\Delta\xi)^4 n^4] \delta_{nm} \end{aligned} \quad (8.37)$$

an, wobei die Indizes n, m nun endlich viele Werte $n, m = -L/\Delta\xi, \dots, L/\Delta\xi$ annehmen können.

Bestimmen Sie die 10 niedrigsten Energieniveaus für $\tilde{\lambda} = 0.2$ näherungsweise numerisch, indem Sie die Matrix (8.37) diagonalisieren auf einem durch $L = 10$ gegebenen Intervall und mit $\Delta\xi = 0.1$ wie in Aufgabe 1 in Kapitel 6. Sie sollen dazu wieder die entsprechenden Routinen aus dem Lapack-Paket einbinden. Betrachten Sie zuerst zur Kontrolle den Fall $\lambda = 0$, der das bekannte Resultat $\epsilon_n = 2(n + 1/2)$ liefern sollte.

c) Nun wollen wir für den gleichen Hamiltonoperator eine endlich-dimensionale Darstellung aus der Besetzungszahldarstellung gewinnen. Dazu verwenden wir die Besetzungszahl-Eigenzustände $|n\rangle$ des ungestörten Oszillators ($\lambda = 0$). Wir berechnen dazu zunächst die Matrixelemente

$$H_{nm} = \langle n | \hat{H} | m \rangle \quad (8.38)$$

Wir verwenden Erzeuger $\hat{a}^+ = (\xi - \partial_\xi)/\sqrt{2}$ und Vernichter $\hat{a} = (\xi + \partial_\xi)/\sqrt{2}$ und schreiben die Anharmonizität mittels $\xi = (\hat{a} + \hat{a}^+)/\sqrt{2}$ um. Nach längerer Rechnung (wer möchte, kann selbst nachrechnen...) erhält man folgendes Ergebnis für den ξ^4 -Term:

$$\begin{aligned} \langle n | \xi^4 | m \rangle &= \frac{1}{4} \left([m(m-1)(m-2)(m-3)]^{1/2} \delta_{n,m-4} \right. \\ &\quad + [(m+1)(m+2)(m+3)(m+4)]^{1/2} \delta_{n,m+4} \\ &\quad + [m(m-1)]^{1/2} (4m-2) \delta_{n,m-2} + [(m+1)(m+2)]^{1/2} (4m+6) \delta_{n,m+2} \\ &\quad \left. + (6m^2 + 6m + 3) \delta_{n,m} \right) \end{aligned} \quad (8.39)$$

Prüfen Sie zuerst nach, dass die durch (8.39) gegebene Matrix wirklich hermitesch bzw. symmetrisch ist. Wie lauten dann die Matrixelemente H_{nm} für den gesamten Hamiltonoperator in dieser Darstellung?

Man berechnet nun die Energieniveaus für $\tilde{\lambda} = 0.2$ näherungsweise numerisch, indem man H_{nm} in einem endlich-dimensionalen Unterraum $n = 0, 1, \dots, N$ diagonalisiert. Berechnen Sie auf diese Art die 10 niedrigsten Energieniveaus für $N = 50$ mit Hilfe der Lapack-Routinen.

d) Verkleinern Sie N und verifizieren Sie das Hylleraas-Undheim-Theorem.

e) Vergleichen Sie die Ergebnisse aus b) und c). Welche Ergebnisse halten Sie für genauer und warum? (Tipp: Dazu bei b) die Diskretisierung verändern bzw. bei c) N verändern und beobachten, wie sich die Ergebnisse ändern.)

3. Chemische Bindung

In einem einfachen Modell für die chemische Bindung betrachten wir ein Teilchen in einem “Doppelmuldenpotential”

$$\begin{aligned} \hat{H} &= -\frac{\hbar^2}{2m} \partial_x^2 + V(x) \\ &= -\frac{\hbar^2}{2m} \partial_x^2 + \lambda [(x^2 - x_0^2)^2 - x_0^4] \end{aligned} \quad (8.40)$$

Damit untersuchen wir die “chemische Bindung” zwischen zwei “Oszillator-Atomen”.

a) Skizzieren Sie $V(x)$. Welche Bedeutung haben x_0 und λ ? Skalieren Sie Längen ($x = \alpha\xi$) und Energien ($E = \beta\epsilon$) so um, dass die stationäre Schrödinger-Gleichung die Form

$$[-\partial_\xi^2 + \tilde{\lambda}(\xi^4 - 2\xi^2)] \psi(\xi) = \epsilon \psi(\xi) \quad (8.41)$$

nimmt. Wie lautet hier die resultierende Beziehung zwischen $\tilde{\lambda}$ und dem ursprünglichen λ ?

b) Diskretisieren Sie das Problem wieder in der Ortsdarstellung wie in Aufgabe 2, Teil b). Berechnen Sie die 10 niedrigsten Energieniveaus für ein großes $\tilde{\lambda} = 100$ und ein kleines $\tilde{\lambda} = 1$.

c) Warum sind die beiden niedrigsten Eigenwerte ϵ_0 und ϵ_1 für großes $\tilde{\lambda} = 100$ (fast) entartet und warum wird die Entartung für kleine $\tilde{\lambda}$ aufgehoben? Was hat dies mit “chemischer Bindung” zwischen den Oszillator-Zentren zu tun?

Gegen welchen Wert sollte die Grundzustandsenergie ϵ_0 für große $\tilde{\lambda}$ gehen? Vergleichen Sie Ihre Vorhersage mit numerischen Resultaten für große $\tilde{\lambda}$.

Berechnen Sie die Aufspaltung $\epsilon_1 - \epsilon_0$ numerisch für verschiedene $\tilde{\lambda}$ und fertigen Sie einen entsprechenden Plot an. Warum wird die Aufspaltung exponentiell klein für große $\tilde{\lambda}$?

Welche Symmetrie bezüglich Raumsymmetrie erwarten Sie für den Grundzustand und den ersten angeregten Zustand? Versuchen Sie dies an Hand der zugehörigen numerisch bestimmten Eigenvektoren zu verifizieren.

4. Transfermatrix des 1D Ising-Modell

Wir betrachten das 1D Ising-Modell mit der Transfermatrixmethode. In der Vorlesung wurde folgender Ausdruck für die Zustandssumme des 1D Ising Modells mit N Gitterplätzen und periodischen Randbedingungen hergeleitet

$$Z = \text{Sp} \underline{\underline{T}}^N \quad (8.42)$$

wobei

$$\underline{\underline{T}} = \begin{pmatrix} e^{\bar{J} + \bar{H}} & e^{-\bar{J}} \\ e^{-\bar{J}} & e^{\bar{J} - \bar{H}} \end{pmatrix}$$

($\bar{J} = J/k_B T$, $\bar{H} = H/k_B T$).

a) Berechnen Sie numerisch die Freie Energie $F = -k_B T \ln Z$ als Funktion von T für $N = 100$ Spins und $J = 1/2$ und bei einem Feld $H = 0.1$

b) Zeigen Sie, dass man die Magnetisierung $m = \langle s_i \rangle$ als

$$m = \frac{1}{Z} \text{Sp} (\underline{\underline{S}} \cdot \underline{\underline{T}}^N) \quad (8.43)$$

mit einer Matrix $S(s_i, s_{i+1}) \equiv s_i \delta_{s_i, s_{i+1}}$

$$\underline{\underline{S}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

berechnen kann. Berechnen Sie damit numerisch die Magnetisierung als Funktion von T für $N = 100$, $J = 1/2$ und $H = 0.1$. Sie können das Resultat mit dem der Mean-Field Rechnung aus Aufgabe 1 in Kapitel 7 vergleichen.

c) Auch Spin-Korrelationen $\langle s_i s_j \rangle$ lassen sich mit der Transfermatrix-Methode berechnen. Zeigen Sie, dass

$$\langle s_i s_j \rangle = \frac{1}{Z} \text{Sp} (\underline{\underline{S}} \cdot \underline{\underline{T}}^{j-i} \cdot \underline{\underline{S}} \cdot \underline{\underline{T}}^{N-(j-i)}) \quad (8.44)$$

(für $j > i$) gilt. Berechnen Sie dann numerisch die Spin-Korrelationen als Funktion von $j - i = 0, \dots, N$ für $N = 100$, $J = 1/2$, $H = 0$ bei Temperaturen $k_B T = 0.01$ und $k_B T = 1$. Zeigen Sie, dass die Spin-Korrelationen exponentiell abfallen.

9 Minimierung

Literatur zu diesem Teil:

Z.B. Numerical Recipes [1, 2].

Die numerische Aufgabe besteht darin, **Minima** $f_{\min} = f(\vec{x}_{\min})$ einer reellen Funktion $f(\vec{x})$ zu suchen. Äquivalent ist natürlich das Problem Maxima zu suchen (von $-f(\vec{x})$).

Die **Anwendungen** in der Physik sind zahlreich:

Least-square Fits

Wenn für Datenpunkte (x_n, y_n) $n = 1, \dots, N$ ein Fit mit einer Funktion $y(x) = f(x, \vec{\alpha})$ mit **Fitparametern** $\vec{\alpha}$ vorgenommen wird, minimiert man, und zwar im Normalfall Fehlerquadrate. Bei diesem Normalfall ist die zu Grunde liegende Annahme, dass die Messwerte y_n gaußverteilt um $y(x)$ mit einer Varianz σ (unabhängig von n) liegen. Diese Annahme fußt im zentralen Grenzwertsatz, der hier besagt, dass bei vielen unabhängigen Fehlerquellen, die sich addieren, im Gesamtergebnis ein gaußverteilter Fehler zustande kommen sollte. Die “beste” Wahl des Parametersatzes $\vec{\alpha}$ (die das “Modell” $f(x, \vec{\alpha})$ beschreiben, für die die vorliegenden Datenpunkte am wahrscheinlichsten sind) erhält man durch **Minimierung** des **quadratischen Fehlers**

$$\chi^2 = \sum_{n=1}^N \frac{(y_n - f(x_n, \vec{\alpha}))^2}{\sigma^2} \quad (9.1)$$

bezgl. $\vec{\alpha}$. Dies ist ein sogenannter “**Least-square-Fit**”.

Klassische Energieminima

Klassische Grundzustände minimieren die Gesamtenergie. Energieminimierung ist aber keine triviale Aufgabe: Bei Systemen mit **vielen Freiheitsgraden** ist die Energieminimierung in einem entsprechend hochdimensionalen Raum aufwendig. Außerdem wird die Energieminimierung prinzipiell problematisch, wenn viele **metastabile Minima** existieren.

Dazu 2 Beispiele:

- Das Schaumodell im Physik-Foyer aus **magnetischen Dipolen auf einem Dreiecksgitter**.

Konfiguration	$\begin{array}{c} \uparrow \\ \uparrow \end{array}$	$\begin{array}{c} \uparrow \\ \downarrow \end{array}$	$\begin{array}{c} \uparrow \\ \uparrow \end{array}$	$\begin{array}{c} \uparrow \\ \downarrow \end{array}$
Energie	$[...] = -2$	$[...] = -1$	$[...] = 1$	$[...] = 2$

Tabelle 9.1: Relative Stärke der Dipol-Dipol-Wechselwirkung bei verschiedenen Dipol-Anordnungen. Dipol-Ketten geben die niedrigste Energie

Aus der Elektrostatik-Vorlesung ist bekannt, dass Dipole eine Anordnung in Ketten bevorzugen, da ihre Wechselwirkung orientierungsabhängig ist, $E \sim r^{-3} \left[-3(\vec{m} \cdot \hat{r})(\vec{n} \cdot \hat{r}) + (\vec{m} \cdot \vec{n}) \right]$, für zwei Dipole \vec{m} und \vec{n} , siehe Tab. 9.1.

Bei einem Dreiecksgitter sieht daher eine typische Konfiguration niedriger Energie wie auf dem Bild rechts aus, wo der äußere Ring von Dipolen eine Art Kette bildet. Dann sind aber alle Orientierungen des inneren Dipols energetisch fast gleich gut. Damit gibt es kein klares Minimum, sondern viele metastabile Minima mit ähnlicher Energie. Man spricht auch von "Frustration" des inneren Dipols.



- b) Ein anderes Beispiel für Frustration sind **Spingläser**, d.h. Magneten mit zufällig ausgewählten Kopplungsstärken, z.B. das 2D Ising-Modell mit **Zufallskopplungen**:

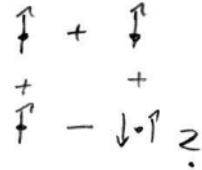
$$\mathcal{H} = - \sum_{\langle ij \rangle} J_{ij} s_i s_j.$$

Beim Ising-Modell haben wir nächste Nachbar Kopplungen mit identischer Stärke $J_{ij} = J > 0$. Bei einem Spinglas werden die J_{ij} dagegen **zufällig** z.B. aus einer Gaußverteilung mit Wahrscheinlichkeiten

$$P(J_{ij}) \sim e^{-J_{ij}^2/2\Delta^2}$$

gezogen. Dann gilt bei Mittelung $\overline{\dots}$ über viele verschiedene J_{ij} -Konfigurationen: $\overline{J_{ij}} = 0$ und $\overline{J_{ij}^2} = \Delta^2$.

Dann haben die J_{ij} auch zufällige Vorzeichen. Beispielsweise kann es Kopplungen mit Vorzeichen wie für das Spinquadrat auf dem Bild rechts geben. Dann ist für den Spin rechts unten nicht klar, ob er nach oben oder unten zeigt. Dieser Spin ist also wieder "frustriert", und es gibt mehrere metastabile Minima.



Für solche Spingläser ist das Auffinden des tatsächlichen Grundzustandes ein notorisch schwieriges Minimierungsproblem.

Variationsprinzipien

In der Quantenmechanik kennt man das **Ritzsche Variationsprinzip**

$$E_0 \leq \frac{\langle \psi_{\vec{\alpha}} | \hat{H} | \psi_{\vec{\alpha}} \rangle}{\langle \psi_{\vec{\alpha}} | \psi_{\vec{\alpha}} \rangle} \quad (9.2)$$

mit einem Variationsansatz $|\psi_{\vec{\alpha}}\rangle$ für den Grundzustand mit Variationsparametern $\vec{\alpha}$. Um eine möglichst gute Approximation an den Grundzustand E_0 zu erhalten wird dann die rechte Seite bezgl. der Parameter $\vec{\alpha}$ minimiert.

Auch in der statistischen Physik gibt es solche Variationsverfahren. Die sogenannte **Bogoliubov-Ungleichung**

$$F \leq F_{\vec{\alpha}} + \langle \mathcal{H} - \mathcal{H}_{\vec{\alpha}} \rangle_{\mathcal{H}_{\vec{\alpha}}} \quad (9.3)$$

Dabei ist \mathcal{H} der "echte" Hamiltonian und $\mathcal{H}_{\vec{\alpha}}$ ein Variationshamiltonian mit Variationsparametern $\vec{\alpha}$. Dieser Variationshamiltonian sollte im Gegensatz zum echten Hamiltonian hinreichend einfach sein, so dass Mittelwerte $\langle \dots \rangle_{\mathcal{H}_{\vec{\alpha}}}$ bezgl. dieses Hamiltonians und die zugehörige freie Energie $F_{\vec{\alpha}}$ berechnet werden können. Um eine möglichst gute Approximation an die echte freie Energie F zu erhalten, wird dann die rechte Seite bezgl. der Parameter $\vec{\alpha}$ minimiert.

Generell lassen sich numerische Minimierungsverfahren für $f(\vec{x})$ nach folgenden Kriterien einteilen:

- Anzahl der Variablen: eine Dimension x oder mehrere Dimensionen \vec{x} .
- Verwendung der Ableitungen $\vec{\nabla} f$ (oder sogar der zweiten Ableitungen) oder nicht.

Wie obige Beispiele zeigen, sind wir in der Physik oft am nicht-trivialen Fall der hoch-dimensionalen Energieminimierung interessiert. Auch die Verfahren in mehreren Dimensionen basieren aber auf (möglichst robusten) Verfahren zur Minimierung in einer Dimension, die dann in verschiedene Minimierungsrichtungen angewandt werden.

9.1 Intervallhalbierung, Goldener Schnitt

Die Intervallhalbierung bzw. der goldene Schnitt sind robuste iterative Verfahren zur Minimierung einer Funktion einer Variablen.

Wir beginnen mit Funktionen $f(x)$ einer Variablen x . Ist die Ableitung $f'(x)$ bekannt, können natürlich auch Verfahren zur Nullstellensuche aus Kapitel 7.2 auf die Gleichung $f'(x) = 0$ angewendet werden, um Extrema zu finden. Um das schnelle **Newton-Raphson Verfahren** (mit Konvergenz vom Grad $p = 2$) zu verwenden, ist dann auch Information über die zweite Ableitung $f''(x)$ notwendig. Das Newton-Raphson Verfahren kann dabei die gleichen Probleme entwickeln wie bei der Nullstellensuche, die zum Verfehlen des Minimums führen können.

Wir wollen hier zwei robuste Verfahren vorstellen, die keine Information über die Ableitungen $f'(x)$ verwenden, die **Intervallhalbierung** oder der **goldene Schnitt**.

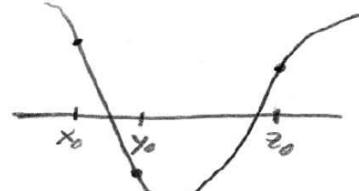
Beide Verfahren sind iterativ und laufen in folgenden Schritten ab:

- 1) Wie bei der Nullstellensuche sollte das Minimum zunächst "eingeklammert" werden:

Dazu werden 3 Punkte $x_0 < y_0 < z_0$ bestimmt, so dass

$$f(x_0) > f(y_0) \text{ und } f(z_0) > f(y_0). \quad (9.4)$$

Im Laufe des Verfahrens bleibt das Minimum immer eingeklammert.



- 2) Im n -ten Iterationsschritt generieren wir aus einer Klammer $x_n < y_n < z_n$ einen neuen Punkt $u_{n+1} \in [x_n, z_n]$.

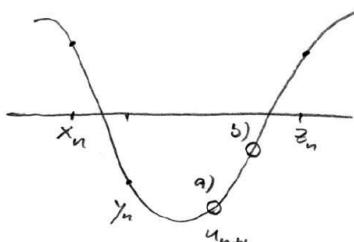
Dazu werden wir 2 Verfahren diskutieren: Intervallhalbierung und goldener Schnitt.

- 3) Dann generieren wir eine neue Klammer.

Für $u_{n+1} \in [y_n, z_n]$:

- a) Wenn $f(u_{n+1}) < f(y_n)$,
dann $(x_{n+1}, y_{n+1}, z_{n+1}) = (y_n, u_{n+1}, z_n)$.
- b) Wenn $f(u_{n+1}) > f(y_n)$,
dann $(x_{n+1}, y_{n+1}, z_{n+1}) = (x_n, y_n, u_{n+1})$.

Und analog für den anderen Fall $u_{n+1} \in [x_n, y_n]$.



- 4) Ist $z_{n+1} - x_{n+1} < \text{Genauigkeitsziel } \varepsilon$ Abbruch, sonst wieder weiter mit Schritt 2).

Die Verfahren in Schritt 2) sind:

- a) **Intervallhalbierung:**

Halbiere mit u_{n+1} das längere der Intervalle $[x_n, y_n]$ oder $[y_n, z_n]$:

$$u_{n+1} = \frac{x_n + y_n}{2} \text{ oder } u_{n+1} = \frac{y_n + z_n}{2}. \quad (9.5)$$

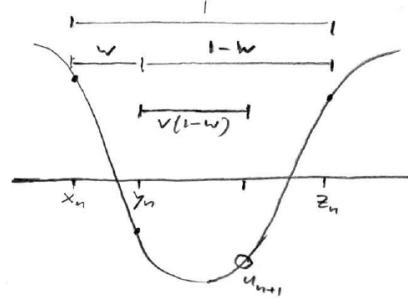
b) **Goldener Schnitt:**

Sei $[y_n, z_n]$ das längere Intervall, also

$$\frac{y_n - x_n}{z_n - x_n} = w < \frac{1}{2}. \quad (9.6)$$

Wähle den Punkt u_{n+1} im längeren Intervall bei Bruchteil v (von y_n aus)

$$\frac{u_{n+1} - y_n}{z_n - y_n} = v. \quad (9.7)$$



Frage: Wie ist Parameter v optimal zu wählen?

Wenn $f(u_{n+1}) < f(y_n)$, ergibt Schritt 3a) des Algorithmus eine Intervallverkürzung von

$$\frac{z_{n+1} - x_{n+1}}{z_n - x_n} = \frac{z_n - y_n}{z_n - x_n} \stackrel{(9.6)}{=} \frac{1-w}{1}. \quad (9.6)$$

Wenn $f(u_{n+1}) > f(y_n)$, ergibt Schritt 3b) des Algorithmus eine Intervallverkürzung von

$$\frac{z_{n+1} - x_{n+1}}{z_n - x_n} = \frac{u_{n+1} - x_n}{z_n - x_n} \stackrel{(9.6)(9.7)}{=} \frac{w + v(1-w)}{1}.$$

Bei optimaler Wahl von v

(i) sollten beide Möglichkeiten die *gleiche* Verkürzung ergeben, also

$$1-w = w + v(1-w), \quad (9.8)$$

(ii) sollte das neue Teilungsverhältnis v (oder $\frac{v(1-w)}{w+v(1-w)} = v$ nach (9.8)) gleich dem alten Teilungsverhältnis w sein:

$$v = w. \quad (9.9)$$

Aus den Bedingungen (9.8) und (9.9) folgt

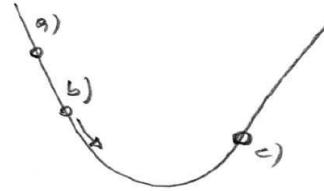
$$\begin{aligned} 1-w &= 2w - w^2, \quad w^2 - 3w + 1 = 0 \\ v = w &= \frac{3}{2} \pm \left(\frac{9}{4} - 1 \right)^{1/2} = \frac{1}{2} (3 - \sqrt{5}) \simeq 0.38197, \end{aligned} \quad (9.10)$$

wobei in der letzten Gleichung wegen $w < 1/2$ das “-“-Vorzeichen gelten muss. Damit wird die Intervalllänge $z_n - x_n$ in jedem Schritt um den Faktor $1-w = \frac{1}{2}(1+\sqrt{5}) \simeq 0.618$ (den **goldenen Schnitt**¹) reduziert. Damit ist die Konvergenz vom Grade $p = 1$.

¹ Beim “goldenen Schnitt” wird ein Intervall so geteilt, dass das Verhältnis des ganzen Intervalls zum größeren Teil gleich dem Verhältnis des größeren zum kleineren Teil ist. Bei einer Aufteilung eines Intervalls der Länge 1 in $w < 1/2$ und $1-w > 1/2$ heißt das $1/(1-w) = (1-w)/w$ oder $w = (1-w)^2$, was wieder auf $w^2 - 3w + 1 = 0$ führt.

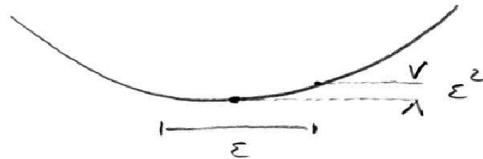
In Schritt 1) benötigen wir noch ein Verfahren zur Konstruktion einer ersten Klammer $x_0 < y_0 < z_0$:

- Rate Startstelle.
- Wähle benachbarte Stelle.
- Gehe in Abwärtsrichtung, bis es wieder aufwärts geht.



Das gesamte Intervallhalbierungs- oder Goldene Schnitt-Verfahren findet **immer** ein Minimum, aber Achtung ist bei der Wahl des Genauigkeitsziels ε geboten:

Minima sind typischerweise parabolisch, wie rechts auf der Abbildung gezeigt. Wenn $\varepsilon^2 <$ Rechengenauigkeit, ergibt der Größenvergleich bei Auswertungen der Funktion $f(x)$ in einem Intervall der Länge ε dann Zufallsergebnisse.



9.2 Funktionen mehrerer Variablen

Die mehrdimensionale Minimierung lässt sich auf wiederholte eindimensionale Minimierung in verschiedene Richtungen zurückführen, die allerdings zueinander konjugiert sein sollten. Das Powell-Verfahren generiert konjugierte Richtungen ohne Gradienteninformation zu verwenden. Steepest Descent beruht auf Gradienteninformation, allerdings ohne konjugierte Richtungen zu erzeugen. Das beste Verfahren sind konjugierte Gradienten.

Nun betrachten wir Funktionen $f(\vec{x})$ **mehrerer** Variablen $\vec{x} \in \mathbb{R}^N$. Die Idee ist immer, die eindimensionalen Verfahren aus dem vorigen Abschnitt 9.1 auf eine Minimierung in verschiedene Richtungen $\vec{p}_0, \vec{p}_1, \dots$ nacheinander anzuwenden. Dabei wird jeweils die eindimensionale Funktion

$$F_i(\lambda) = f(\vec{x}_i + \lambda \vec{p}_i) \quad i = 0, 1, \dots$$

bezgl. λ minimiert, was dann den neuen Startpunkt $\vec{x}_{i+1} = \vec{x}_i + \lambda_{\min} \vec{p}_i$ für die nächste Minimierung in die nächste Richtung \vec{p}_{i+1} generiert.

9.2.1 Konjugierte Richtungen

Das offensichtlich wichtigste Problem hierbei ist die optimale Wahl der Richtungen \vec{p}_i . **Notwendig** ist auf jeden Fall, dass die \vec{p}_i den gesamten \mathbb{R}^N aufspannen. Die einfachste Wahl, die diese notwendige Bedingung erfüllt, sind die kartesischen Einheitsvektoren $\vec{p}_i = \vec{e}_i$ für $i = 1, \dots, N$. Nach N Minimierungsschritten würde man wieder bei $\vec{p}_{N+i} = \vec{p}_i + \vec{e}_i$ starten bis das Verfahren konvergiert.

Das Problem bei dieser einfachen Wahl ist, dass die Richtungen "unangepasst" sind, wie man in der Abbildung rechts sieht (Linien sind Höhenlinien von $f(\vec{x})$). Nach jeder Minimierung in Richtung \vec{p}_i enden wir wegen

$$0 = \partial_\lambda F_i(\lambda) = \vec{\nabla} f(\vec{x}_{i+1}) \cdot \vec{p}_i$$

in einem Punkt senkrecht zum lokalen Gradienten und damit *tangential* zur lokalen Höhenlinie (siehe Abbildung rechts); dies gilt für alle Verfahren. Es sind viele Minimierungsschritte nötig, weil die Minimierung in Richtung \vec{e}_2 wieder die Minimaleigenschaft bezgl. der Richtung \vec{e}_1 zerstört, usw.

Daher müssen wir sogenannte **konjugierte Richtungen** \vec{p}_i finden; das sind Richtungen, die sich beim Minimieren nicht „stören“. Um herauszuarbeiten, was das heißt, betrachten wir die Taylorentwicklung von $f(\vec{x})$ um einen beliebigen Punkt \vec{P} bis zur 2. Ordnung:

$$f(\vec{P} + \vec{y}) \approx f(\vec{P}) - \vec{b} \cdot \vec{y} + \frac{1}{2} \vec{y}^t \cdot \underline{\underline{A}} \cdot \vec{y} \equiv g(\vec{y})$$

mit $\vec{b} = -\nabla f|_{\vec{P}}$ und $a_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}|_{\vec{P}}$. (Hesse-Matrix)

$g(\vec{y})$ ist eine quadratische Form mit Gradient

$$\vec{\nabla}g = -\vec{b} + \underline{\underline{A}} \cdot \vec{y}.$$

Wenn wir $g(\vec{y})$ in Richtung \vec{p}_0 und \vec{p}_1 minimieren wollen, erhalten wir

$$G(\lambda, \mu) = g(\lambda \vec{p}_0 + \mu \vec{p}_1)$$

$$0 \stackrel{!}{=} \partial_\mu G = \vec{\nabla}g|_{\lambda \vec{p}_0 + \mu \vec{p}_1} \cdot \vec{p}_1 = [-\vec{b} + \lambda \underline{\underline{A}} \cdot \vec{p}_0 + \mu \underline{\underline{A}} \cdot \vec{p}_1] \cdot \vec{p}_1.$$

Wir sehen, dass die Minimierung in Richtung \vec{p}_1 nur dann unabhängig von λ und damit von der Minimierung in Richtung \vec{p}_0 ist, wenn der gemischte Term verschwindet, d.h. wenn

$$(\underline{\underline{A}} \cdot \vec{p}_0) \cdot \vec{p}_1 = \vec{p}_1^t \cdot \underline{\underline{A}} \cdot \vec{p}_0 = 0.$$

Solche „ $\underline{\underline{A}}$ -orthogonalen“ Richtungen heißen **konjugiert**. Wenn der Punkt \vec{P} das Minimum selbst ist, sind die konjugierten Richtungen die Hauptachsen der quadratischen Form zur Matrix $\underline{\underline{A}}$. Für die typischen elliptischen Höhenlinien um ein Minimum in zwei Dimensionen sind dies genau die Hauptachsen der Ellipse.

Das Ziel bei der mehrdimensionalen Minimierung ist also immer, N konjugierte Richtungen \vec{p}_i mit

$$\boxed{\vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j = 0 \quad \forall i \neq j} \quad (9.11)$$

zu konstruieren. Dann gelingt die Minimierung einer quadratischen Form in *höchstens* N Schritten!

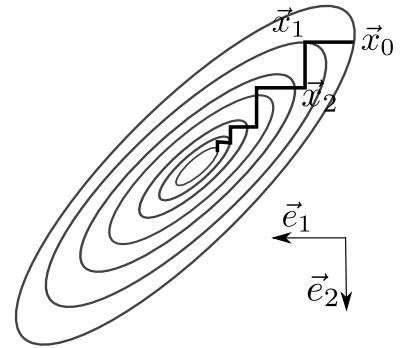
9.2.2 Powell-Verfahren

Das **Powell-Verfahren** benutzt **keine** Information über Gradienten von f und ist folgendermaßen definiert:

- 1) Wähle zuerst N Richtungen $\vec{p}_i = \vec{e}_i$ für $i = 1, \dots, N$.
- 2) Starte bei \vec{x}_0 und minimiere für $i = 1, \dots, N$

$$f(\vec{x}_{i-1} + \lambda \vec{p}_i) \Rightarrow \text{Minimum bei } \vec{x}_i \equiv \vec{x}_{i-1} + \lambda_{i,\min} \vec{p}_i.$$

- 3) Setze $\vec{p}_i \equiv \vec{p}_{i+1}$ für $i = 1, \dots, N-1$.
Setze $\vec{p}_N \equiv \vec{x}_N - \vec{x}_0$, also auf die **mittlere Richtung**, die der Algorithmus bisher genommen hat.



4) Minimiere

$$f(\vec{x}_N + \lambda \vec{p}_N) \Rightarrow \text{Minimum bei } \vec{x}_0 \equiv \vec{x}_N + \lambda_{\min} \vec{p}_N.$$

Danach wieder Schritt 2)

Für eine quadratische Form führt das Verfahren nach $N(N+1) = \mathcal{O}(N^2)$ Minimierungsschritten zu konjugierten Richtungen \vec{p}_i . Wir werden die Konvergenz zu konjugierten Richtungen hier nicht beweisen.

Für beliebige Funktionen $f(\vec{x})$ wird das Verfahren solange iteriert, bis Konvergenz erreicht wird. In der Praxis zeigt sich dabei oft das Problem, dass die Mittelwertbildung im Laufe der Iteration zu linear abhängigen Richtungen tendiert. Dann muss der Richtungssatz geeignet neu initialisiert werden.

9.2.3 Steepest Descent

Nun wenden wir uns Verfahren zu, die Information über die **Gradienten** $\vec{\nabla} f(\vec{x})$ verwenden. Die einfachste Idee ist dabei der Richtung $-\vec{\nabla} f(\vec{x})$ zu folgen, die ja die Richtung des schnellsten Abstiegs angibt.

Dieses Verfahren heißt **Steepest Descent**:

1) Wähle $\vec{p}_i = -\vec{\nabla} f(\vec{x}_i)$.

2) Minimiere

$$f(\vec{x}_i + \lambda \vec{p}_i) \Rightarrow \text{Minimum bei } \vec{x}_{i+1} \equiv \vec{x}_i + \lambda_{\min} \vec{p}_i.$$

Danach wieder Schritt 1) bis zur Konvergenz.

Hierzu ist anzumerken, dass im jeweiligen Minimum

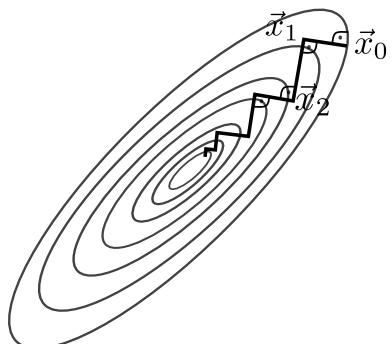
$$0 = \frac{d}{d\lambda} f(\vec{x}_i + \lambda \vec{p}_i) = \vec{\nabla} f(\vec{x}_{i+1}) \cdot \vec{p}_i$$

gilt, d.h. aufeinanderfolgende Richtungen sind orthogonal

$$\vec{p}_{i+1} \cdot \vec{p}_i = 0$$

(siehe Abb. rechts).

Das bedeutet aber herkömmliche 1-Orthogonalität, und **nicht A-Orthogonalität**, also sind die so entstehenden Richtungen **nicht konjugiert**. Die Minimierung nimmt daher wieder einen nicht optimalen *rechteckigen* Zickzackweg, der viele Minimierungsschritte benötigt, da die Richtungen nicht konjugiert sind, siehe Abb. rechts. Bei der Abb. ist zu beachten, dass die Minimierungsrichtungen bei steepest descent auch immer senkrecht auf den Höhenlinien von f stehen wegen $\vec{\nabla} f \perp$ Höhenlinie.



9.2.4 Konjugierte Gradienten

Das beste Verfahren ist das **konjugierte Gradienten Verfahren**, das iterativ konjugierte Richtungen \vec{p}_i aus den Gradienten $\vec{\nabla} f(\vec{x}_i)$ macht, **ohne** dabei die lokale Hesse-Matrix A in (9.11) explizit zu benutzen:

- 1) Starte mit \vec{x}_0 und $\vec{p}_0 = \vec{g}_0 = -\vec{\nabla}f(\vec{x}_0)$.
 - 2) Minimiere $f(\vec{x}_i + \lambda \vec{p}_i)$.
Daraus erhält man $\vec{x}_{i+1} = \vec{x}_i + \lambda_{i,\min} \vec{p}_i$ und $\vec{g}_{i+1} = -\vec{\nabla}f(\vec{x}_{i+1})$.
 - 3) Die neue Richtung ist
- $$\vec{p}_{i+1} = \vec{g}_{i+1} + \mu_i \vec{p}_i \quad \text{mit} \quad \mu_i = \frac{\vec{g}_{i+1} \cdot \vec{g}_{i+1}}{\vec{g}_i \cdot \vec{g}_i}.$$
- 4) Weiter mit 2) bis zur Konvergenz.

Wir wollen versuchen, uns klarzumachen, dass die so definierten Richtungen \vec{p}_i tatsächlich konjugiert sind, zumindest wenn $f(\vec{x})$ eine quadratische Form ist:

- $\vec{g}_i = -\vec{\nabla}f(\vec{x}_i)$ sind die Gradienten.
- Im Minimum in Schritt 2) gilt:

$$\vec{\nabla}f(\vec{x}_i + \lambda_{i,\min} \vec{p}_i) \cdot \vec{p}_i = -\vec{g}_{i+1} \cdot \vec{p}_i = 0.$$

- $f(\vec{x})$ lässt sich lokal um das Minimum (bei $\vec{x} = 0$ o.B.d.A) näherungsweise als quadratische Form

$$f(\vec{x}) \approx c - \vec{b} \cdot \vec{x} + \frac{1}{2} \vec{x}^t \cdot \underline{\underline{A}} \cdot \vec{x}$$

schreiben. Dann kann der Schritt 2) explizit gemacht werden:

$$\begin{aligned} \vec{\nabla}f(\vec{x}) &= -\vec{b} + \underline{\underline{A}} \cdot \vec{x} \\ \vec{g}_{i+1} &= -\vec{\nabla}f(\vec{x}_i + \lambda_i \vec{p}_i) = \vec{b} - \underline{\underline{A}} \cdot \vec{x}_i - \lambda_i \underline{\underline{A}} \cdot \vec{p}_i \end{aligned} \tag{9.12}$$

$$= \vec{g}_i - \lambda_i \underline{\underline{A}} \cdot \vec{p}_i \tag{9.13}$$

und

$$\begin{aligned} 0 &\stackrel{!}{=} \partial_{\lambda_i} f(\vec{x}_i + \lambda_i \vec{p}_i) = \vec{\nabla}f(\vec{x}_i + \lambda_i \vec{p}_i) \cdot \vec{p}_i \\ &= (-\vec{g}_i + \lambda_i \underline{\underline{A}} \cdot \vec{p}_i) \cdot \vec{p}_i \\ \Rightarrow \lambda_{i,\min} &= \frac{\vec{g}_i \cdot \vec{p}_i}{\vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_i}. \end{aligned} \tag{9.14}$$

- Weiter kann dann induktiv gezeigt werden (längerer Beweis, siehe z.B. [3]), dass folgende Relationen gelten,

- | | |
|---|--------|
| a) $\vec{g}_i \cdot \vec{p}_j = 0$ für $j < i$
$\vec{g}_i \cdot \vec{p}_i = \vec{g}_i \cdot \vec{g}_i$ für $j = i$ | (9.15) |
| b) $\vec{g}_i \cdot \vec{g}_j = 0$ für $j < i$ | |
| c) $\vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j = 0$ für $j < i$, | |

wenn Schritt 2) bis zum Index i ausgeführt wurde.

Als Beispiel wollen wir induktiv (9.15 c) für $i + 1$ zeigen, d.h. die Eigenschaft, dass die \vec{p}_i

konjugiert sind:

$$\begin{aligned}
 \vec{p}_{i+1}^t \cdot \underline{\underline{A}} \cdot \vec{p}_j &\stackrel{\text{Def. } \vec{p}_{i+1} = \vec{g}_{i+1} + \mu_i \vec{p}_i}{=} \vec{g}_{i+1}^t \cdot \underline{\underline{A}} \cdot \vec{p}_j + \mu_i \vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j \\
 &\stackrel{(9.13)}{=} \vec{g}_{i+1}^t \cdot \frac{1}{\lambda_j} (\vec{g}_j - \vec{g}_{j+1}) + \mu_i \vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j \\
 &\stackrel{\text{Def. } \vec{p}_{i+1} = \vec{g}_{i+1} + \mu_i \vec{p}_i}{=} \vec{g}_{i+1}^t \cdot \frac{1}{\lambda_j} (-\vec{p}_{j+1} + \mu_j \vec{p}_j + -\vec{p}_j - \mu_{j-1} \vec{p}_{j-1}) + \mu_i \vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j \\
 &\stackrel{(9.15) \text{ fuer } i+1}{=} -\frac{1}{\lambda_j} \vec{g}_{i+1} \cdot \vec{p}_{j+1} + \mu_i \vec{p}_i^t \cdot \underline{\underline{A}} \cdot \vec{p}_j \\
 &= \begin{cases} i > j : 0 \text{ wegen (9.15) a) fuer } i+1, (9.15) c) \text{ fuer } i \\ i < j : 0 \text{ wegen (9.15) a) fuer } i+1, \text{ Def. von } \mu_i, \lambda_i \end{cases}
 \end{aligned}$$

- Wegen (9.15) c) werden konjugierte Richtungen generiert und die Minimierung erfolgt in höchstens N Schritten, wenn $f(\vec{x})$ eine quadratische Form ist.
- Da $f(\vec{x})$ nur näherungsweise eine quadratische Form ist, muss iteriert werden. Trotzdem reichen in der Regel $\mathcal{O}(N)$ Operationen.

9.3 Literaturverzeichnis Kapitel 9

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis*. 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.

9.4 Übungen Kapitel 9

1. Nicht-lineare Feder

Eine nicht-lineare Feder habe die potentielle Energie

$$u(x) = 2.715 - 5.132x + 18.88x^2 - 38.82x^3 + 47.24x^4 - 32.26x^5 + 9.711x^6$$

Bevor Sie die Aufgaben bearbeiten, ist es hilfreich, die Funktion $u(x)$ zu plotten, um sich einen Eindruck zu verschaffen und nachher die Ergebnisse von a) und b) auf Plausibilität prüfen zu können.

- Berechnen Sie numerisch die Gleichgewichtslage der Feder mit Intervallhalbierung und goldenem Schnitt. Sie dürfen die Information benutzen, dass es nur ein Minimum gibt.
- Berechnen Sie zum Vergleich die Nullstellen der ersten Ableitung $u'(x)$ (Ableitung analytisch berechnen) mit den Methoden aus Kapitel 7. Intervallhalbierung, Regula Falsi und Newton-Raphson-Methode (für Newton-Raphson müssen Sie vorher auch die zweite Ableitung $u''(x)$ analytisch berechnen).

10 Zufallszahlen

Literatur zu diesem Teil:

Zu empfehlen sind die Numerical Recipes [1][2]. Instruktiv sind dabei auch die Unterschiede zwischen 2. und 3. Auflage. Kurze Kapitel auch in Landau und Binder [3], Krauth [4] oder im Hjorth-Jensen Skript [5].

10.1 Zufallszahlengeneratoren

Wir stellen verschiedene Zufallszahlengeneratoren, insbesondere linear kongruente, Xorshift-Generatoren und Kombinationen vor. Wir erläutern den Marsaglia-Effekt für linear kongruente Generatoren und andere Gütekriterien.

Wir werden uns in den folgenden Kapiteln [11] (Monte-Carlo Simulation) und [13] (stochastische Bewegungsgleichung) mit stochastischen Methoden auseinandersetzen, um die statistische Physik von Vielteilchensystemen zu simulieren. Diese Methoden benutzen im Gegensatz zur MD-Simulation aus Kapitel 5 zufällige Bewegungen oder Kräfte, insbesondere um die thermischen Fluktuationen im kanonischen Ensemble zu simulieren. Dazu benötigen wir **Zufallszahlen**: Wir müssen in der Lage sein, im Computer zufällige Stichproben x aus einer vorgegebenen Verteilung $p(x)$ zu generieren. Am wichtigsten sind dabei glücklicherweise einfache Gleichverteilungen. Dazu benötigen wir vor allem **Zufallszahlengeneratoren**.

10.1.1 Echter Zufall

Computer sind **deterministisch** und können daher keinen “echten” Zufall erzeugen. Mittels Computeralgorithmen erzeugte Zufallszahlen sind daher immer **Pseudo-Zufallszahlen**. Echten Zufall gibt es nur in physikalischen Systemen.

Beispiele sind zum einen **quantenmechanische Experimente**, die nicht deterministisch gedeutet werden können. Im Rahmen der Kopenhagener Deutung gibt es das Postulat, dass mögliche Messwerte Eigenwerte a_n des zugehörigen Operators \hat{A} sind (mit zugehörigen Eigenzuständen $|a_n\rangle$) und lediglich die **Wahrscheinlichkeit** $|\langle a_n | \psi \rangle|^2$ bekannt ist, in einem quantenmechanischen Zustand $|\psi\rangle$ den Messwert a_n zu erhalten. Die Zeitentwicklung eines **chaotischen klassischen Systems** ist dagegen zwar prinzipiell deterministisch, aber prinzipiell auch beliebig schlecht vorhersagbar.

Beide Phänomene können zur Erzeugung **echter Zufallszahlen** genutzt werden. Dazu bedarf es dann aber sehr spezieller Hardware. Ein etwas kurioses Beispiel ist der Zufallszahlengenerator **La-varand** von SGI [6], der auf der chaotischen Fluidodynamik in einer **Lavalampe** beruht, siehe Abb. 10.1. Aus den Digitalaufnahmen der Lavalampe wird die Information für jeden Pixel ausgelesen und zu einer großen Binärzahl aneinander gereiht. Auf diese wird dann noch eine kryptographische Hashfunktion (surjektiv, nicht invertierbar) angewandt, um eine kurze Zahl zu gewinnen, die dann wiederum als Seed für einen Pseudozufallszahlengenerator fungiert. Durch diese Kopplung an die chaotischen Fluidodynamik kann dies als “echter Zufall” angesehen werden. Offensichtlich involviert dieses Verfahren aber eine Menge “Hardware” und ist auch nicht besonders schnell. Ähnliche andere Verfahren beruhen auf dem thermischen Rauschen einer Digitalkamera, thermischem Rauschen in

einem Widerstand oder auch radioaktiven Zerfällen (hier ist Quantenmechanik im Spiel). Alle diese Verfahren sind aber viel zu aufwendig und langsam.

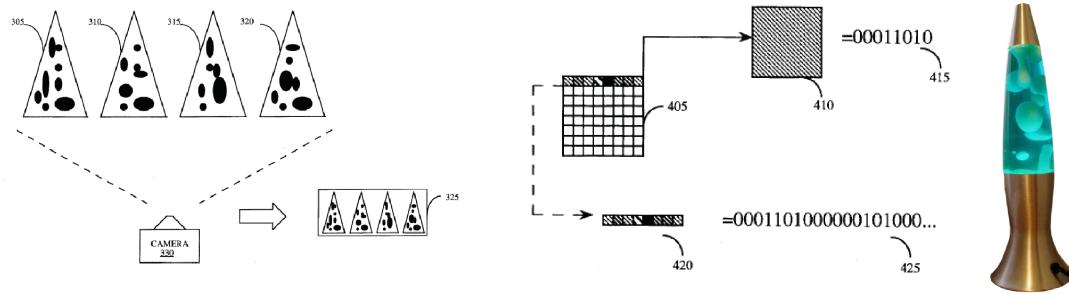


Abbildung 10.1: Links, Mitte: Prinzip des SGI-Lavalampengenerators (aus [6]): Aus einer Digitalaufnahme wird durch Aneinanderreihen aller digitalen Pixelinformationen eine lange Binärzahl. Auf diese wird noch eine Hashfunktion angewandt und deren Output als Seed für einen Pseudozufallszahlengenerator verwendet. Rechts: Lavalampe.

10.1.2 Pseudo-Zufallszahlengeneratoren

In der Regel werden keine “echten” Zufallszahlen erzeugt, sondern **Pseudo-Zufallszahlen** mit Hilfe bestimmter Algorithmen. Später diskutieren wir linear kongruente Generatoren und Xorshift-Generatoren im Detail. Wir starten mit einigen grundsätzlichen Eigenschaften von **Pseudo-Zufallszahlengeneratoren (pseudo random number generators = PRNGs)**:

- 1) PRNGs erzeugen **gleichverteilte Zufallszahlen**. Beispielsweise erzeugt der in C eingebaute `int rand(void)` gleichverteilte integers im Bereich $0 \dots \text{RANDMAX}$. `(double)rand() / (\text{RANDMAX} + 1.0)` erzeugt dann gleichverteilte double in $[0, 1[$. Gleichverteilung in $[0, 1[$ ist der Standard.
- 2) PRNGs sind **iterativ**, sie benötigen einen **Startwert oder Seed** und generieren dann eine Folge von Zahlen durch **wiederholtes Aufrufen**. Im Beispiel des `rand()` in C wird der Seed mit `void srand(unsigned int seed)` gesetzt. Also wird am Anfang einmal initialisiert mit `srand(beliebig)`, danach erzeugen Aufrufe `rand()` zufällige integer.
- 3) PRNGs erzeugen **reproduzierbare Sequenzen** bei gleichem Seed. Die Algorithmen sind deterministisch. Dies zeigt bereits, dass der Zufall nicht “echt” ist, kann aber hilfreich sein beim Testen (Reproduzierbarkeit).
- 4) PRNGs wiederholen sich irgendwann nach einer gewissen **Periodenlänge**. Dies ist der andere Aspekt, der zeigt, dass der Zufall nicht “echt” ist. Die Periodenlänge sollte natürlich *möglichst groß* sein. Dies gilt insbesondere für Monte-Carlo Simulationen, wo z.B. bei einer ernstzunehmenden Monte-Carlo Simulation des 3D Ising-Modells leicht 10^{15} Zufallszahlen benötigt werden (entspricht 10^6 Sweeps eines Systems mit 1000^3 Gitterplätzen).

Die Eigenschaften 3) und 4) scheinen problematisch, lassen sich aber prinzipiell nicht verhindern, wie wir sehen werden.

Für unsere “physikalischen” Zwecke brauchen und werden wir auch nicht auf kryptographische Aspekte von PRNGs eingehen (d.h. die Frage, ob die nächste Zufallszahl vorhersagbar ist); die vorgestellten Generatoren sind i.Allg. nicht kryptographisch sicher.

10.1.3 Linear kongruente Generatoren

Die wichtigsten PRNGs sind die **linear kongruenten Generatoren**, die alle mit Rekursionen

$$r_{n+1} = (ar_n + c) \bmod m \quad (10.1)$$

arbeiten, der sogenannten **Lehmer-Sequenz**. Dabei sind alle Zahlen, d.h. r_n , der **Multiplikator** a , das **Inkrement** c und der **Modulus** m integers. Eingebaute PRNG wie der bereits erwähnte `rand()` in C sind meist von diesem einfachen Typus. Wir werden bald einsehen, dass man diese schlichten PRNGs **niemals** für ernsthaftes wissenschaftliches Rechnen verwenden sollte, höchstens zum Testen.

Generatoren vom Typ (10.1) haben folgende Eigenschaften:

- Die **Periode** des Generators (10.1) kann höchstens m betragen wegen der $\bmod m$ Operation.
- Bei einer geeigneten Wahl von a , c und m kann diese maximale Periodenlänge auch erreicht werden.¹
- Die Zufallsintegers liegen im Bereich $0, \dots, m - 1$. Ein Zufalls float/double in $[0, 1[$ wird durch Division durch m erhalten.
- Es sollte gelten $a(m - 1) <$ größte Integer (2^{32} oder 2^{64}) wegen Overflow-Fehlern bei der Berechnung. Es gibt aber Tricks, um die Rekursion auch für beliebige 32bit-Zahlen a , m nur mit 32bit durchzuführen (Schrage 1979, siehe Numerical Recipes [1, 2]).
- Die Numerical Recipes (2. Ausgabe) empfehlen die Implementation `ran1()` mit

$$a = 7^5 = 16807, \quad c = 0 \quad \text{und} \quad m = 2^{31} - 1 \quad (10.2)$$

plus einem zusätzlichem ‘‘Mischen’’ der letzten 32 Werte, da linear kongruente Generatoren zu Korrelationen in den niedrigen Bits neigen: Wegen $a \ll m$ folgt auf eine kleine Zahl wieder eine relativ kleine Zahl. Linear kongruente Generatoren mit $c = 0$ wie in (10.2) heißen auch **multiplikative linear kongruente Generatoren**.

Die Periode von `ran1()` ist (ohne Mischen) $2^{31} - 2 \sim 10^9$, also maximal.

Wir wollen an Hand der linear kongruenten Generatoren einige **Gütekriterien** für PRNG diskutieren. Dies sind die **Geschwindigkeit** (die möglichst groß sein soll), die **Periodenlänge** (die möglichst lang sein soll), aber auch die **Abwesenheit von Korrelationen** zwischen aufeinanderfolgenden Zahlen.

Linear kongruente Generatoren sind sehr schnell (wenige Rechenoperationen). Allerdings ist ihre Periodenlänge etwas kurz, was bei Monte-Carlo Simulationen zu Problemen führen kann. Vielleicht problematischer ist allerdings, dass sie tatsächlich Korrelationen aufweisen in höheren Dimensionen:

Die k -Tupel (r_n, \dots, r_{n+k-1}) von gleichverteilten Zufallszahlen in $[0, 1]$ liegen im k -dimensionalen Einheitswürfel $[0, 1]^k$ in **Hyperebenen**

(10.3)

Dies ist der sogenannte **Marsaglia-Effekt** (Marsaglia 1968 [7]), siehe Abb. 10.2.

Wir wollen uns den Marsaglia-Effekt nur für $k = 2$ -Tupel veranschaulichen (siehe Abb. 10.2, links). Klar ist, dass die Rekursion (10.1) ohne die Modulo-Operation einfach eine Geradengleichung in

¹ Knuth hat gezeigt, dass die Periodenlänge (bei $c \neq 0$) genau dann maximal ist, wenn:

c ist relativ prim zu m (d.h. keine gemeinsamen Primfaktoren)

$a - 1$ ist Vielfaches von jeder Primzahl, die m teilt

$a - 1$ ist ein Vielfaches von 4, falls m ein Vielfaches von 4 ist.

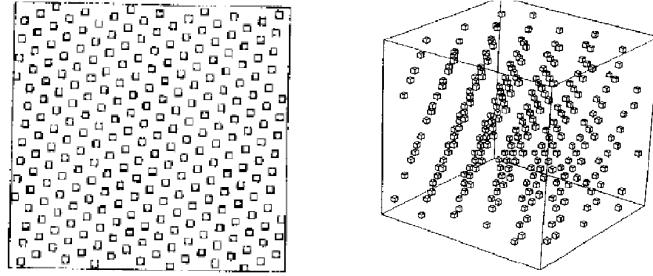


Abbildung 10.2: Marsaglia-Effekt für 2-Tupel (r_n, r_{n+1}) (links) und 3-Tupel (r_n, r_{n+1}, r_{n+2}) (rechts) mit $r_{n+1} = (137r_n + 187) \bmod 256$.

der Ebene (r_n, r_{n+1}) beschreibt und zwar Geraden $(x, y) = x(1, a) + (0, c)$. Die Modulo-Operation verschiebt diese Gerade um ein Vielfaches $k_n m$ von m in y -Richtung:

$$(r_n, r_{n+1}) = r_n(1, a) + (0, c) + k_n m(0, 1).$$

Das Ergebnis ist ein **Marsaglia-Gitter** mit **Gitterebenen**, jede Gitterebene ist charakterisiert durch einen reziproken Gittervektor. Aus der Festkörperphysik ist bekannt, dass der Abstand dieser Gitterebenen $\sim 1/|\text{reziproker Gittervektor}|$ beträgt.

Dies erlaubt uns, ein neues Gütemaß für PRNG einzuführen:

$$\nu_k = 1/(\text{größter Hyberebenen-Abstand}) \quad \text{für } k\text{-Tupel in } [0, 1]^k \quad (10.4)$$

Unser Ziel ist es, ein möglichst großes ν_k (kleiner größerer Ebenen-Abstand) zu erreichen für alle $k = 2, 3, 4, 5, 6, \dots$. Wir wollen in der Festkörperphysik-Sprache also einen möglichst kleinen reziproken Gittervektor im reziproken Marsaglia-Gitter.

Wir können auch eine obere Schranke für ν_k angeben mit einer einfachen Abschätzung. Ein Kristall aus m Atomen in $[0, 1]^k$ hat eine Gitterkonstante $\sim m^{-1/k}$ (wegen $1/m = \text{Gitterkonst}^k$). Also gibt es bestenfalls $m^{1/k}$ ($k-1$)-dimensionale Ebenen mit jeweils $m^{k-1/k}$ Punkten bei gleichmäßiger Verteilung der Atome. Das heißt, dass

$$\nu_k < m^{1/k}$$

ist. Oft sind PRNG (bei bestimmten k) aber viel schlechter.

Eine Strategie, die linear kongruenten Generatoren zu verbessern, ist die Kombination mehrerer verschiedener Generatoren zu sogenannten **zusammengesetzten linearen Kongruenzgeneratoren**. Ein Beispiel dazu ist der **Wichmann-Hill Generator**, der eine Periodenlänge 10^{12} aufweist und auf 3 Rekursionen mit integern x_n , y_n und z_n beruht, um seinen float-Output $u_n \in [0, 1[$ zu erzeugen:

$$\begin{aligned} x_n &= [171x_{n-1}] \pmod{30296} \\ y_n &= [172y_{n-1}] \pmod{30307} \\ z_n &= [170z_{n-1}] \pmod{30323} \\ u_n &= \left[\frac{x_n}{30269} + \frac{y_n}{30307} + \frac{z_n}{30323} \right] \pmod{1} \end{aligned}$$

10.1.4 Xorshift und Kombinationen

Eine modernere Generation von PRNGs sind die **Xorshift-Generatoren** (Marsaglia 2003). Sie beruhen auf Kombinationen von bitweisen xor- und shift-Operationen:

- Der Binäroperator **Xor** ($x \oplus y$) gibt immer 1, wenn x ungleich y , und 0, wenn x gleich y . In C/C++ ist der Operator für bitweises Xor $x \wedge y$.
- Der **Shiftoperator** verschiebt um a bits nach links $x << a$. Die Rechtsverschiebung $x >> a$ entspricht ohne Overflow einer Multiplikation mit 2^a .

Diese Generatoren werden auch in den Numerical Recipes besprochen [2].

PRNGs können weiter verbessert werden durch geeignete **Kombinationen** von PRNGs. Man kann den Output eines PRNG als Input des anderen verwenden oder den Output zweier Generatoren addieren oder den Output zweier Generatoren mit einem Xor verknüpfen.

Die Numerical Recipes empfehlen in der neuen 3. Ausgabe solch Kombinationen. So ist die Empfehlung **Ran()** eine Kombination aus 4 Generatoren, davon sind zwei 64bit-Xorshift und einer linear kongruent.

Wir schließen mit einer Zusammenstellung einiger **Tests** für PRNGs:

- 1) Bereits behandelt wurde der Korrelationstest durch **Plotten von k-Tupeln** im Würfel $[0, 1]^k$. Für linear kongruente Generatoren gibt es den besprochenen "Spektraltest" durch den Abstand der Hyperebenen. Aber auch beliebige andere Generatoren kann man einfach testen, indem man im k-Tupelplot nach "verdächtigen" Strukturen Ausschau hält.
- 2) Im χ^2 -Test oder **Gleichverteilungstest** teilt man das Intervall $[0, 1]$ in M "bins" gleicher Länge. Dann generiert man N Zufallszahlen und misst die Zahl der Zufallszahlen n_i in jedem bin i . Der erwartete Mittelwert über viele Versuche ist $\langle n_i \rangle = N/M$. Die erwartete Varianz $\langle (n_i - \langle n_i \rangle)^2 \rangle = \langle n_i \rangle = N/M$, weil die n_i Poisson-verteilt sein sollten. Außerdem gibt es eine Nebenbedingung zu den M Werten n_i , nämlich $\sum_{i=1}^M n_i = N$ und nur $M - 1$ Werte n_i sind stochastische Variablen ("Freiheitsgrade"). Damit ist der erwartete Wert für $\chi^2 = \sum_{i=1}^M \langle (n_i - \langle n_i \rangle)^2 \rangle / \langle n_i \rangle = M - 1$, wobei $M - 1$ die Zahl der "Freiheitsgrade" ist. Alles, was von diesem erwarteten χ^2 -Wert stark abweicht, ist verdächtig.
- 3) Weitere Tests sind auch der **Maximums-Test**, wo das Maximum von jeweils k Zufallszahlen auch wieder einer bestimmten Verteilung folgen muss oder der **Kollisionstest**, wo man auch wieder M bins anlegt, aber nur $N \ll M$ Zufallszahlen zieht und dann die Zahl der Kollisionen, d.h. j Zahlen in demselben bin zu finden, angibt. Für diese gibt es auch wieder eine Erwartung.

10.2 Erzeugung verschiedener Verteilungen

Der Transformationssatz erlaubt es andere Zufallszahlverteilungen als die Gleichverteilung zu erzeugen. Für gaußverteilte Zufallszahlen diskutieren wir u.a. den Box-Muller-Algorithmus. Die Rückweisungsmethode benötigt nur eine Vergleichsverteilung, die erzeugt werden kann und eine obere Schranke für eine zu erzeugende Verteilung darstellt.

Zufallszahlengeneratoren erzeugen normalerweise **gleichverteilte** Zufallszahlen x im Intervall $[0, 1]$. Damit haben diese x folgende **Wahrscheinlichkeitsdichte** $p(x)$:

$$p(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{sonst} \end{cases} \quad (10.5)$$

Allgemein ist $p(x)dx$ ist die Wahrscheinlichkeit, eine Zufallszahl in $[x, x + dx]$ zu ziehen.

In der Praxis werden oft andere Verteilungen benötigt als die einfache Gleichverteilung (10.5), z.B. eine Gaußverteilung oder Exponentialverteilung.

10.2.1 Transformations- oder Inversionsmethode

Bei der Transformationsmethode wendet man eine Funktion $y = f(x)$ auf Zufallszahlen aus einer Verteilung $p(x)$ an, die auf $x \in [x_1, x_2]$ definiert sei und die man bereits erzeugen kann, z.B. auf ein gleichverteiltes x mit (10.5). Dann gilt für die damit erzeugte Verteilung $\tilde{p}(y)$ der neuen Zufallszahl y (wegen Wahrscheinlichkeitserhaltung):

$$|\tilde{p}(y)dy| = |p(x)dx|$$

$$\boxed{\tilde{p}(y) = p(x) \left| \frac{dx}{dy} \right|} \quad (10.6)$$

Dies können wir uns auch mit Hilfe der Repräsentation einer Wahrscheinlichkeitsverteilung als Mittelwert einer δ -Funktion klarmachen:

$$\begin{aligned} \tilde{p}(y) &= \langle \delta(y - f(x)) \rangle_x = \int_{x_1}^{x_2} p(x) \delta(y - f(x)) \\ &= \int_{f([x_1, x_2])} d\tilde{y} p(f^{-1}(\tilde{y})) |(f^{-1})'(\tilde{y})| \delta(y - \tilde{y}) \\ &= p(f^{-1}(y)) |(f^{-1})'(y)| \end{aligned}$$

$$\boxed{\tilde{p}(y) = \begin{cases} p(f^{-1}(y)) |(f^{-1})'(y)| & y \in f([x_1, x_2]) \\ 0 & \text{sonst} \end{cases}} \quad (10.7)$$

was genau Gl. (10.6) entspricht. Da bei einer Gleichverteilung $p(x) = \text{const}$ die erzeugte Verteilung im Wesentlichen durch die Ableitung der inversen Funktion $(f^{-1})'(y)$ gegeben ist, heißt die Methode auch Inversionsmethode.

Wir schauen uns **Beispiele** an, wo wir die Transformationsmethode jeweils direkt auf die Gleichverteilung (10.5) anwenden:

- 1) $y = e^{-\lambda x}$ oder $f^{-1}(y) = -\frac{1}{\lambda} \ln y$ für die Umkehrfunktion. Wir wenden also eine Exponentialfunktion auch jeden Wert x aus dem PRNG an. Dann gilt nach (10.7) $\tilde{p}(y) = \frac{1}{\lambda y}$ für $y \in]e^{-\lambda}, 1]$ für die transformierte Verteilung für y .
- 2) Wir können auch umgekehrt einen Logarithmus auf die x aus dem PRNG anwenden,

$$y = -\frac{1}{\lambda} \ln x$$

(der PRNG darf dann natürlich keine 0 liefern). Dann gilt $f^{-1}(y) = e^{-\lambda y}$ für die Umkehrfunktion und

$$\tilde{p}(y) = \lambda e^{-\lambda y} \quad (10.8)$$

für die transformierte Verteilung der y auf $]0, \infty[$. Wir erhalten also eine **Exponentialverteilung**.

- 3) Wenden wir ein Potenzgesetz $f(x) = x^{-a}$ an mit Umkehrfunktion $f^{-1}(y) = y^{-1/a}$ bekommen wir

$$\tilde{p}(y) = \frac{1}{a} y^{-1/a-1} \quad (10.9)$$

für $y \in]1, \infty[$, also eine **Potenzverteilung**.

10.2.2 Gaußverteilungen

Bei Gaußverteilungen ist die Lage komplizierter. Eine Verteilung $\tilde{p}(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$ lässt sich nämlich nicht direkt mit der normalen 1-dimensionalen Transformationsmethode erzeugen, sondern nur über die 2-dimensionale Verallgemeinerung

$$\tilde{p}(y_1, y_2) dy_1 dy_2 = p(x_1, x_2) dx_1 dx_2 = p(x_1, x_2) \left| \frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} \right| dy_1 dy_2 \quad (10.10)$$

Wir suchen also eine Abb. $\mathbb{R}^2 \rightarrow \mathbb{R}^2 : \vec{x} \rightarrow \vec{y}(\vec{x})$, so dass die Jacobi-Determinante folgende Form hat:

$$\left| \frac{\partial(x_1, x_2)}{\partial(y_1, y_2)} \right| = \left(\frac{1}{\sqrt{2\pi}} e^{-y_1^2/2} \right) \left(\frac{1}{\sqrt{2\pi}} e^{-y_2^2/2} \right).$$

Dann werden durch Anwendung von $\vec{y}(\vec{x})$ auf (x_1, x_2) gleichverteilt aus $[0, 1]^2$ nämlich jeweils *zwei* gaußverteilte Zahlen y_1 und y_2 generiert.

Die Lösung dieses Problems (Beweis selbst durch Nachrechnen der Jacobi-Determinante) ist der **Box-Muller-Algorithmus**

$$\begin{aligned} y_1 &= \sqrt{-2 \ln x_1} \cos(2\pi x_2) \\ y_2 &= \sqrt{-2 \ln x_1} \sin(2\pi x_2) \end{aligned} \quad (10.11)$$

Wenn die $x_2 \in [0, 1]$ gleichverteilt sind, folgt, dass der Vektor $(\cos(2\pi x_2), \sin(2\pi x_2))$ gleichverteilte Punkte auf der Einheitskreislinie erzeugt. Also können wir den Box-Muller-Algorithmus etwas schneller machen durch die sogenannte **Polarmethode**:

- 1) Ziehe (v_1, v_2) gleichverteilt aus $[-1, 1]^2$ (benutze so etwas wie `2*ran()-1`)
- 2) Wenn $R^2 = v_1^2 + v_2^2 < 1$, dann $x_1 = R^2$, $\cos(2\pi x_2) = v_1/\sqrt{R^2}$ und $\sin(2\pi x_2) = v_2/\sqrt{R^2}$ (es muss nur eine Wurzel gezogen werden).
- 3) Dann einsetzen in (10.11), um gaußverteilte y_1, y_2 zu erzeugen.

Die Wahrscheinlichkeit $p(R)dR$ ist dann der Anteil der Fläche der Kreisrings $2\pi R dR$ an der Einheitskreisfläche π , also gilt $p(R)dR = 2RdR = dR^2$ und damit tatsächlich nach Transformationsmethode (10.6) $\tilde{p}(R^2) = p(R)dR/dR^2 = 1$, d.h. R^2 ist in der Tat gleichverteilt auf $[0, 1[$ in der Polarmethode.

Neben dem Box-Muller-Algorithmus gibt es noch eine viel einfachere, schnellere, dafür allerdings nicht besonders genaue Methode, eine Gaußverteilung zu generieren, die auf dem **zentralen Grenzwertsatz** beruht: Die Summe *vieler, unabhängiger, identisch verteilter* Zufallsvariablen ist **gaußverteilt**.

Etwas genauer kann man den Satz so fassen: Wir ziehen N unabhängige Zufallsvariablen x_i jeweils aus der gleichen Verteilung $p(x)$ mit Mittel $\langle x_i \rangle = \langle x \rangle$ und Varianz $\langle x_i^2 \rangle - \langle x_i \rangle^2 = \sigma_x^2$. Dann ist die neue Zufallsvariable $y = \sum_{i=1}^N x_i$ im Limes $N \rightarrow \infty$ gaußverteilt mit Mittelwert $\langle y \rangle = N\langle x \rangle$ und Varianz $\sigma_y^2 = N\sigma_x^2$.

Dies motiviert folgendes Vorgehen:

$$\begin{aligned} \text{Wir addieren einfach } N \text{ gleichverteilte Zufallsvariablen } x_i \text{ aus } [0, 1[\text{ und ziehen } N/2 \text{ ab:} \\ \text{Das Ergebnis } y = (\sum_{i=1}^N x_i) - N/2 \text{ ist dann } \textbf{gaußverteilt} \text{ mit } \langle y \rangle \approx 0 \text{ und } \sigma_y^2 = N/12. \end{aligned} \quad (10.12)$$

In der Praxis reichen typischerweise schon $N \sim 6$ Zahlen, um eine Gaußverteilung zu bekommen! Ein Artefakt sollte man bei dieser Methode aber im Auge behalten: Zahlen $y < -N/2$ oder $y > N/2$ kommen *niemals* vor.

10.2.3 Rückweisungsmethode

Die Rückweisungsmethode beruht darauf, dass Zufallszahlenpaare, d.h. Punkte (x, y) in zwei Dimensionen generiert werden, die gleichverteilt in der Fläche $A = \{(x, y) | x \in D \text{ und } 0 < y < p(x)\}$ unterhalb einer Verteilungsfunktion mit dem Definitionsbereich D liegen.

Die Verteilungsfunktion für solche Paare ist dann $p_A(x, y) = 1/|A| = \text{const}$ mit dem Flächeninhalt $|A| = \int_D dx p(x)$ (wenn $p(x)$ normiert ist, gilt $|A| = 1$). Die Wahrscheinlichkeit, dass einer dieser gleichverteilten Punkte eine Koordinate x hat, bekommt man dann nach den Regeln der Wahrscheinlichkeitstheorie durch Integration über alle Möglichkeiten für die zweite Koordinate y :

$$\int_0^{p(x)} dy p_A(x, y) = p(x)/|A| \stackrel{\text{normiert}}{=} p(x). \quad (10.13)$$

Wenn also gleichverteilt aus der Fläche A unterhalb der Funktion $p(x)$ gezogen werden kann, und der Flächeninhalt $|A|$ bekannt ist, folgt die x -Komponente dieser Punkte der normierten Verteilung $p(x)/|A|$.

Umgekehrt gilt auch, wenn x aus der normierten Verteilung $p(x)/|A|$ gezogen wird und dazu ein y gleichverteilt aus $[0, p(x)]$ gezogen wird, also gemäß der Verteilung $p_y(y) = \text{const} = 1/p(x)$, dann sind die Punkte (x, y) in der Fläche A unterhalb der Funktion $p(x)$ gleichverteilt. Die Verteilung $p_A(x, y)$ ist dann nämlich eine Produktverteilung mit

$$p_A(x, y) = \frac{p(x)}{|A|} p_y(y) = \frac{p(x)}{|A| p(x)} = \frac{1}{|A|}. \quad (10.14)$$

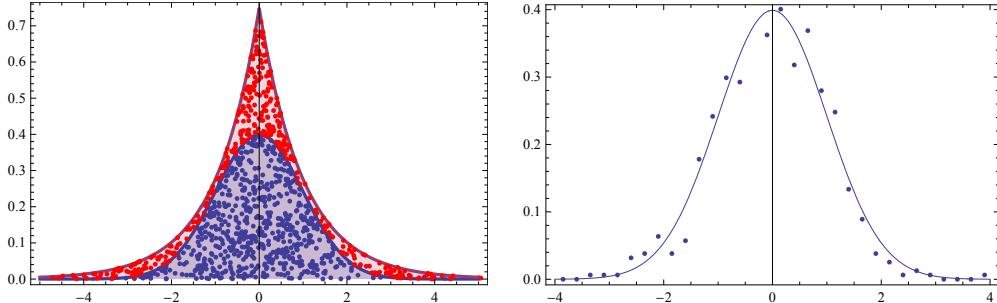


Abbildung 10.3: Rückweisungsverfahren zur Erzeugung von gaußverteilten Zufallszahlen mit $p(x) = (1/\sqrt{2\pi}) \exp(-x^2/2)$ mit Hilfe der exponentiellen Vergleichsverteilung $g(x) = (1/2) \exp(-|x|)$ und $k = 1.5$. Zufallszahlen x aus der Verteilung $g(x)$ können mit der Transformationsmethode erzeugt werden (aus in $]0, 1]$ gleichverteilten Zufallszahlen durch Anwendung von $f(x) = -\ln x$ und $f(x) = +\ln x$ in jeweils der Hälfte der Fälle, siehe oben). Links: Es wurden 1000 Punkte (x, y) gleichverteilt unter der roten Vergleichskurve $kg(x)$ generiert (mit Schritten 1 und 2). Die Rückweisungsmethode lehnt die roten Punkte mit $y > p(x)$ ab und akzeptiert nur die blauen Punkte mit $y < p(x)$ (Schritt 3). Rechts: Die Wahrscheinlichkeitsverteilung der x -Komponente der blauen Punkte stimmt mit $p(x)$ überein.

Bei der Rückweisungsmethode kennt man für eine zu erzeugende Wahrscheinlichkeitsverteilung $p(x) > 0$ (Normierung $\int dx p(x) = 1$) eine *Vergleichsverteilung* $g(x) > 0$, die ebenfalls normiert ist ($\int dx g(x) = 1$) und zu der es eine Konstante $k > 1$ gibt, so dass $kg(x)$ eine *obere Schranke* für $p(x)$ ist, d.h.

$$kg(x) > p(x) > 0 \quad \text{für alle } x$$

(weil beide Verteilungen $p(x)$ und $g(x)$ normiert sind, muss eine Konstante $k > 1$ verwendet werden, sonst kann die Schranke nicht für alle x gelten) und für die Zufallszahlen leicht generiert werden können. Beispielsweise bieten sich für $g(x)$ einfache kastenförmige Gleichverteilungen oder Exponentialverteilungen an, die mit der Transformationsmethode erzeugt werden können (siehe Gl. (10.8)).

Wir wollen nun zunächst zufällige Punkte (x, y) generieren, die gleichverteilt in der Fläche unterhalb der Funktion $kg(x)$ liegen. Dafür ziehen wir nach (10.14)

- 1) eine Zufallszahl x aus der Verteilung $g(x)$ und
- 2) eine zweite Zufallszahl y gleichverteilt in $[0, kg(x)]$.

Daraus können wir dann Punkte (x, y) generieren, die gleichverteilt in der Fläche unterhalb der zu erzeugenden Funktion $p(x)$ liegen, indem wir

- 3) nur Punkte (x, y) "akzeptieren", für die auch $y < p(x)$ gilt.
- 4) Die x -Komponente dieser Punkte ist dann nach (10.13) gemäß $p(x)$ verteilt.

Diese Schritte beschreiben die Rückweisungsmethode. Sie lässt sich oft einsetzen: Es wird lediglich eine Vergleichsverteilung $g(x)$ benötigt, die problemlos erzeugt werden kann und ein k , so dass $kg(x)$ eine obere Schranke darstellt. In Abb. 10.3 ist ein Beispiel zur Erzeugung gaußverteilter Zufallszahlen gezeigt.

10.3 Literaturverzeichnis Kapitel 10

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] D. P. Landau und K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2005.
- [4] W. Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Statistical, Computational, and Theoretical Physics. Oxford University Press, 2006.
- [5] M. Hjorth-Jensen. *Computational Physics (Skript)*. Oslo: University of Oslo, 2012.
- [6] L. Noll, R. Mende und S. Sisodiya. *Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system*. US Patent 5,732,138. März 1998.
- [7] G. Marsaglia. *Random numbers fall mainly in the planes*. Proce. Nat. Acad. Sci. U.S.A. **61** (1968), 25–28.

10.4 Übungen Kapitel 10

1. Linear kongruente Generatoren

Generieren Sie Pseudo-Zufallszahlen, indem Sie einen linear kongruenten Generator (10.1),

$$r_{n+1} = ar_n + c \pmod{m}$$

selbst implementieren.

a) Schreiben Sie ein Programm, um die ersten N Glieder ($N < m$) der Integer-Folge r_n abhängig von den 4 Parametern r_0 (seed), a , c und m zu generieren (verwenden Sie hierbei 64-Bit-Integer). Teilen Sie durch m um einen floating point Generator für Zufallszahlen in $[0, 1[$ zu bekommen.

b) Untersuchen Sie für die vier Parametersätze

- (i) $r_0 = 1234$, $a = 20$, $c = 120$, $m = 6075$
- (ii) $r_0 = 1234$, $a = 137$, $c = 187$, $m = 256$
- (iii) $r_0 = 123456789$, $a = 65539$, $c = 0$, $m = 2^{31} = 2147483648$
(RANDU Generator von IBM)
- (iv) $r_0 = 1234$, $a = 7^5 = 16807$, $c = 0$, $m = 2^{31} - 1$
(ran1() aus Num. Rec. 2. Ausgabe bzw. Matlab bis Version 4)

ihren floating point Generator zuerst auf Gleichverteilung, indem Sie für $N = 10^4$ Werte ein Histogramm erstellen, indem Sie das Intervall $[0, 1[$ in 10 bins der Länge 0.1 aufteilen.

c) Testen Sie die vier floating point Generatoren (i)–(iv) nun auf Korrelationen, indem Sie jeweils $N/2$ Paare (r_n, r_{n-1}) aus aufeinanderfolgenden Punkten in einem zweidimensionalen Quadrat $[0, 1]^2$ auftragen. Benutzen Sie bis zu $N = 10^5$ Werte (beachten Sie, dass nur $N < m$ Sinn macht).

2. Beliebige Verteilungen erzeugen

Ein Zufallszahlengenerator, der gleichverteilte Zahlen zwischen 0 und 1 erzeugt, kann auch eingesetzt werden, um beliebige Verteilungen zu erzeugen.

a) Benutzen Sie den Box–Muller–Algorithmus, um eine Gaußverteilung mit Varianz 1 und Mittelwert 0 zu erzeugen.

b) Benutzen Sie den zentralen Grenzwertsatz, um eine Gaußverteilung zu erzeugen. Bilden Sie dafür die Summe von N (geeignet wählen) gleichverteilten Zufallszahlen aus $[0, 1]$. Wie bekommt man eine Verteilung mit Mittelwert 0 und Standardabweichung 1? Welche Nachteile hat diese Methode, z.B. in Korrektheit und Effizienz?

c) Benutzen Sie das Rückweisungsverfahren, um die Verteilung $p_1(x) = \sin(x)/2$ in den Grenzen 0 bis π zu erzeugen.

d) Benutzen Sie die Transformationsmethode, um die Verteilung $p_2(x) = 3x^2$ in den Grenzen 0 bis 1 zu erzeugen.

Erzeugen Sie jeweils 10^4 Zufallszahlen und erstellen Sie ein Histogramm, wo die Häufigkeiten $p(x_i)$ der um x_i zentrierten Bins der Länge Δx gemäß $\sum_i p(x_i)\Delta x = 1$ normiert werden sollen. Plotten Sie auch die zugehörige normierte analytische Verteilung.

11 Monte-Carlo (MC) Simulation

Literatur zu diesem Teil:

neben MD die andere wichtige Simulationsmethode für klassische Vielteilchensysteme. Sehr zu empfehlen ist Frenkel [1], aber auch Landau und Binder [2] oder Krauth [3], Kinzel [4], Gould/Tobochnik [5], Koonin/Meredith [6] und Thijssen [7]. Für die Monte-Carlo Integration natürlich auch Numerical Recipes [8, 9].

Der Name aller **Monte-Carlo Methoden** stammt von der Assoziation mit dem Casino und damit mit Zufallszahlen. Die grundsätzliche Idee wird immer sein, Zufallszahlen zu benutzen, um Integrale, Mittelwerte, usw. zu berechnen, und zwar durch Mittelung über sogenannte “**Samples**” (**Stichproben**), die **zufällig** gezogen werden anstatt deterministisch vorzugehen. Dabei sollen die Samples aus bestimmten Wahrscheinlichkeitsverteilungen gezogen werden, typischerweise in hochdimensionalen Räumen. Daher ist dieses Kapitel eng verknüpft mit dem vorangehenden Kapitel 10.2, wo wir bereits einfache Methoden diskutiert haben, um Zufallszahlen (samples) x aus *ein-dimensionalen* Wahrscheinlichkeitsverteilungen $p(x)$ zu generieren. Monte-Carlo Methoden wie die Metropolis-Methode erlauben das samplen beliebiger Wahrscheinlichkeitsverteilungen $p(\vec{x})$ auch für hochdimensionale \vec{x} .

Bei der **MC-Integration** wird ein Integral als Mittelwert über eine Wahrscheinlichkeitsverteilung aufgefasst; durch samplen der Wahrscheinlichkeitsverteilung kann dieser Mittelwert und damit das Integral berechnet werden. Die Methode wird weitgehend unabhängig von der Dimension des Integrals funktionieren, also auch für hoch-dimensionale Probleme.

Bei der **MC-Simulation** wollen wir thermodynamischen Mittelwerte durch Mittelung über die Boltzmann-Verteilung berechnen. Dazu wollen wir Samples in Form von Boltzmann-verteilten Systemkonfigurationen generieren. Hier arbeitet man natürlicherweise im **kanonischen Ensemble** (im Gegensatz zur MD-Simulation, die natürlicherweise im mikrokanonischen Ensemble durchgeführt wird). Die ersten MC-Simulationen wurden von Nicholas Metropolis *et al.* an einem zweidimensionalen System harter Scheiben durchgeführt [10] (dem gleichen System, an dem auch die ersten MD-Simulationen von Alder und Wainwright durchgeführt wurden).



Abbildung 11.1: Links: Casino von Monte-Carlo in Monaco, Namensgeber der Monte-Carlo Methode, weil es hier (hoffentlich) zufällig zugeht. Rechts: Nicolas Metropolis (1915-1999), der Erfinder der Monte-Carlo Methode auf seinem Los Alamos Badge. (Quelle: Wikipedia).

11.1 Monte-Carlo Integration

Wir diskutieren die Monte-Carlo Integration mit Hilfe zufälliger Stützstellen und schätzen ihren Fehler ab, sowohl für eindimensionale als auch für mehrdimensionale Integrale. Dabei wird zuerst einfaches gleichverteiltes Sampling, dann Importance-Sampling erläutert.

Als erstes betrachten wir die einfache **Monte-Carlo Integration**, bei der wir integrieren möchten, indem wir **zufällige Stützstellen** „sampeln“ anstatt deterministisch Stützstellen zu generieren, mit den bereits in Kapitel 3.2 vorgestellten Methoden, die in der Regel alle äquidistante Stützstellen verwendeten. Wir schreiben dazu das Integral $\int d\vec{x}g(\vec{x}) = \int d\vec{x}p(\vec{x})f(\vec{x}) = \langle f \rangle_p$ als Mittelwert einer „Observablenfunktion“ $f(\vec{x})$ über eine Wahrscheinlichkeitsverteilung $p(\vec{x})$. Es wird verschiedene Möglichkeiten geben $p(\vec{x})$ (und damit $f(x)$) zu wählen (einfaches Sampling, Importance-Sampling), aber der Mittelwert wird immer durch sampeln von Stichproben (Stützstellen) \vec{x}_i aus der Verteilung $p(\vec{x})$ und Mittelwertbildung über diese Samples berechnet werden.

Wir machen dies zunächst an zwei Beispielen klar. Das erste Beispiel ist bereits eine mehrdimensionale Integration, und wir werden sehen, dass MC-Integration für **mehrdimensionale Integrale** genauso gut funktioniert und damit besonders geeignet ist.

11.1.1 Zwei Beispiele

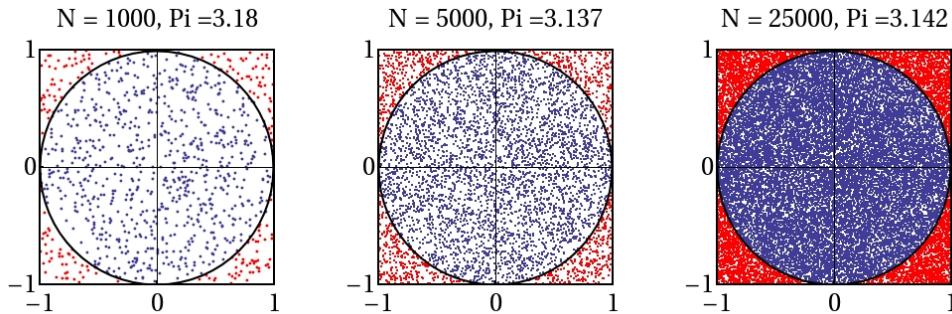


Abbildung 11.2: $\pi = 3.14159\dots$ wird durch Ziehen von N gleichverteilten Zufallspunkten in $[-1, 1]^2$ bestimmt. N_\circ blaue Punkte liegen im Einheitskreis, was auf $\pi \approx 4N_\circ/N$ führt. Die Abbildung zeigt das Ergebnis für $N = 1000, 5000, 25000$ Punkte.

Beispiel 1: Berechnung von π

Wir wollen π als Flächeninhalt des Einheitskreises berechnen:

- 1) Dazu werden wir **zufällig** N Punkte \vec{r}_i (Samples) **gleichverteilt** im Quadrat $[-1, 1]^2$ ziehen (einfach zu realisieren mit durch Ziehen zweier gleichverteilter Zufallszahlen in $[-1, 1]$).
- 2) Wir zählen ein Sample als „Erfolg“, wenn $\vec{r}_i^2 < 1$, also wenn der Punkt im Einheitskreis liegt. Die Wahrscheinlichkeit dafür ist durch das Verhältnis der Flächen von Einheitskreis und Quadrat gegeben,

$$p = \frac{A_\circ}{A_\square} = \frac{\pi}{4}. \quad (11.1)$$

- 3) Die Zahl der Erfolge N_\circ nach N -maligem Ziehen ist **binomialverteilt**. Die Wahrscheinlichkeit

für genau N_o Erfolge ist

$$p(N_o) = \binom{N}{N_o} p^{N_o} (1-p)^{N-N_o}.$$

Die mittlere Zahl von Erfolgen ist

$$\langle N_o \rangle = \sum_{N_o=0}^N N_o p(N_o) = Np = N \frac{\pi}{4}. \quad (11.2)$$

Die Streuung um diesen Mittelwert ist durch die Varianz $\sigma_{N_o}^2$ gegeben:

$$\sigma_{N_o}^2 = \langle (N_o - \langle N_o \rangle)^2 \rangle = Np(1-p). \quad (11.3)$$

- 4) Dies motiviert folgendes **Monte-Carlo Verfahren**, um $\pi = A_o$ zu messen (siehe Abb. 11.2):

Ziehe N Samples gleichverteilt aus dem Quadrat und messe die Zahl der Erfolge N_o , die im Kreis liegen. Für große N wird $N_o \approx \langle N_o \rangle$ mit dem Mittelwert aus (11.2) gelten (siehe nächster Absatz). Daher können wir π nach (11.2) als

$$\pi = A_o \approx \frac{N_o}{N} A_{\square} \quad (11.4)$$

bestimmen. Diese Formel sollte auch anschaulich einleuchten: Das Verhältnis der Treffer im Einheitskreis zur Gesamtzahl der Versuche im Quadrat, verhält sich wie die entsprechenden Flächeninhalte bei im Quadrat gleichverteilten Versuchen.

Der Fehler dabei ist durch die Wurzel der Varianz unserer π -Schätzung auf der rechten Seite, $\frac{N_o}{N} A_{\square}$, gegeben. Mit (11.3) für $\sigma_{N_o}^2$ erhalten wir:

$$\sigma_{A_o}^2 = \frac{A_{\square}^2}{N^2} \sigma_{N_o}^2 = \frac{1}{N} A_o (A_{\square} - A_o) \sim \frac{1}{N}. \quad (11.5)$$

Der **Fehler** ist also $\sim 1/\sqrt{N}$ und verschwindet mit großem N , was $N_o \approx \langle N_o \rangle$ und damit (11.4) nachträglich rechtfertigt. Anhand der Fehlerabschätzung (11.5) sehen wir auch, dass für größere Quadrate als $A_{\square} = 4$ um den Einheitskreis das Verfahren natürlich auch funktionieren würde mit Formel (11.4) für π , dass aber der Fehler $\sim (A_{\square} - A_o)^{1/2}$ dann auch größer wäre.

Beispiel 2: Integral $I = \int_a^b dx g(x)$

In Kapitel 3.2 haben wir diverse Verfahren mit *deterministisch* ausgewählten (äquidistanten) Stützstellen kennengelernt (Trapezregel, Simpsonregel, ...), um ein einfaches Integral $I = \int_a^b dx g(x)$ numerisch zu berechnen. Hier wollen wir nun N **zufällige** Stützstellen x_i (Samples) **gleichverteilt** aus dem Intervall $[a, b]$ ziehen. Der mittlere Abstand zwischen den Stützstellen ist dann $(b-a)/N$. Daher erscheint folgende Formel zu Berechnung des Integrals plausibel:

$$I_{MC} = \frac{b-a}{N} \sum_{i=1}^N g(x_i). \quad (11.6)$$

Dies ist die einfachste **Monte-Carlo Integrationsformel**.

11.1.2 Einfaches Sampling

Wir wollen die Formel (11.6) nun systematisch herleiten und untersuchen. Unsere Stützstellen x_i sind Stichproben (Samples) aus einer Gleichverteilung

$$p(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{sonst} \end{cases}. \quad (11.7)$$

Mit dieser Verteilung $p(x)$ gilt

$$I = \int_a^b dx g(x) = \int dx p(x) \underbrace{(b-a)g(x)}_{\equiv f(x)} = \langle f \rangle_p. \quad (11.8)$$

Also kann I als **Mittelwert** der Funktion $f(x)$ bezüglich der Wahrscheinlichkeitsdichte $p(x)$ geschrieben werden.

Einen Mittelwert approximiert man nach dem Gesetz der großen Zahlen durch häufiges (N -faches) Ziehen von **Stichproben (Samples)**:

$$\langle f \rangle \approx \frac{1}{N} \sum_{i=1}^N f(x_i) \stackrel{(11.8)}{=} \frac{b-a}{N} \sum_{i=1}^N g(x_i) = I_{MC} \quad (11.9)$$

wie in (11.6). Den **Fehler** bei dieser Approximation des Integrals kann man wieder über die Varianz der Zufallsgröße $I_{MC} = \frac{1}{N} \sum_{i=1}^N f(x_i)$ abschätzen, ähnlich wie bei der π -Bestimmung (Beispiel 1).

Dabei wissen wir allerdings nur, dass die x_i einer Verteilung $p(x)$ folgen. Wir wissen nicht *a priori*, wie $f(x_i)$ verteilt ist, geschweige denn die Summe in I . Hier hilft der **zentrale Grenzwertsatz**, der genau eine Aussage macht über die neue Zufallsvariable $y \equiv \sum_{i=1}^N f(x_i)$, wenn diese die **Summe aus N unabhängigen, identisch verteilten Zufallsvariablen $f(x_i)$** ist. Mit deren Mittelwert und Varianz

$$\begin{aligned} \langle f(x_i) \rangle &= \int dx p(x) f(x) = \langle f \rangle \\ \sigma_f^2 &= \langle (f - \langle f \rangle)^2 \rangle = \langle f^2 \rangle - \langle f \rangle^2 = \int dx p(x) f^2(x) - \langle f \rangle^2 \end{aligned}$$

besagt der zentrale Grenzwertsatz, dass auch y **gaußverteilt** ist und im Limes großer N Mittelwert und Varianz

$$\langle y \rangle = N \langle f \rangle \quad \text{und} \quad \sigma_y^2 = N \sigma_f^2 \quad (11.10)$$

beträgen.

Demnach ist die Varianz von $I_{MC} = y/N$

$$\sigma_I^2 = \frac{1}{N^2} \sigma_y^2 = \frac{1}{N} \sigma_f^2 = \frac{1}{N} (\langle f^2 \rangle - \langle f \rangle^2) \sim \frac{1}{N}. \quad (11.11)$$

Wir finden also einen **Fehler** $\sim 1/\sqrt{N}$ bei der MC-Integration (genau wie bei dem Beispiel der π -Bestimmung).

Wir bemerken, dass dieser Fehler zunächst einmal *schlechter* ist als beispielsweise bei der einfachen Trapezregel (mit gleicher Zahl N von Stützstellen war dort $h \sim 1/N$ und der Fehler $\sim Nh^3 \sim N^{-2}$) oder gar bei der Simpsonregel (Fehler $\sim Nh^5 \sim N^{-4}$). Dies wird sich aber in *höheren Dimensionen* ändern.

Ein großer Vorteil der Monte-Carlo Integration ist, dass sie auch problemlos für **mehrdimensionale Integrale** in n Raumdimensionen funktioniert. Dazu wird auch das n -dimensionale Volumenintegral $I = \int_V d^n \vec{r} g(\vec{r})$ über ein Integrationsvolumen V als **Mittelwert** aufgefasst:

$$I = \int_V d^n \vec{r} g(\vec{r}) = \int d^n \vec{r} p(\vec{r}) f(\vec{r}) = \langle f \rangle \quad (11.12)$$

mit einer Funktion $f(\vec{r})$ und einer Wahrscheinlichkeitsverteilung $p(\vec{r})$ die so gewählt werden, dass

$$p(\vec{r}) f(\vec{r}) = \begin{cases} g(\vec{r}) & \vec{r} \in V \\ 0 & \text{sonst} \end{cases}. \quad (11.13)$$

Genau wie in einer Raumdimension approximieren wir $\langle f \rangle$, indem wir N gemäß der Wahrscheinlichkeitsverteilung $p(\vec{r})$ verteilte Samples \vec{r}_i ziehen:

$$\boxed{\langle f \rangle \approx \frac{1}{N} \sum_{i=1}^N f(\vec{r}_i) = I_{MC}} \quad (11.14)$$

Der Fehler folgt auch wieder genau wie in einer Raumdimension über den zentralen Grenzwertsatz aus der Varianz der Zufallsgröße $I_{MC} = \frac{1}{N} \sum_{i=1}^N f(\vec{r}_i)$:

$$\boxed{\sigma_I^2 = \frac{1}{N} \sigma_f^2 = \frac{1}{N} (\langle f^2 \rangle - \langle f \rangle^2) \sim \frac{1}{N}}. \quad (11.15)$$

Genau wie in einer Raumdimension finden wir einen **Fehler** $\sim 1/\sqrt{N}$.

Der Fehler $\sim 1/\sqrt{N}$ (11.15) bei der MC-Integration ist insbesondere **unabhängig von der Dimension n** des Volumenintegrals! Wir wollen erneut mit Trapez- oder Simpsonregel aus Kapitel 3.2 vergleichen. Dort hätte man N äquidistante Stützstellen im n -dimensionalen Raum verteilt. Das heißt, das Integrationsvolumen pro Stützstelle ist V/N und die Kantenlänge dieses würfelförmigen Volumens $h \sim (V/N)^{1/n}$. Die **Trapezregel** ergibt in n Raumdimensionen in jedem Würfel analog zu (3.6) einen Fehler $h^n h^2$ (“Intervallvolumen” h^n mal Fehler h^2 im Integranden). Über alle N Würfel ergibt sich ein **Fehler** $\mathcal{O}(Nh^{2+n}) \sim N^{1-(2+n)/n} \sim N^{-2/n}$. Eine analoge Argumentation mit der **Simpsonregel** gibt einen **Fehler** $\mathcal{O}(Nh^{4+n}) \sim N^{1-(4+n)/n} \sim N^{-4/n}$. Wir sehen, dass hier die Fehler mit der Anzahl n der Raumdimensionen anwachsen!

Daher wird der Fehler der MC-Integration in hohen Raumdimensionen irgendwann kleiner als bei Trapez- oder Simpsonregel. Der MC-Fehler wird kleiner als bei der Trapezregel für Dimensionen $n > 4$ und er wird kleiner als bei der Simpsonregel für $n > 8$.¹ In der statistischen Physik entsprechen die Zustandssummen und Mittelwerte eines kontinuierlichen Systems (wie dem Lennard-Jones Fluid) nach Abspaltung der Impulsintegration Konfigurationsintegralen mit $n \sim 3$ mal Teilchenzahl, was selbst in einer Simulation mit “nur” 100 Teilchen immer noch dazu führt, dass die MC-Methode hier deterministischen Methoden haushoch überlegen ist.

Die **MC-Integration mehrdimensionaler Integrale** ist den deterministischen Methoden auch praktisch, d.h. vom Programmieraufwand her, überlegen bei kompliziert geformten Integrationsvolumina V . Um sie numerisch zu implementieren brauchen wir lediglich eine Vorschrift, um Samples mit einer geeigneten Verteilung $p(\vec{r})$ und Funktion $f(\vec{r})$ nach (11.13) zu generieren. Dies kann praktisch beispielsweise wieder wie bei der π -Berechnung implementiert werden: Wir setzen unser Integrationsvolumen V in ein größeres, einfaches, quaderförmiges Volumen V_\square bekannten Volumeninhalts $\text{Vol}(V_\square)$, das V vollständig umfasst. Dann wählen wir

$$f(\vec{r}) = \begin{cases} \text{Vol}(V_\square) g(\vec{r}) & \vec{r} \in V \\ 0 & \text{sonst} \end{cases}. \quad (11.16)$$

¹ Der Vergleich der Fehler ist etwas “unfair”, da die Fehler $\sim N^{-2/n}$ und $\sim N^{-4/n}$ bei Trapez- bzw. Simpsonregel eine “worst case” Abschätzung darstellen, während der Fehler $\sim N^{-1/2}$ eher ein “typischer” Fehler ist, der sich aus der Varianz ergibt.

Außerdem ziehen wir Samples \vec{r}_i gleichverteilt aus V_{\square} , d.h.

$$p(\vec{r}) = \begin{cases} \frac{1}{\text{Vol}(V_{\square})} & \vec{r} \in V_{\square} \\ 0 & \text{sonst} \end{cases}. \quad (11.17)$$

Dies führt dann insgesamt nach (11.14) zu der einfachen Vorschrift

$$I_{MC} = \frac{\text{Vol}(V_{\square})}{N} \sum_{\vec{r}_i \in V} g(\vec{r}_i)$$

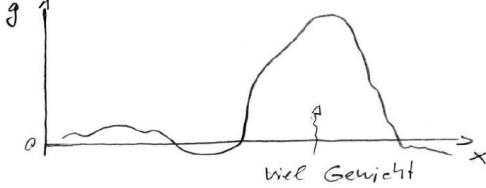
(11.18)

zur MC-Berechnung von $I = \int_V d^n \vec{r} g(\vec{r})$, wobei die \vec{r}_i gleichverteilt aus V_{\square} gezogen werden. Bei einem kompliziert geformten Integrationsvolumen V brauchen wir keine vollständige Parametrisierung dieses Volumens wie bei den deterministischen Verfahren, sondern lediglich einen einfachen Algorithmus, der es erlaubt zu entscheiden, ob ein Sample innerhalb oder außerhalb von V liegt. Wir sehen auch, dass sich (11.18) in zwei Raumdimensionen und mit $g(\vec{r}) = 1$ und $V = \text{Einheitskreis}$ auch gerade wieder auf die Formel (11.4) aus dem ersten Beispiel der π -Berechnung reduziert. Die Vorschrift (11.18) ist die mehrdimensionale Verallgemeinerung von (11.6) und dementsprechend analog aufgebaut: $\text{Vol}(V_{\square})/N$ ist hier das mittlere Volumen pro Stützstelle, und es wird über zufällige Stützstellen in V summiert.

11.1.3 Importance-Sampling

Bisher haben wir bei der Wahl von $p(\vec{r})$ in (11.13) immer (stückweise) **konstante** $p(\vec{r})$ verwendet, siehe (11.7) oder (11.17), d.h. die Samples waren **gleichverteilt** im Integrationsvolumen V selbst oder in einem V umgebenden einfachen Hilfsvolumen (wie V_{\square} in (11.17)). Dieses Sampling bezeichnet man als **einfaches Sampling**. Man kann aber prinzipiell **beliebige** Verteilungen $p(\vec{r})$ wählen, z.B. um den Fehler der MC-Integration zu verbessern. Dies bezeichnet man als **Importance-Sampling**.

Wenn wir ein $p(\vec{r})$ wählen können, sollten wir versuchen $p(\vec{r})$ dort viel Gewicht zu geben, wo auch der Integrand $g(\vec{r})$ betragsmäßig groß ist, da wir diese Regionen heuristisch als "wichtig" für die Integration einschätzen.



Offensichtlich müsste $p(\vec{r})$ dann ähnlich aussehen, wie $|g(\vec{r})|$ selbst; dabei muss $p(\vec{r})$ allerdings normiert sein. Tatsächlich stellt sich als ideale Wahl

$$p(\vec{r}) = \begin{cases} c|g(\vec{r})| & \vec{r} \in V \\ 0 & \text{sonst} \end{cases}$$

mit $\frac{1}{c} = \int_V d^n \vec{r} |g(\vec{r})|$

(11.19)

heraus, so dass $f(\vec{r}) = g(\vec{r})/p(\vec{r}) = \text{const}$ nach (11.13), wenn $g(\vec{r})$ sein Vorzeichen nicht ändert. Warum ist diese Wahl ideal?

Importance-Sampling soll den **Fehler** der MC-Integration **verkleinern**: Der Fehler von I_{MC} in (11.14) war gegeben durch die Wurzel der Varianz (11.15):

$$\sigma_I^2 = \frac{1}{N} \sigma_f^2 = \frac{1}{N} (\langle f^2 \rangle - \langle f \rangle^2) \sim \frac{1}{N}.$$

Der Fehler war $\sim 1/\sqrt{N}$, wobei beim Importance-Sampling allerdings der Vorfaktor dieses Skalenverhaltens wichtig wird:

$$\sigma_f^2 = \left\langle \frac{g^2}{p^2} \right\rangle - \left\langle \frac{g}{p} \right\rangle^2. \quad (11.20)$$

Dieser Vorfaktor $\sigma_f^2 \geq 0$ wird minimal, d.h. im Prinzip $\sigma_f = 0$, für $f = \text{const}$: Wenn der Integrand g sein Vorzeichen nicht ändert (was immer zu erreichen ist durch die Addition einer Konstanten), wird mit der Wahl (11.19) der Fehler also tatsächlich auf $\sigma_f = 0$ gedrückt.

Dies sollte praktisch natürlich nicht möglich sein; das Problem bei der Argumentation ist die Bestimmung der Normierung c in (11.19). Diese Normierung $\frac{1}{c} = \int_V d^n \vec{r} |g(\vec{r})| = |I|$ kann in (11.19) nicht bekannt sein, da I ja gerade zu berechnen ist. Daher ist die ideale Wahl von c in (11.19) praktisch unmöglich. Ein anderes Problem kann darin liegen, ein Verfahren zu implementieren, um mit $p(\vec{r})$ verteilte Samples zu ziehen. Wir haben z.B. in Kapitel 10.2 zu Zufallszahlen gesehen, dass es oft praktisch nicht-trivial ist, aus bestimmten Verteilungen direkt zu sampeln, beispielsweise war es bereits nicht-trivial, gaußverteilte Zufallszahlen zu generieren.

Ein **Beispiel**, wo Importance-Sampling sich als nützlich erweist, ist ein Integral

$$I = \int_0^1 dx g(x) \quad \text{mit } g(x) \sim x^{-1+\varepsilon} \quad (\varepsilon > 0) \quad \text{für } x \approx 0.$$

Für $\varepsilon < 1/2$ ist $\langle g^2 \rangle = \infty$ und bei einfaches Sampling (mit $p(x) = 1$) divergiert die Varianz $\sigma_f^2 = \langle g^2 \rangle - \langle g \rangle^2 = \infty$ und damit der Fehler der MC-Integration. Dieser ‘katastrophale Fehler’ lässt sich mit Importance-Sampling vermeiden, indem $p(x) = \varepsilon x^{\varepsilon+1}$ gewählt wird. Dann wird

$$f(x) = \frac{g(x)}{p(x)} \sim x^0 \quad \text{bei } x \approx 0.$$

Bei dieser Wahl von $p(x)$ wird der wichtige Bereich um $x \approx 0$ sehr häufig gesampelt, wodurch der Fehler letztlich endlich bleibt. Importance-Sampling mit einer Verteilung $p(x) = \varepsilon x^{\varepsilon+1}$ lässt sich leicht implementieren mit Hilfe der Transformationsmethode, siehe Gl. (10.9).

Oft ist es aber gar nicht so einfach, eine direktes Sampling aus einer Verteilung $p(\vec{r})$ praktisch zu realisieren, um ein Imprtance-Sampling vorzunehmen. In den folgenden Kapiteln 11.2 und 11.3 werden wir sehen, wie wir das Importance-Sampling praktisch erst wirklich nutzbar machen können, wenn wir die Verteilung $p(\vec{r})$ gar nicht **direkt** sampeln, sondern über einen Markov-Prozess erzeugen. Dann wird es aber ein sehr mächtiges Werkzeug, um z.B. statistische Physik durch Sampling mit der Boltzmann-Verteilung zu betreiben.

11.2 Markov-Sampling, Metropolis-Algorithmus

Wir führen mit dem Markov-Sampling eine neue Sampling-Methode ein, die auf stochastischen Markov-Prozessen beruht. Wir diskutieren stationäre Wahrscheinlichkeitsverteilungen, das detaillierte Gleichgewicht (detailed balance) eines Markov-Prozesses und den Metropolis-Algorithmus zur Erzeugung einer gegebenen Wahrscheinlichkeitsverteilung.

Bisher hatten wir bei der MC-Integration **direktes Sampling** einer Verteilung $p(\vec{r})$ angewendet, d.h. die Stichproben/Samples \vec{r}_i sollten mit Hilfe von Zufallszahlen direkt mit der gewünschten Verteilung $p(\vec{r})$ generiert werden. Wir hatten aber bei der Diskussion des Importance-Sampling auch eingesehen, dass es schwierig ist, eine korrekt normiert und beliebige Wahrscheinlichkeitsverteilung $p(\vec{r})$ direkt zu sampeln.

Eine alternative Möglichkeit besteht darin, die Samples \vec{r}_i durch einen **dynamischen Zufallsprozess**, genauer einen **Markov-Prozess** zu generieren. Wir beginnen wieder mit einem bereits bekannten Beispiel.

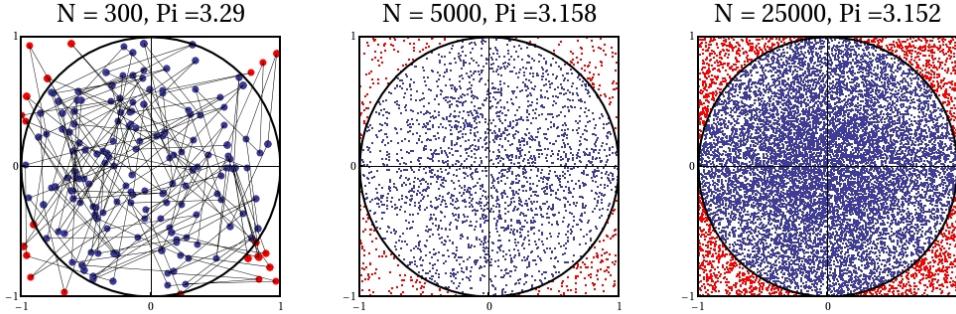


Abbildung 11.3: $\pi = 3.14159\dots$ wird durch einen Random Walk mit N Schritten der Länge $|\Delta\vec{r}| = 0.75$ und zufälliger Richtung in $[-1, 1]^2$ bestimmt. N_0 blaue Punkte liegen im Einheitskreis, was auf $\pi \approx 4N_0/N$ führt. Die Abbildung zeigt das Ergebnis für π für $N = 300, 5000, 25000$ Punkte. Links ist die Spur des Random Walk explizit gezeigt.

Beispiel: Berechnung von π

Nun wollen wir gleichverteilte Samples $\vec{r}_i \in [-1, 1]^2$ durch einen **Random Walk** generieren:

1a) Wir wählen einen Schritt $\Delta\vec{r}$ mit fester Länge $|\Delta\vec{r}| < 1$ und zufälliger Richtung.

1b)

$$\vec{r}_{i+1} = \begin{cases} \vec{r}_i + \Delta\vec{r} & \text{wenn } \vec{r}_i + \Delta\vec{r} \in [-1, 1]^2 \\ \vec{r}_i & \text{sonst} \end{cases}.$$

Dann wieder 1a) usw.

Die so generierten Samples \vec{r}_i füllen $[-1, 1]^2$ **gleichmäßig** aus (siehe Abb. 11.3). Die übrigen Schritte 2)-4) der MC- π -Berechnung verlaufen wie in Kapitel 11.1.

Ein Random Walk, wie er hier benutzt wird, um in $[-1, 1]^2$ gleichverteilte Samples zu generieren ist ein einfaches Beispiel für einen **Markov-Prozess**.

11.2.1 Markov-Prozesse, Master-Gleichung

Ein Markov-Prozess wird beschrieben durch:

- **Zustände** eines Teilchens oder Systems. Diese können entweder **diskrete Zustände** i sein oder **kontinuierlich** verteilte Zustände \vec{r} .
- Zwischen den Zuständen finden **stochastische Übergänge** statt, für die wir Wahrscheinlichkeiten angeben. Dabei unterscheiden wir zwei Möglichkeiten:
 - a) Eine **diskrete Zeit** $t = n\Delta t$, dann spricht man auch oft von einer **Markov-Kette**.
Die Übergänge sind durch **Übergangswahrscheinlichkeiten** M_{ij} von Zustand i nach j oder $M(\vec{r}, \vec{r}')$ von \vec{r} nach \vec{r}' charakterisiert.
 - b) Eine **kontinuierliche Zeit** t , dann spricht man von einem eigentlichen **Markov-Prozess**.
Die Übergänge sind durch eine **Übergangsrate** k_{ij} oder $k(\vec{r}, \vec{r}')$ charakterisiert, die die Bedeutung einer **Übergangswahrscheinlichkeit pro Zeiteinheit** hat.

- Die **Markov-Eigenschaft**: Die Übergangswahrscheinlichkeiten hängen nur vom Anfangszustand (i, t) und vom Endzustand $(j, t + \Delta t)$ ab (bei diskreter Zeit) und **nicht** von der “Vorgeschichte”, wie der Zustand i erreicht wurde.

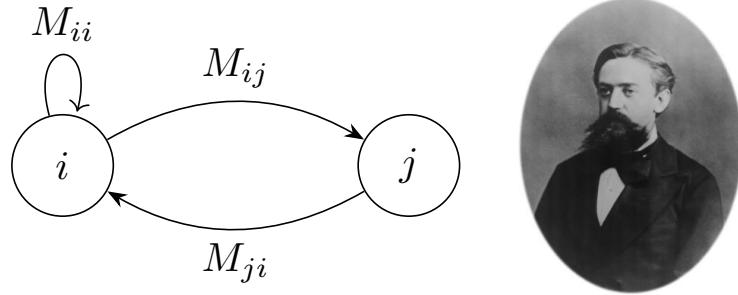


Abbildung 11.4: Links: Veranschaulichung der Markov-Übergänge zwischen Zuständen i und j . Rechts: Andrey Andreyevich Markov (1856-1922), russischer Mathematiker. (Quelle: Wikipedia).

Als Ergebnis des stochastischen Markov-Prozesses stellt sich eine **Aufenthaltswahrscheinlichkeit** $p_i(t)$ bzw. $p(\vec{r}, t)dV$ ein; dies ist die Wahrscheinlichkeit, dass System zur Zeit t im Zustand i bzw. in einem \vec{r} -Volumen dV um Zustand \vec{r} zu finden. Diese zeitabhängigen Wahrscheinlichkeitsverteilungen erfüllen **Bewegungsgleichungen**. Die diskrete Wahrscheinlichkeitsverteilung $p_i(t)$ erfüllt eine sogenannte **Master-Gleichung**, eine kontinuierliche Wahrscheinlichkeitsverteilung erfüllt eine **Fokker-Planck-Gleichung**. Der kontinuierliche Fall wird später im Kapitel 13.3 über stochastische Bewegungsgleichungen besprochen werden. Wir fokussieren uns hier auf den diskreten Zustandsraum und betrachten speziell diskrete Zustände i und eine diskrete Zeit $t = n\Delta t$ (im Computer muss ja ohnehin alles diskretisiert werden, wie wir bereits wissen) und leiten für diesen Fall die **Master-Gleichung** für die Wahrscheinlichkeitsverteilung $p_i(t)$ her.

Aus der Markov-Eigenschaft folgt zunächst, dass die Wahrscheinlichkeiten $p_j(t + \Delta t)$ nur von den $p_i(t)$ einen Schritt vorher und den Übergangswahrscheinlichkeiten M_{ij} abhängen können. Der Zustand j zur Zeit $t + \Delta t$ muss durch einen Übergang aus einem Zustand i zur Zeit t erreicht worden sein und $p_i(t)M_{ij}$ ist die Wahrscheinlichkeit in Zustand i zur Zeit t zu starten und in Zustand j zur Zeit $t + \Delta t$ zu enden. Dann ist die Wahrscheinlichkeit $p_j(t + \Delta t)$ in j zur Zeit $t + \Delta t$ zu sein durch Summation über alle möglichen Ausgangszustände i gegeben:

$$\boxed{\begin{aligned} p_j(t + \Delta t) &= \sum_i p_i(t)M_{ij} \\ \iff \vec{p}^t(t + \Delta t) &= \vec{p}^t(t) \cdot \underline{\underline{M}} \\ \iff \vec{p}^t(t + n\Delta t) &= \vec{p}^t(t) \cdot \underline{\underline{M}}^n. \end{aligned}} \quad (11.21)$$

Dies ist bereits die **Master-Gleichung** für die Zeitentwicklung des Wahrscheinlichkeitsvektors $\vec{p}(t)$ (= Spaltenvektor, $\vec{p}^t(t)$ = Zeilenvektor), wobei $\underline{\underline{M}}$ die **Übergangsmatrix** mit den Matrixelementen M_{ij} ist. Sie hat folgende **Eigenschaften**:

- $0 \leq M_{ij} \leq 1$, die M_{ij} sind Wahrscheinlichkeiten.
- $\sum_j M_{ij} = 1$, da die Normierung $\sum_i p_i = 1$ in jedem Zeitschritt erhalten bleiben muss.

Matrizen mit den Eigenschaften (i) und (ii) heißen auch **stochastische Matrizen**.²

²Die Definition in (ii) mit Summation über den zweiten Index unterscheidet sich von der Definition (8.30) in Kapitel 8.3.3 wo bei der Normierung über den ersten Index summiert wurde. Dies liegt daran, dass die Master-gleichung

Wir können die Master-Gleichung (11.21) auch anders schreiben, wenn wir an der Änderung von $P_i(t)$ interessiert sind:

$$\begin{aligned} p_i(t + \Delta t) - p_i(t) &= \sum_j M_{ji} p_j(t) - \underbrace{\sum_j M_{ij} p_i(t)}_{=1} \\ &= \sum_j \left[\underbrace{M_{ji} p_j(t)}_{\text{Gewinn aus } j \rightarrow i} - \underbrace{M_{ij} p_i(t)}_{\text{Verlust aus } i \rightarrow j} \right]. \end{aligned} \quad (11.22)$$

In dieser Form wird die Master-Gleichung auch **Ratengleichung** genannt. Der **Gesamtstrom** von Zustand i nach Zustand j pro Zeit Δt ist

$$J_{ij} = \frac{1}{\Delta t} (-M_{ji} p_j + M_{ij} p_i) = -J_{ji}. \quad (11.23)$$

Damit kann man die Ratengleichung (11.22) auch als **Kontinuitätsgleichung** schreiben:

$$\frac{p_i(t + \Delta t) - p_i(t)}{\Delta t} = \partial_t p_i = - \sum_{j(\neq i)} J_{ij}. \quad (11.24)$$

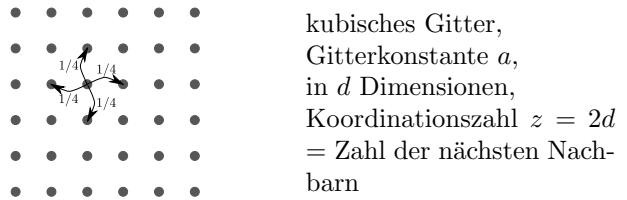
Als **Beispiel** betrachten wir einen **Random Walk** bzw. **Diffusion** auf einem Gitter. Dieser stochastische Prozess wird als ein stochastischer **Hüpfprozess** beschrieben:

In jedem Zeitschritt hüpfst ein Teilchen auf einen zufällig ausgewählten nächsten Nachbarplatz. Alle nächsten Nachbarn seien gleich wahrscheinlich:

$$M_{ij} = \begin{cases} \frac{1}{z} & \text{für } z \text{ nächste Nachbarn} \\ 0 & \text{sonst} \end{cases}.$$

Im Spezialfall $d = 1$ gilt

$$M_{ij} = \frac{1}{2} (\delta_{j,i+1} + \delta_{j,i-1}).$$



Dieser Hüpfprozess definiert einen Markov-Prozess. Die zugehörige Master-Gleichung in $d = 1$ lautet

$$p_i(t + \Delta t) = \frac{1}{2} p_{i-1}(t) + \frac{1}{2} p_{i+1}(t). \quad (11.25)$$

Die Ratengleichung lautet

$$\begin{aligned} p_i(t + \Delta t) - p_i(t) &= \frac{1}{2} p_{i-1}(t) - p_i(t) + \frac{1}{2} p_{i+1}(t) \\ \Delta t \rightarrow 0 : \quad \partial_t p_i(t) &= \frac{1}{2\Delta t} \underbrace{(p_{i-1}(t) - 2p_i(t) + p_{i+1}(t))}_{\approx a^2 \partial_x^2 p(x,t)}. \end{aligned}$$

Im Kontinuumslimes $a \rightarrow 0$, $x = ia$ wird daraus die eindimensionale **Diffusionsgleichung**

$$\partial_t p(x, t) = \frac{a^2}{2\Delta t} \partial_x^2 p(x, t) = D \partial_x^2 p(x, t) \quad (11.26)$$

in (11.21) hier mit Zeilenvektoren formuliert ist. Die entsprechende auch als Master-Gleichung interpretierbare Iteration (8.29) der Google-Matrix war mit Spaltenvektoren formuliert.

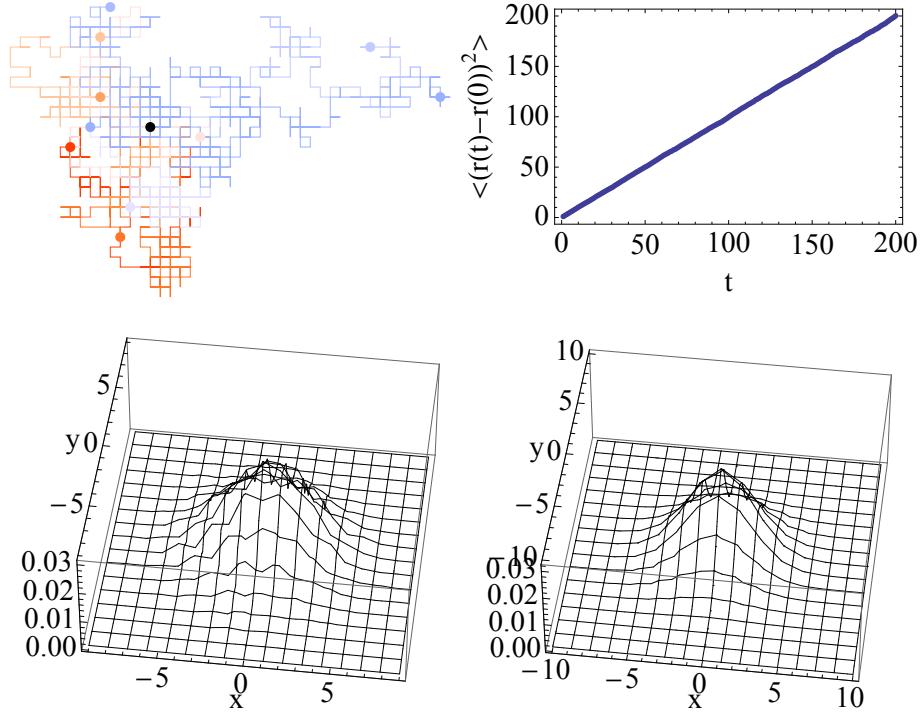


Abbildung 11.5: Random Walks auf einem zweidimensionalen Quadratgitter ($z = 2d = 4$, Gitterkonstante $a = 1$). Oben links: 10 verschiedene Random Walks (in verschiedenen Farben), die alle bei $\vec{r} = (0, 0)$ (schwarzer Punkt) starten und bei den entsprechenden farbigen Punkten enden. Oben rechts: Wir mitteln den quadratischen Abstand vom Startpunkt nach Zeit $t = n\Delta t$, $\langle(\vec{r}(t) - \vec{r}(0))^2\rangle$ (hier haben wir für $\langle\dots\rangle$ über 10000 Random Walks mit je $n \leq 200$ Schritten gemittelt). Wir finden das Diffusionsgesetz $\langle(\vec{r}(t) - \vec{r}(0))^2\rangle = 4Dt$ mit $D = a^2/4\Delta t$. Unten: Vergleich der Verteilung der Endpunkte der 10000 Random Walks nach $t = 10\Delta t$ Schritten (links) mit folgt der analytischen Lösung (11.28) der Diffusionsgleichung (11.26) für $d = 2$ und mit $D = a^2/4\Delta t$ für $t = 10\Delta t$ (rechts).

mit der **Diffusionskonstanten** $D = \frac{a^2}{2\Delta t}$. In höheren Dimensionen (Hüpfen mit Wahrscheinlichkeit $1/z$ zu nächsten Nachbarn, z.B. auf kubischem Gitter $z = 2d$) verallgemeinert sich dies zu

$$\partial_t p(\vec{r}, t) = D \vec{\nabla}^2 p(x, t) \quad (11.27)$$

mit $D = \frac{a^2}{z\Delta t}$. Die Lösung dieser Gleichung zur Anfangsbedingung $p(\vec{r}, 0) = \delta(\vec{r})$ eines ursprünglich bei $\vec{r} = 0$ lokalisierten Teilchens lautet (Lösung durch Fouriertransformation, siehe Physik III)

$$p(\vec{r}, t) = \left(\frac{1}{4\pi Dt} \right)^{d/2} e^{-\vec{r}^2/4Dt}. \quad (11.28)$$

Mit Hilfe dieser Wahrscheinlichkeitsverteilung $p(\vec{r}, t)$ können wir Mittelwerte berechnen, z.B. den mittleren quadratischen Abstand vom Startpunkt:

$$\langle(\vec{r}(t) - \vec{r}(0))^2\rangle = \int d^d \vec{r} (\vec{r}(t) - \vec{r}(0))^2 p(\vec{r}, t) = 2dDt. \quad (11.29)$$

Dies ist das **Diffusionsgesetz** mit der charakteristischen linearen Zeitabhängigkeit $\langle(\vec{r}(t) - \vec{r}(0))^2\rangle \propto t$. Der Zusammenhang zwischen dem Vorfaktor und der Diffusionskonstanten kann auch als alternative Definition der Diffusionskonstanten D verwendet werden.

Die Mittelung $\langle \dots \rangle$ mit der Lösung $p(\vec{r}, t)$ einer Mastergleichung (hier die Diffusionsgleichung) ist äquivalent zu einer Mittelung über sehr viele Realisationen des entsprechenden Markov-Prozesses (hier Mittelung über viele Random Walks). Wir können $\langle(\vec{r}(t) - \vec{r}(0))^2\rangle$ also auch berechnen, indem wir z.B. 10000 Random Walks generieren und den mittleren quadratischen Abstand vom Startpunkt zur Zeit t über alle diese Random Walks mitteln. So wurde in Abb. (11.5) (Mitte) verfahren. Auch $p(\vec{r}, t)$ selbst kann als Mittelwert $p(\vec{r}, t) = \langle \delta(\vec{r}(t) - \vec{r}) \rangle$ geschrieben werden und damit durch Mittelung über viele Realisationen des Markov-Prozess gewonnen werden. Bei der Diffusion können wir $p(\vec{r}, t)$ damit als Aufenthaltswahrscheinlichkeit der Random Walks zur Zeit t am Ort \vec{r} (=Aufenthaltshäufigkeit der Endpunkte zur Zeit t bei \vec{r} geteilt durch Zahl der Random Walks) berechnen, siehe Abb. (11.5) rechts.

11.2.2 Detailed Balance

Nun suchen wir **stationäre Zustände** der Master-Gleichung (11.21), d.h. Wahrscheinlichkeitsvektoren \vec{p}_{st} , die sich nicht mehr ändern unter der Mastergleichung. Ein stationärer Zustand \vec{p}_{st} muss daher

$$\vec{p}_{st}^t(t + \Delta t) = \vec{p}_{st}^t(t) \cdot \underline{\underline{M}} = \vec{p}_{st}^t(t)$$

erfüllen, d.h.³

\vec{p}_{st}^t ist Links-Eigenvektor von $\underline{\underline{M}}$, bzw.

\vec{p}_{st}^t ist Rechts-Eigenvektor von $\underline{\underline{M}}^t$ zum Eigenwert 1.

(11.30)

Für die Ströme J_{ij} heißt die Stationarität nach (11.23)

$$\sum_{j(\neq i)} J_{ij} = 0 \quad \text{für alle } i \quad (11.31)$$

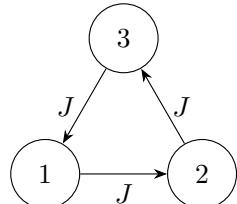
d.h. die Summe aller Ströme, also der “Nettostrom”, aus einem Zustand i heraus (oder in einen Zustand i hinein wegen $J_{ij} = -J_{ji}$) ist Null.

Die Google-Matrix S_{ij} aus Kapitel 8.3.3 ist ein Beispiel für eine stochastische Übergangsmatrix $M_{ij} = S_{ji}$ (siehe Fußnote oben zu den vertauschten Indizes), für die wir dort eine stationäre Verteilung \vec{r} , nämlich den “Wichtigkeitsvektor” \vec{r} , gesucht haben.

Eine stärkere Bedingung als Stationarität ist das sogenannte **detaillierte Gleichgewicht (detailed balance)**. Statt (11.31) fordert man dafür stärker

$$J_{ij} = 0 \quad \text{für alle } i, j. \quad (11.32)$$

Diese Forderung besagt, dass es im System **gar keine** Ströme mehr gibt.



Der entscheidende Unterschied zwischen Stationarität (11.31) und detailed balance (11.32) ist, dass bei Stationarität noch Kreisströme zugelassen sind wie in der Abb. links, während diese bei detailed balance verboten sind.

³ Ein Eigenvektor zum Eigenwert 1 und damit ein stationärer Zustand existieren nach Frobenius-Perron Theorem, wenn die Matrix $\underline{\underline{M}}$ irreduzibel ist.

Dies ist ein aus der statistischen Physik oder Thermodynamik bekannter Sachverhalt. Dort wird das Gleichgewicht so definiert, dass *keine* makroskopischen Ströme mehr fließen im System. Ein solcher Gleichgewichtszustand ist damit analog zu einem Wahrscheinlichkeitsvektor \vec{p}_{eq} , der detailed balance erfüllt. Ein Zustand mit Kreisströmen kann dagegen stationär sein in dem Sinne, dass sich die Verteilung im Zustandsraum zeitlich nicht mehr ändert, aber er beschreibt kein thermodynamisches Gleichgewicht. Gibt es z.B. einen Wärmekreisstrom, gilt nach dem zweiten Hauptsatz nach Clausius sofort $\Delta S > 0$ und der Prozess ist irreversibel und daher nicht im Gleichgewicht.

Wir suchen nun Gleichgewichts-Wahrscheinlichkeitsvektoren \vec{p}_{eq} , die detailed balance der Master-Gleichung (11.21) erfüllen. Nach der Definition (11.23) des Stroms J_{ij} gilt für diese

$$J_{ij} = 0 \iff M_{ij} p_{eq,i} = M_{ji} p_{eq,j}$$

oder

$$\frac{p_{eq,i}}{p_{eq,j}} = \frac{M_{ji}}{M_{ij}}. \quad (11.33)$$

Dies ist die **detailed balance Bedingung** für den Gleichgewichtszustand \vec{p}_{eq} .

11.2.3 Markov-Sampling, Metropolis-Algorithmus

Die Idee beim **Markov-Sampling** ist nun, zu einer gegebenen Wahrscheinlichkeitsverteilung p_i eine Markov-Kette M_{ij} so zu konstruieren, dass die Verteilung p_i genau die Gleichgewichtsverteilung der Markov-Kette ist, die die detailed balance Bedingung (11.33)

$$\frac{p_i}{p_j} = \frac{M_{ji}}{M_{ij}}$$

erfüllt. Dann wird man während eines langen Laufes der Markov-Kette genau die vorgegebene Wahrscheinlichkeitsverteilung p_i sampeln.

Ein solcher Markov-Prozess kann tatsächlich **immer** gefunden werden, und zwar in Form des **Metropolis-Algorithmus** (Metropolis, Rosenbluth, Teller 1953 [10])

$$M_{ij} = V_{ij} A_{ij} \quad \text{mit} \quad A_{ij} = \min \left(1, \frac{p_j}{p_i} \right) \quad \text{für } i \neq j, \quad (11.34)$$

wobei $V_{ij} > 0$ eine **Vorschlagswahrscheinlichkeit** für den neuen Zustand j ist und A_{ij} auch **Akzeptanzwahrscheinlichkeit** für den Übergang von i nach j genannt wird. Die Vorschlagswahrscheinlichkeit soll lediglich $V_{ij} = V_{ji}$ erfüllen, d.h. der Übergang von i nach j soll mit gleicher Wahrscheinlichkeit vorgeschlagen werden wie der Übergang zurück von j nach i und normiert sein $\sum_{j \neq i} V_{ij} = 1$ (dann gilt auch $0 < M_{ij} < 1$ wegen $0 < A_{ij} < 1$). Außerdem gilt

$$M_{ii} = 1 - \sum_{j \neq i} V_{ij} \min \left(1, \frac{p_j}{p_i} \right) \quad (11.35)$$

für die Wahrscheinlichkeit, im Zustand i zu verbleiben, um die Matrix M_{ij} stochastisch zu machen (d.h. $\sum_j M_{ij} = 1$).

Bew. der detailed balance:

Wegen $V_{ij} = V_{ji}$ fällt die Vorschlagswahrscheinlichkeit aus der detailed balance Bedingung heraus:

$$\frac{M_{ji}}{M_{ij}} = \frac{A_{ji}}{A_{ij}}.$$

Dann müssen wir 2 Fälle unterscheiden:

$$\begin{aligned} p_i > p_j \Rightarrow \frac{p_j}{p_i} < 1 &\xrightarrow{(11.34)} A_{ij} = \frac{p_j}{p_i}, \quad A_{ji} = 1 \\ \Rightarrow \frac{A_{ji}}{A_{ij}} &= \frac{p_i}{p_j} \end{aligned}$$

und

$$\begin{aligned} p_i < p_j \Rightarrow \frac{p_i}{p_j} < 1 &\xrightarrow{(11.34)} A_{ij} = 1, \quad A_{ji} = \frac{p_i}{p_j} < 1 \\ \Rightarrow \frac{A_{ji}}{A_{ij}} &= \frac{p_i}{p_j} \end{aligned}$$

In beiden Fällen erfüllt der in (11.34) definierte Metropolis-Algorithmus also detailed balance (11.33), was zu zeigen war.

Eine einfache Wahl für die Vorschlagswahrscheinlichkeit V_{ij} ist z.B. $V_{ij} = \text{const} = 1/(W-1) = V_{ji}$, wobei W die Gesamtzahl aller zugänglichen Zustände i ist. Die Metropolis-Vorschrift (11.34) sagt dann in Worten, dass wir zuerst (i) einen möglichen neuen Zustand j aus allen zugänglichen Zuständen *zufällig* mit *gleicher* Wahrscheinlichkeit auswählen und dann (ii) mit der Wahrscheinlichkeit $\min(1, p_j/p_i)$ zu diesem Zustand wechseln. Diese Vorschrift ist tatsächlich sehr einfach zu implementieren und eröffnet nun die Möglichkeit, beliebige Verteilungen p_i zu samplen, um dann z.B. in einer MC-Integration Importance-Sampling durchführen zu können. Die bei dem Beispiel der Integration benötigte Verallgemeinerung auf kontinuierliche Verteilungen ist auch leicht zu realisieren.

Bei einer MC-Simulation benutzt man den Metropolis-Algorithmus (11.34), um die Boltzmann-Verteilung der Zustandsenergien zu samplen.

11.3 MC Simulation (Beispiel Ising-Modell)

Der Metropolis MC-Algorithmus zur Simulation eines kanonischen Ensembles wird hergeleitet und am Beispiel des Ising-Modells ausführlich erläutert. Wir erläutern auch den grundsätzlichen Aufbau einer MC-Simulation.

Wir wollen im Folgenden klassische statistische Physik im **kanonischen Ensemble** betreiben. Ein System habe **Mikrozustände** i (die der Einfachheit halber erstmal als diskret angenommen werden) mit Energie $\mathcal{H}(i)$ (d.h. bei einem quantenmechanischen System nehmen wir an, dass der Hamiltonian bereits diagonalisiert ist und unsere Mikrozustände Eigenzustände sind, damit wird die statistische Physik wieder klassisch). Die kanonische **Zustandssumme** ist dann

$$Z(T) = \sum_i e^{-\beta \mathcal{H}(i)}, \quad (11.36)$$

wobei $\beta = 1/k_B T$ ist. Im Gleichgewicht folgen die Mikrozustände der **Boltzmann-Verteilung**

$$p_B(i) = \frac{1}{Z} e^{-\beta \mathcal{H}(i)} \quad (11.37)$$

Mit dieser Verteilung können dann alle **Ensemble-Mittelwerte** berechnet werden. Der Mittelwert einer beliebigen Observable $O = O(i)$ ist

$$\langle O \rangle = \sum_i p_B(i) O(i) = \sum_i \frac{1}{Z} e^{-\beta \mathcal{H}(i)} O(i). \quad (11.38)$$

Statt Integralen berechnen wir in einer **MC-Simulation** Mittelwerte vom Typ (11.38) also Summen über Mikrozustände i . Die Anzahl der Mikrozustände ist dabei

$$\#\text{Mikrozustände} \sim e^S \sim e^{10^{23}}.$$

Damit ist deterministisches systematisches Aufsummieren hoffnungslos und MC-Methoden kommen ins Spiel: Wir berechnen Summen der Art (11.38), indem wir Mikrozustände geschickt sammeln und dann Mittelwerte approximieren. Dazu werden wir ein Markov-Importance-Sampling der Boltzmann-Verteilung (11.37) implementieren mit Hilfe des Metropolis-Algorithmus.

11.3.1 Ising-Modell

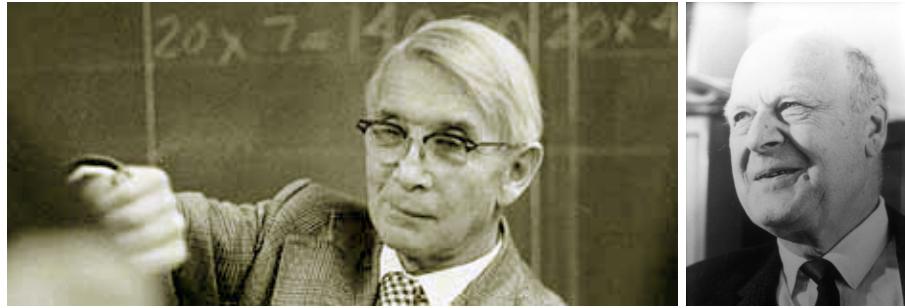


Abbildung 11.6: Links: Ernst Ising (1900-1998), deutscher Physiker. Das (eindimensionale) Ising-Modell war Thema seiner Doktorarbeit [11]. Rechts: Lars Onsager (1903-1976), norwegischer Physiker (Nobelpreis 1968). Auf ihn geht die analytische Transfermatrixelösung des zweidimensionalen Ising Modells zurück (1944).

Unser wichtigstes **Beispiel** wird im Folgenden immer das **Ising-Modell** (Ernst Ising 1925 [11]) sein. Das Ising-Modell ist ein Gittermodell, auf jedem der N Gitterplätze ($n = 1, \dots, N$) sitzt ein Spin mit jeweils zwei möglichen Zuständen $s_n = \pm 1$ (s_n beschreibt die Eigenwerte $\frac{\hbar}{2} s_n$ eines Spin-Operators $\hat{S}_{n,z}$). Die **Hamiltonfunktion** ist

$$\mathcal{H} = -\frac{1}{2} \sum_{n \neq m} J_{nm} s_n s_m - H \sum_{n=1}^N s_n. \quad (11.39)$$

Wir betrachten einfache **kubische** Gitter in D Dimensionen. Meist werden wir uns auf den Fall $D = 2$ beschränken.

$\begin{matrix} \uparrow & \uparrow & \downarrow & \downarrow \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \uparrow & \uparrow & \downarrow & \uparrow \end{matrix}$

Die Spins sind magnetisch gekoppelt über eine Austausch-Wechselwirkung mit einer **magnetischen Kopplung** J_{ij} . Wir konzentrieren uns auf eine ferromagnetische Kopplung, die gleiche Ausrichtung gekoppelter Spins bevorzugt mit $J_{ij} > 0$ (der Fall $J_{ij} < 0$ entspricht einer antiferromagnetischen Kopplung). Die Matrix $J_{ij} = J_{ji}$ ist symmetrisch und es gilt $\frac{1}{2} \sum_{i \neq j} \dots = \sum_{\text{bonds } ij} \dots$, wobei über jeden Bond (oder Paar) ij dann einmal summiert wird. Wir werden uns auch auf den Fall einer **nächsten Nachbar Wechselwirkung** beschränken, d.h.

$$J_{ij} = \begin{cases} J > 0 & i, j \text{ nächste Nachbarn} \\ 0 & \text{sonst} \end{cases}.$$

Das **Magnetfeld** H wird als homogen angenommen und der zweite Term in (11.39) entspricht gerade der Zeeman-Energie von Spins im Magnetfeld.

Jeder Mikrozustand i ist dann charakterisiert durch vollständige Angabe aller N Spins $\{s_n\} = (s_1, \dots, s_N)$. Die zugehörige Energie ist

$$\mathcal{H}(i) = \mathcal{H}(\{s_n\}) = -J \sum_{\langle n,m \rangle} s_n s_m - H \sum_{n=1}^N s_n \quad (11.40)$$

(n.N. \equiv nächste Nachbarn). Im kanonischen NHT-Ensemble (festes N , Magnetfeld H und Temperatur T) ist die Zustandssumme

$$\begin{aligned} Z(N, H, T) &= \sum_{\{s_n\}} e^{-\beta \mathcal{H}(\{s_n\})} \\ &= \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{-\beta \mathcal{H}(s_1, s_2, \dots, s_N)}. \end{aligned} \quad (11.41)$$

Typische **Observablen** im Ising-Modell sind:

- (i) \mathcal{H} selbst, d.h. $E = \langle \mathcal{H} \rangle =$ **mittlere Energie**.
- (ii) **Magnetisierung:** $M = \langle \sum_n s_n \rangle = N \langle s_n \rangle$ bzw. $m = M/N = \langle s_n \rangle$ pro Spin.
- (iii) **Spezifische Wärmekapazität:**

$$C = \frac{\partial E}{\partial T} = \frac{1}{k_B T^2} (\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2),$$

also misst die spezifische Wärme auch **Energiefluktuationen**.

- (iv) **Spin-Korrelationen:**

$$g(n, m) = \langle s_n s_m \rangle - \langle s_n \rangle \langle s_m \rangle.$$

Das Ising-Modell spielt in der statistischen Physik und der Computerphysik aus mehreren Gründen eine wichtige Rolle:

- Es ist von großer physikalischer Bedeutung in der Theorie des Magnetismus.
- Es war eines der ersten nicht-trivialen Modelle, wo analytisch streng (von Onsager 1944) gezeigt werden konnte, dass für $D = 2$ und $H = 0$ ein kontinuierlicher Phasenübergang existiert, der sich am kritischen Punkt durch Skalengesetze mit kritischen Exponenten auszeichnet, die von den Erwartungen aus der Landau-Theorie abweichen. Dies war wegweisend für die statistische Physik kritischer Phänomene.
- Das Ising-Modell lässt sich äquivalent abbilden auf **Gitter-Gas Modelle**, die den Kondensationsübergang beschreiben und auf **binäre Legierungen**, die Entmischung zeigen. Dies zeigt bereits die Universalität dieses Modells für verschiedene Phasenübergänge, die sich tatsächlich alle sehr ähnlich verhalten.
- Für die Computerphysik ist diese Modell auf Grund seiner Einfachheit wichtig. Wichtige Konzepte und neue MC-Methoden (z.B. Cluster-Algorithmen, siehe unten) sind an diesem einfachen Modell entwickelt worden.

Die Physik des Ising-Modells ist dimensionsabhängig. Die Hauptfrage im Kontext kritischer Phänomene ist das Verhalten bei $H = 0$: In einer Dimension $D = 1$ gibt es **keinen** echten Phasenübergang (bzw. $T_c = 0$) und das System ist für alle $T > 0$ im ungeordneten paramagnetischen Zustand mit verschwindender Magnetisierung $m = 0$. Für $D \geq 2$ gibt es eine kritische Temperatur $T_c > 0$, unterhalb der sich das System **spontan** in einen ferromagnetischen Zustand ordnet und eine Magnetisierung $m \neq 0$ ausbildet, siehe Abb. 11.7 und 11.8. Die Magnetisierung m ist der **Ordnungsparameter** des Ising-Modells.

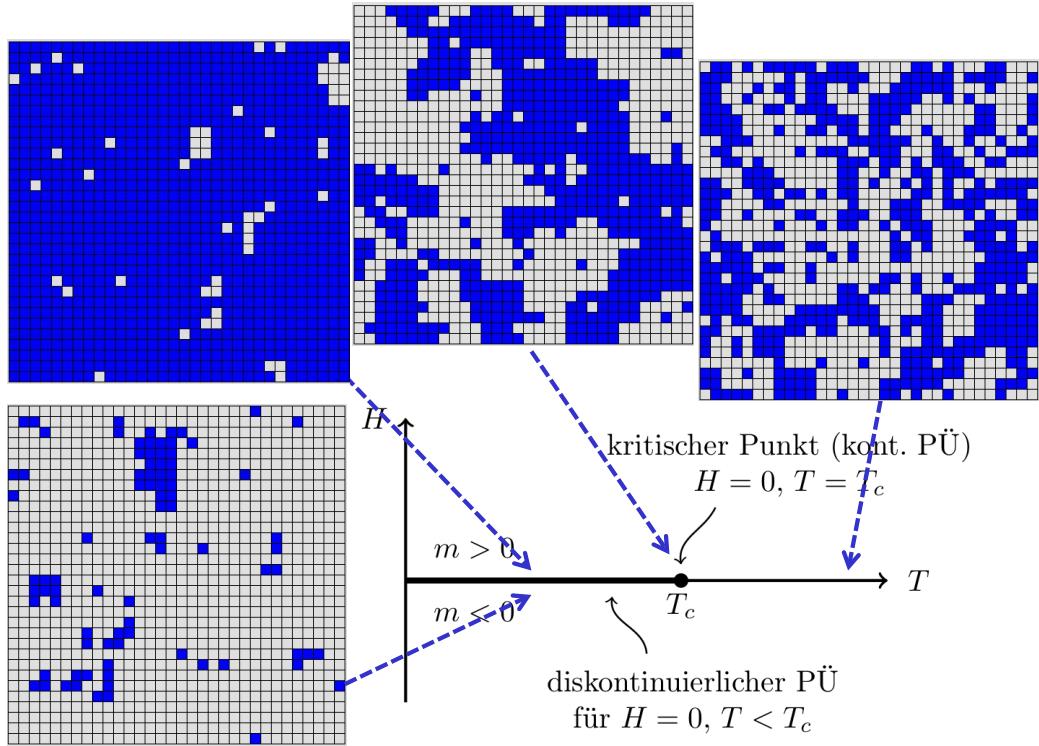


Abbildung 11.7: Schematisches Phasendiagramm des 2D Ising-Modells in der H - T -Ebene und MC-Simulations-Schnapschüsse für $H = 0$. Die kritische Temperatur beträgt $k_B T_c = 2J / \ln(1 + \sqrt{2}) = 2.269J$. Die Schnapschüsse links sind in der Tieftemperaturphase bei $k_B T = 2J < k_B T_c$ aufgenommen und zeigen spontane Ordnung auch bei $H = 0$ (entweder Spin hoch oder runter). Der Schnapschuss in der Mitte ist genau bei $T = T_c$ aufgenommen. Der Schnapschuss rechts liegt in der Hochtemperaturphase bei $k_B T = 4J$ und zeigt ein ungeordnetes System auf Grund thermischer Fluktuationen.

Neben dem **kritischen Punkt** bei $H = 0, T = T_c$ gibt es eine **Linie von Phasenübergängen 1. Ordnung** bei $H = 0, T < T_c$, die im kritischen Punkt endet, siehe Abb. 11.7. An dieser Linie springt die Magnetisierung von $M < 0$ für $H < 0$ zu $M > 0$ für $H > 0$ und wir finden die bekannte magnetische **Hysterese**.

Die MC-Simulation sollte uns in die Lage versetzen, die vorgestellten Observablen, insbesondere die Magnetisierung, zu berechnen und für $D \geq 2$ Phasenübergänge zu erkennen und zu analysieren.

11.3.2 Metropolis-Algorithmus und Ising-Modell

Eine naive Idee, um Mittelwerte in einer MC-Simulation zu berechnen, wäre ein **direktes, einfaches Sampling** einer Gleichverteilung aller Mikrozustände i . D.h. wir würden zufällig einen Mikrozustand i auswählen, also im Ising-Modell zufällig eine Spinkonfiguration (s_1, \dots, s_N) auswürfeln. Mit diesen ganz zufällig bestimmten Mikrozuständen i würden wir Summen wie in der Zustandssumme (11.36) oder dem Mittelwert einer Observablen (11.38) berechnen: Also für jeden Zustand i den Boltzmann-Faktor berechnen, evtl. den Wert einer Observablen und aufsummieren. Dies ist ein hoffnungsloses Unterfangen, da wir beispielsweise im Ising Modell 2^N Mikrozustände haben wobei

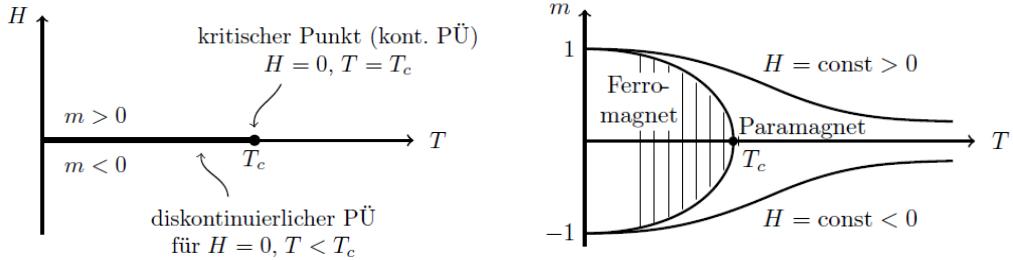


Abbildung 11.8: Links: Phasendiagramm in der H - T -Ebene (für $D \geq 2$) Eine Linie von Phasenübergängen erster Ordnung bei $H = 0$, $T < T_c$ endet im kritischen Punkt. Rechts: Magnetisierung m als Funktion der Temperatur T für verschiedene H . Für $H = 0$ gibt es Phasentrennung in ferromagnetische Domänen für $T < T_c$. Für $H > 0$ bzw. $H < 0$ ist das System in einem stabilen Gleichgewicht für $m > 0$ bzw. $m < 0$. (Für kleine $|H|$ gibt es bei tiefen Temperaturen auch noch einen metastabilen Zustand mit der jeweils anderen Magnetisierung, den man dann in der Hysterese sieht.)

$N = 100 \times 100 = 10^4$ in 2D eine normale Systemgröße ist!

Allerdings sind nicht alle Mikrozustände i gleich wichtig in den Summen (11.36) oder (11.38) wegen der Boltzmann-Verteilung. In den Boltzmann-Gewichten verschiedener Zustände gibt es exponentielle Unterschiede. Die Lösung dieses Problems liegt dann auch im **Importance-Sampling**: Wir werden die Mikrozustände i bereits genau mit den Boltzmann-Wahrscheinlichkeiten $p_B(i)$ sammeln. Um dies wiederum zu erreichen, benutzen wir das **Markov-Sampling** und letztlich den **Metropolis-Algorithmus**.

Wir suchen also einen Markov-Prozess mit Übergangswahrscheinlichkeiten M_{ij} zwischen Mikrozuständen i und j , die die **detailed balance Bedingung** erfüllen für die **Boltzmann-Verteilung** $p_B(i)$:

$$p_B(i)M_{ij} = p_B(j)M_{ji} \quad (11.42)$$

Dazu benutzen wir den Metropolis-Algorithmus (11.34). Wir zerlegen den Übergang von Zustand i nach j in jedem MC-Zeitschritt wieder in zwei Teilschritte:

- 1) Wir schlagen einen “MC-move” vor mit einer **Vorschlagswahrscheinlichkeit** V_{ij} .
- 2) Wir akzeptieren diesen Vorschlag mit einer **Akzeptanzwahrscheinlichkeit** A_{ij} .

Insgesamt gilt dann

$$M_{ij} = V_{ij}A_{ij} \quad (11.43)$$

(für $j \neq i$ und $M_{ii} = 1 - \sum_{j \neq i} M_{ij}$ auf Grund der Normierung $\sum_j M_{ij} = 1$; die Vorschlagswahrscheinlichkeit ist so normiert, dass $\sum_{j \neq i} V_{ij} = 1$, so dass $\sum_{j \neq i} M_{ij} < 1$ sichergestellt ist).

Zunächst zur Vorschlagswahrscheinlichkeit: Wir wollen MC-moves so anbieten, dass

- (i) Alle Mikrozustände i erreicht werden können, damit unsere MC-Dynamik **ergodisch** wird. D.h. dass \underline{V} irreduzibel sein muss.
- (ii) Hin- und Rückmove sollen gleich wahrscheinlich vorgeschlagen werden, d.h.

$$V_{ij} = V_{ji}. \quad (11.44)$$

Wegen (11.44) kürzt sich dann V_{ij} aus der detailed balance Bedingung (11.42) heraus und es bleibt

$$p_B(i)A_{ij} = p_B(j)A_{ji} \quad (11.45)$$

zu erfüllen. Diese detailed balance Bedingung an die Akzeptanzwahrscheinlichkeiten A_{ij} kann dann durch den **Metropolis-Algorithmus** (11.34) erfüllt werden:

$$A_{ij} = \min \left(1, \frac{p_B(j)}{p_B(i)} \right) = \min \left(1, e^{-\beta(\mathcal{H}(j) - \mathcal{H}(i))} \right). \quad (11.46)$$

Diese Wahl für A_{ij} ist tatsächlich nicht eindeutig, wird aber sehr häufig verwendet. Eine Alternative stellt der **Glauber-Algorithmus** [12] dar, wo

$$A_{ij} = \frac{1}{1 + \frac{p_B(i)}{p_B(j)}} = \frac{1}{1 + e^{\beta(\mathcal{H}(j) - \mathcal{H}(i))}} \quad (11.47)$$

gewählt wird. Auch diese Wahl erfüllt die detailed balance Bedingung:

$$\frac{A_{ji}}{A_{ij}} = \frac{1 + \frac{p_B(i)}{p_B(j)}}{1 + \frac{p_B(j)}{p_B(i)}} = \frac{p_B(i) \frac{p_B(j)}{p_B(i)} + 1}{p_B(j) 1 + \frac{p_B(j)}{p_B(i)}} = \frac{p_B(i)}{p_B(j)}.$$

Wir werden im Folgenden immer den Metropolis-Algorithmus verwenden.

Bei der Wahl der MC-moves ist etwas Vorsicht geboten: Verletzt man die unscheinbar aussehende Bedingung (11.44) an V_{ij} , müssen A_{ij} in (11.46) und der nachfolgende Algorithmus entsprechend korrigiert werden, damit insgesamt die detailed balance nicht verletzt wird (für Beispiele siehe Frenkel [1]). Also sollte man (11.44), d.h. ob Hin- und Rückmove gleich wahrscheinlich vorgeschlagen werden, immer genau prüfen.

Die Formeln (11.43), (11.44), (11.45) und (11.46) definieren dann einen **Metropolis MC-Zeitschritt** in unserer Simulation:

- 0) Zeit t , Zustand $i(t) = i$.
- 1) Biete MC-move $i \rightarrow j$ an (so dass $V_{ij} = V_{ji}$ (11.44) erfüllt).
- 2) Akzeptiere den vorgeschlagenen Move gemäß (11.46):
 - a) Berechne $\Delta E = \mathcal{H}(j) - \mathcal{H}(i)$.
 - b) Wenn $\Delta E < 0$, immer akzeptieren.
 - c) Wenn $\Delta E > 0$, ziehe gleichverteilte Zufallszahl $p \in [0, 1]$ und berechne $e^{-\beta\Delta E}$. Wenn $p < e^{-\beta\Delta E}$, akzeptieren, sonst ablehnen.
- 3) Akzeptieren \rightarrow neuer Zustand $i(t + \Delta t) = j$
Ablehnen \rightarrow neuer Zustand $i(t + \Delta t) = i = \text{alter Zustand}$.

Danach geht es in einer Zeitschleife wieder weiter mit 1). Während der Schleife (jeweils nach 3)) sollte dann gemessen werden:

- 4) Messen:
“MC-Mittel” über gesamplete Zustände (nach t_{MC} MC-Schritten)

$$\langle O \rangle_{MC} = \frac{1}{t_{MC}} \sum_{n=1}^{t_{MC}} O(i(n\Delta t)) = \langle O \rangle = \text{kanonisches Ensemble-Mittel.} \quad (11.48)$$

Wir schließen mit einigen **Bemerkungen** zum **Metropolis MC-Algorithmus**:

- Die Messvorschrift (11.48) enthält **keinen** Boltzmann-Faktor mehr. Dieser ist gerade durch das Importance-Sampling mit dem Metropolis-Algorithmus bereits berücksichtigt: Die Stichproben i werden mit der “richtigen” Wahrscheinlichkeit $p_B(i)$ gesampelt.
- Wir benötigen nur **Energiedifferenzen**, *keine* Kräfte wie bei der MD-Simulation. Bei Systemen mit **diskreten** Freiheitsgraden (wie Spins im Ising-Modell) kann man auch oft keine “Kräfte” definieren und ist deshalb auf MC angewiesen.
- Um die Energiedifferenz ΔE zu berechnen, müssen normalerweise *nicht* jedesmal die Gesamtenergien $\mathcal{H}(i)$ und $\mathcal{H}(j)$ vollständig berechnet werden.
- Die entstehende “MC-Dynamik” ist (meist) **nicht realistisch** und nur zur Berechnung **statistischer Mittelwerte** geeignet. Dynamische Größen kann man, wenn überhaupt, nur mit größter Vorsicht betrachten.
- Die Wahl der angebotenen MC-moves ist entscheidend für die Effizienz der Simulation. Im geschickten Anbieten von Moves liegt auch oft das kreative Moment bei der MC-Simulation.
- Bei sehr hohen Temperaturen $T \rightarrow \infty$ ($\beta \approx 0$) ist die Annahmewahrscheinlichkeit $e^{-\beta \Delta E} \approx 1$ auch für $\Delta E > 0$, und es wird praktisch *jeder* vorgeschlagene Schritt angenommen, nahezu unabhängig von seiner Energie.
- Bei $T \approx 0$ ($\beta \rightarrow \infty$) ist die Annahmewahrscheinlichkeit $e^{-\beta \Delta E}$ exponentiell klein für $\Delta E > 0$, und es werden praktisch nur noch Schritte mit $\Delta E < 0$ angenommen. D.h. im Limes tiefer Temperaturen $T \approx 0$ ist der Monte-Carlo Metropolis Algorithmus ein **Energie-Minimierungsalgorithmus**. Allerdings sind andere Algorithmen (conjugated Gradients oder ähnliches) oft überlegen. MC-Minimierung wird jedoch häufiger eingesetzt, wenn es viele metastabile Minima in der Energielandschaft gibt. Beim sogenannten **simulated annealing** kühlte man T dann nur schrittweise nach Null, um zu verhindern, dass man in diesen metastabilen Minima hängenbleibt. Das System kann sich dann aus nicht so tiefen metastabilen Minima (Tiefe $< k_B T$) noch befreien, solange $T > 0$ ist.

Wir wollen den Metropolis MC-Algorithmus am **Beispiel** des **Ising-Modells** mit **Einzelspin-Flips** noch detaillierter erläutern.

- Ein möglicher MC-move besteht darin, zufällig einen Spin n auszuwählen und einen Spin-Flip $s_n \rightarrow -s_n$ zu versuchen. Jeder Spin wird dann mit gleicher Wahrscheinlichkeit auch wieder für den Rück-Flip ausgewählt. Damit ist (11.44) erfüllt.
- Die Akzeptanz/Ablehnung erfolgt dann nach obigem Metropolis MC-Algorithmus. Dabei kann man sich noch folgenden **Trick** bei der ΔE -Berechnung zu Nutze machen:
Wir berechnen immer auch das **lokale mittlere Feld**

$$H_n^{MF} = J \sum_{m(\neq n)} s_m + H$$

Damit wird $\Delta E(s_n \rightarrow -s_n) = +2H_n^{MF} s_n$. Im Falle einer Akzeptanz müssen dann im nächsten Zeitschritt nur die H_m^{MF} der nächsten Nachbarn neu berechnet werden.

- Messungen werden typischerweise *nicht* in jedem Zeitschritt (wie in (11.48)), sondern nur nach einem ganzen “**MC-sweep**” durchgeführt: Ein MC-sweep entspricht N MC-moves, so dass jeder Spin (im Mittel) einmal zum Flip ausgewählt wurde,

$$\langle O \rangle_{MC} = \frac{1}{n_{\text{sweeps}}} \sum_{n=1}^{n_{\text{sweeps}}} O(i(nN\Delta t)). \quad (11.49)$$

Mehr dazu im nächsten Abschnitt.

11.3.3 Aufbau einer MC-Simulation

Für eine MC-Simulation ist der grundsätzliche Aufbau ähnlich wie bei der MD-Simulation, siehe Kapitel 5.1:

1) **Definition** des Modellsystems:

Mikrozustände i und ihre Energie $\mathcal{H}(i)$ definieren das Modell, Kräfte sind nicht nötig bei der MC-Simulation. Weiterhin muss das thermodynamische Simulationsensemble klar sein (wir behandeln hier nur das kanonische Ensemble, für andere Ensembles siehe Frenkel [1]). Außerdem müssen Randbedingungen festgelegt werden. Auf der Programmseite müssen passende Datenstrukturen für die Zustände i definiert werden.

2) **Initialisierung:**

Der Anfangszustand $i(t=0)$ wird festgelegt.

3) **Äquilibrierung:**

Auch die MC-Simulation muss viele MC-Schritte “warmlaufen”. Der Anfangszustand sollte “vergessen” werden.

4) **Messung:**

Die Messung erfolgt durch Mittelung über die MC-Zeit, siehe (11.48) bzw. (11.49). Dies sollte dem kanonischen Ensemble-Mittel entsprechen. Es sollte auf der einen Seite möglichst oft gemessen werden, um den Fehler zu drücken. Auf der anderen Seite funktioniert dies nur, wenn die Messungen unabhängig sind, d.h. wenn genügend viele MC-Schritte zwischen den Messungen liegen.

5) **MC-Schleife:**

Sowohl während der Äquilibrierung 3) als auch während der eigentlichen MC-Simulation zur Messung 4) läuft der Metropolis MC Algorithmus in einer Zeitschleife.

Wir wollen wieder einige Punkte am **Beispiel des Ising-Modells** noch detaillierter erläutern:

- zu 1) Wir arbeiten mit obiger Einzelspin-Flip Metropolis-Dynamik, die die Magnetisierung bei gegebenem Magnetfeld ändert, also im NHT-Ensemble.

Wir arbeiten häufig mit **periodischen Randbedingungen**. In 2D mit einem Spinnetz $s_{n,m}$ ($n, m = 1, \dots, N$) heißt das, wir führen an den Rändern Extra-Spins ein mit:

$$\begin{aligned} s_{N+1,m} &\equiv s_{1,m} & s_{0,m} &\equiv s_{N,m} \\ s_{n,N+1} &\equiv s_{n,1} & s_{n,0} &\equiv s_{n,N} \end{aligned}$$



- zu 2) mögliche Anfangszustände sind: alle Spins zeigen in die gleiche Richtung (gut bei tiefen T) oder alle Spins sind zufällig gewählt (gut bei hohen T)
- zu 3) Die nötige Dauer der Äquilibrierung kann z.B. dadurch getestet werden, dass man Messungen mit verschiedenen Anfangszuständen beginnt und feststellt, wann die Messungen beginnen übereinzustimmen.
- zu 4) Man sollte in einer MC-Simulation möglichst oft messen, da

$$\text{Fehler} \sim \frac{1}{\sqrt{\# \text{ Messungen}}}$$

(11.50)

(wie schon bei der MC-Integration). Allerdings gilt (11.50) nur für **unabhängige** Messungen. Es gibt ein ähnliches Problem wie bei der Äquilibrierung; das System sollte den Zustand der vorherigen Messung bei der nächsten Messung “vergessen” haben für wirklich unabhängige

Messungen. Dies geschieht auf der MC-Zeitskala einer **Autokorrelationszeit**. Es ist i.Allg. aber nicht problematisch, wenn man zu oft misst, nur die Fehlerabschätzung (11.50) gilt dann nicht mehr. Ein guter Kompromiss besteht darin, ca. einmal pro MC-Sweep zu messen.

11.4 MC-Simulation kontinuierlicher Systeme

Am Beispiel eines N -Teilchensystems mit Paar-Wechselwirkungen werden einige Besonderheiten von MC-Simulationen an kontinuierlichen Systemen erläutert.

MC-Simulationen kann man natürlich nicht nur auf Gittermodelle mit diskreten Freiheitsgraden wie das Ising-Modell anwenden. Auch kontinuierliche “Off-Lattice” Systeme wie ein einatomiges neutrales Gas mit einer Lennard-Jones-Wechselwirkung, wie es im Kapitel 5.2 zur MD-Simulation eingeführt wurde, können mit Hilfe von MC-Methoden simuliert werden.

Wir wollen hier wie in der MD-Simulation ein System aus N Teilchen mit einer Paar-Wechselwirkung $V(|\vec{r}|)$ und Massen m betrachten. Jeder Mikrozustand ist dann charakterisiert durch die Angabe **aller** N Teilchenpositionen \vec{r}_n und Teilchenimpulse \vec{p}_n in einem Vektor $(\{\vec{r}_n\}, \{\vec{p}_n\})$. Die Energie des Systems ist

$$\mathcal{H}(\{\vec{r}_n\}, \{\vec{p}_n\}) = \sum_{n=1}^N \frac{\vec{p}_n^2}{2m} + \sum_{n < m} V(|\vec{r}_n - \vec{r}_m|) \quad (11.51)$$

Im Gegensatz zur MD-Simulation wollen wir mit Hilfe der MC-Simulation ein **kanonisches Ensemble** mit festem N , V und T (NVT-Ensemble) simulieren. Die kanonische Zustandssumme ist

$$Z(N, V, T) = \frac{1}{N! h^{3N}} \left(\prod_{n=1}^N \int d^3 \vec{r}_n \int d^3 \vec{p}_n \right) e^{-\beta \mathcal{H}(\{\vec{r}_n\}, \{\vec{p}_n\})}. \quad (11.52)$$

Typische **Observablen** $O = O(\{\vec{r}_n\}, \{\vec{p}_n\})$ sind wie im Kapitel 5.4:

- (i) Die **kinetische Energie** $E_{kin} = \sum_n \frac{\langle \vec{p}_i^2 \rangle}{2m}$.
- (ii) Die **potentielle Energie** $E_{pot} = \sum_{n < m} \langle V(|\vec{r}_n - \vec{r}_m|) \rangle$.
- (iii) Die Gesamtenergie $E = E_{kin} + E_{pot}$.
- (iv) Die radiale **Paarverteilungsfunktion** $g(r)$.
- (v) Die spezifische Wärme

$$C_V = \frac{\partial E}{\partial T} = \frac{1}{k_B T^2} (\langle \mathcal{H}^2 \rangle - \langle \mathcal{H} \rangle^2).$$

Die \vec{p} -Integrationen in (11.52) sind gaußisch und können problemlos analytisch ausgeführt werden. Mit der **de Broglie Wellenlänge**

$$\Lambda(T) = \left(\frac{h^2}{2\pi m k_B T} \right)^{1/2} \propto T^{-1/2} \quad (11.53)$$

bekommen wir

$$Z(N, V, T) = \frac{1}{N!} \Lambda^{-3N} \underbrace{\left(\prod_{n=1}^N \int d^3 \vec{r}_n \right) e^{-\beta \mathcal{H}_{pot}(\{\vec{r}_n\})}}_{= Q(N, V, T) \text{ Konfigurationsintegral}}. \quad (11.54)$$

Impulsabhängige Observablen $O = O(\{\vec{p}_n\})$ wie E_{kin} können wegen der einfachen quadratischen \vec{p} -Abhängigkeit in (11.51) oft analytisch berechnet werden (siehe z.B. Äquipartitionstheorem).

Mittelwerte ortsabhängiger Observablen $O = O(\{\vec{r}_n\})$,

$$\langle O \rangle = \frac{1}{Q} \left(\prod_{n=1}^N \int d^3 \vec{r}_n \right) O(\{\vec{r}_n\}) e^{-\beta \mathcal{H}_{pot}(\{\vec{r}_n\})} \quad (11.55)$$

wollen wir dagegen mit einer MC-Simulation im **Konfigurationsraum** $\{\vec{r}_n\}$ mit Hamiltonian $\mathcal{H} = \mathcal{H}_{pot}(\{\vec{r}_n\})$ ohne \vec{p} -Abhängigkeit berechnen. Dazu nutzen wir wieder den **Metropolis MC-Algorithmus**. Dabei gilt es zu beachten:

- **MC-moves** sehen im Kontinuum folgendermaßen aus: Wir wählen zufällig ein Teilchen n aus und versuchen eine Verschiebung
$$\vec{r}_n \rightarrow \vec{r}_n + \Delta \vec{r}$$
um einen gleichverteilten Zufallsvektor $\Delta \vec{r} \in [-\Delta x, \Delta x]^3$.
- MC-moves dieser Art erfüllen die Ergodizitätsbedingung, d.h. jede Teilchenkonfiguration ist erreichbar.
- Diese MC-moves erfüllen auch die Symmetrieverteilung (11.44), d.h. Hin- und Rückmove werden mit gleicher Wahrscheinlichkeit vorgeschlagen.
- Wir können den Vektor $\Delta \vec{r}$ auch aus anderen Verteilungen ziehen, solange Ergodizität und gleiche Wahrscheinlichkeit von Hin- und Rückmove gewährleistet bleiben. Andere Möglichkeiten wären z.B. gleichverteilt aus einer Kugel $|\Delta r| < \Delta x$ zu ziehen oder aus einer 3-dimensionalen Gaußverteilung mit Varianz Δx^2 .
- Dann berechnen wir wieder ΔE für diesen Move. Dabei kann man hier evtl. **Nachbarschaftslisten** verwenden, wo man für jedes Teilchen abspeichert, welche Teilchen in einem gewissen Abschneideradius liegen, so dass deren Wechselwirkungsenergien noch berechnet werden müssen. Diese Nachbarschaftslisten muss man nicht in jedem MC-Zeitschritt updaten. Daher kann so Computerzeit gespart werden.
- Dann wenden wir wieder die Metropolis-Regel an: Wir akzeptieren den Move, wenn $p < e^{-\beta \Delta E}$ mit einer gleichverteilten Zufallszahl $p \in [0, 1]$.

Es bleibt die Frage zu klären, wie groß der **Parameter Δx** für **kontinuierliche** MC-moves zu wählen ist (einen solchen Parameter gab es bei *diskretem* Spin-Flip gar nicht):

Ist Δx zu klein, werden fast alle Moves angenommen, aber nur **kleine** Änderungen der Konfiguration erreicht. Dies führt zu einem schlechten Sampling von Konfigurationen. Ist aber Δx zu groß gewählt, wird ein Großteil der vorgeschlagenen Moves abgelehnt, weil sich die Energie zu stark ändert, was dann auch zu einem schlechten Sampling führt.

Erfahrungsgemäß gibt eine **Akzeptanzrate** $\sim 50\%$ eine optimale Sampling-Geschwindigkeit. Δx sollte also so eingestellt werden, dass sich eine solche Akzeptanzrate einstellt. Allerdings sollte man Δx **nicht während** der Messung ändern in einer MC-Simulation. Dies kann leicht zu einer Verletzung der Symmetrieverteilung (11.44) und damit der detailed balance führen.

11.5 Skalengesetze, Finite-Size-Effekte

Wir erinnern an das Verhalten der Korrelationslänge an Phasenübergängen und kritische Exponenten und Skalengesetze an kritischen Punkten, insbesondere am Beispiel des Ising-Modells. Dann werden Finite-Size-Effekte und Finite-Size-Scaling Techniken zur MC-Datenanalyse diskutiert.

11.5.1 Korrelationslänge und Skalengesetze

Eine überaus wichtige Längenskala an einem Phasenübergang ist die **Korrelationslänge** ξ . Sie beschreibt räumliche **Ordnungsparameterfluktuationen/-korrelationen**.

Im Ising-Modell mit der Magnetisierung $m = \langle s_n \rangle$ als Ordnungsparameter betrachten wir dazu **Spinkorrelationen**

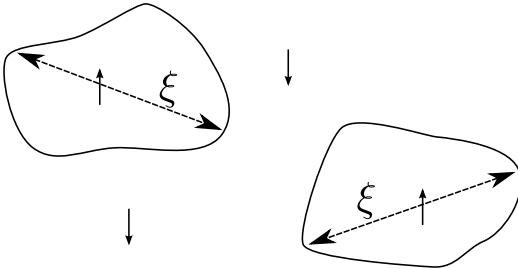
$$G(|\vec{r}_n - \vec{r}_m|) = \langle s_n s_m \rangle - \langle s_n \rangle \langle s_m \rangle, \quad (11.56)$$

wo $|\vec{r}_n - \vec{r}_m|$ die Entfernung zwischen Spins n und m (im Gitter) sein soll. Wenn das System **nicht** an einem kritischen Punkt ist, zerfallen diese Korrelationen exponentiell

$$G(r) \propto e^{-r/\xi}. \quad (11.57)$$

Die **Korrelationslänge** ξ ist gerade als charakteristische Längenskala für diesen Zerfall definiert, d.h. Spins die weiter voneinander entfernt sind als ξ weisen nahezu unabhängige Richtungen auf.

Im Ising-Modell kann man die Korrelationslänge ξ auch anschaulicher deuten als die **typische Domänengröße** von magnetischen Domänen gleicher Spinausrichtung:



Für $T < T_c$ ist ξ die typische Größe der Minoritätsdomänen. Für $T > T_c$ ist ξ die typische Größe der geordneten ferromagnetischen Domänen in einem ansonsten ungeordneten paramagnetischen System.

Bei **kontinuierlichen Phasenübergängen** divergiert ξ am kritischen Punkt $T = T_c$ wie

$$\xi \propto |t|^{-\nu} \quad (11.58)$$

als Funktion der **reduzierten Temperatur**

$$t \equiv \frac{T - T_c}{T_c}$$

und mit einem für den jeweiligen kontinuierlichen Phasenübergang charakteristischem **Exponenten** ν . Diese Divergenz bedeutet im Ising-Modell, dass die Ordnung bei Annäherung an den kontinuierlichen Phasenübergang für $T \downarrow T_c$ durch unendliches Wachstum einer geordneten Domäne entsteht.

Bei einem **diskontinuierlichen Phasenübergang (1. Ordnung)** hat man dagegen **Phasenseparation** und **Koexistenz zweier Phasen** am Phasenübergang. Dann bleibt die Korrelationslänge ξ endlich. Für das Ising-Modell ist das unterschiedliche Verhalten der magnetischen Domänen in Abb. 11.9 erläutert.

Direkt an einem **kritischen Punkt** $T = T_c$, wo $\xi = \infty$ ist nach (11.58), gilt für den Zerfall der Korrelationsfunktion ein **Potenzgesetz**

$$G(r) \propto \frac{1}{r^{D-2+\eta}} \quad (11.59)$$

anstatt des exponentiellen Zerfalls (11.57). Der **Exponent** η ist wieder für den jeweiligen kontinuierlichen Phasenübergang charakteristisch.

In der Nähe des kritischen Punktes $T \approx T_c$ ist also ξ sehr groß nach (11.58). Die Physik wird dort dominiert von Fluktuationen auf dieser großen Längenskala. Im Ising-Modell wird das magnetische

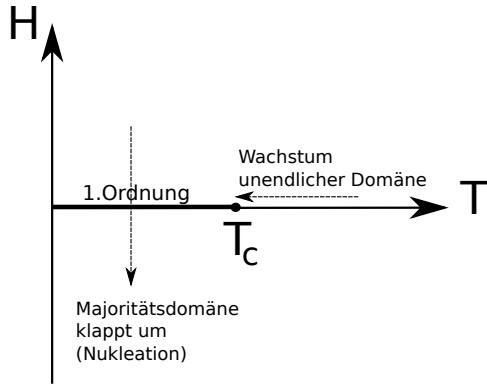


Abbildung 11.9: Unterschiedliches Verhalten der magnetischen Domänen am kritischen Punkt $H = 0$ und $T = T_c$ und am Phasenübergang erster Ordnung für $H = 0$ und $T < T_c$ im Ising-Modell.

Verhalten am kritischen Punkt beispielsweise durch das Verhalten der großen Domänen bestimmt. Dies führt zu **Skalengesetzen** zwischen verschiedenen Observablen mit charakteristischen **Exponenten**, da das Verhalten vieler Observablen über die Korrelationslänge zusammenhängt. Dies führt auch dazu, dass viele mikroskopische Details des Modells, die auf kleineren Längenskalen relevant sind, nicht mehr wichtig sind und ist damit der Grund für die **Universalität** von kritischem Verhalten: Ein Ising-Magnet weist z.B. am kritischen Punkt die gleichen Exponenten wie kondensierendes CO_2 auf. Es stellt sich heraus, dass die kritischen Exponenten i.Allg. nur von a) der **Symmetrie** des Systems und b) der **Dimension** des Systems und des Ordnungsparameters abhängen.

Beispiele für **Skalengesetze** am kritischen Punkt des Ising-Modells sind für die Magnetisierung

$$m = \frac{M}{N} \propto |t|^\beta \propto \xi^{-\beta/\nu} \quad (11.60)$$

oder für die spezifische Wärme (pro Spin)

$$c = \frac{1}{N} \frac{\partial E}{\partial T} \propto |t|^{-\alpha} \propto \xi^{\alpha/\nu} \quad (11.61)$$

mit kritischen Exponenten β bzw. α . Die charakteristischen **kritischen Exponenten des Ising-Modells** sind:

D	β	α	ν	η
2	1/8	0 (\log)	1	1/4
3	0,32	0,11	0,63	0,04

11.5.2 Finite-Size-Scaling

Die kritischen Exponenten charakterisieren das singuläre, nicht-analytische Verhalten der freien Energie und aus ihr abgeleiteter Observablen am kritischen Punkt im Limes $|t| \rightarrow 0$ oder $\xi \rightarrow \infty$. Man kann nun aber in der Computerphysik nur **endliche** Systeme einer Größe L simulieren. Dies führt dazu, dass für $\xi > L$ **Finite-Size-Effekte** in Simulationen auftauchen. Dies bezieht sich natürlich auf alle Simulationen, also nicht nur MC-Simulationen.

Wie oben beschrieben ordnet sich das Ising-Modell bei $H = 0$ und Annäherung an die kritische Temperatur T_c durch unendliches Wachstum einer Domäne, wenn $\xi \rightarrow \infty$. Eine unendlich große Domäne kann dann nicht mehr umklappen (in endlicher Zeit, siehe auch nächste Kapitel, z.B.

Gleichung (11.66)) und eine spontane Magnetisierung $M \neq 0$ stellt sich ein. Im endlichen System können sich nur endliche Domänen bilden der Maximalgröße L . Diese Domänen können (i) in endlicher Zeit auch in der Tieftemperaturphase $T < T_c$ noch umklappen und (ii) sich bereits bei Temperaturen $T > T_c$ bilden. Wegen (i) ist die Magnetisierung in einem endlichen System auch bei $T < T_c$ streng genommen $\langle M \rangle = 0$, weil das System immer noch zwischen den beiden Zuständen $\pm M$ hin- und herwechseln kann in endlicher Zeit. Daher sollte man in einem endlichen System eher $\langle |M| \rangle \equiv \langle |\sum_n s_n| \rangle$ messen als $\langle M \rangle$, um den Phasenübergang festzustellen. Außerdem erscheint wegen (ii) die kritische Temperatur T_c im endlichen System leicht erhöht, da sich endliche Systeme leichter ordnen, weil sich nur eine endliche Domäne der Größe L bilden muss. In Abb. 11.11(E) erkennt man dies an der Verschiebung des Maximums der spezifischen Wärme mit der Systemgröße L .

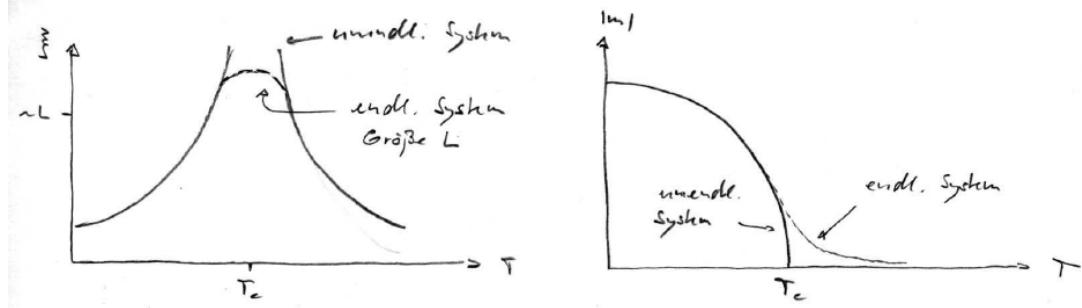


Abbildung 11.10: Schematisch: Finite-Size-Effekte für Korrelationslänge ξ und Magnetisierung m im Ising-Modell.

Die am kritischen Punkt typischen Divergenzen und Nicht-Analytizitäten, z.B. bei Magnetisierung (11.60) und spezifischer Wärme (11.61), werden dadurch bei $\xi \sim L$ "abgeschnitten", siehe Abb. 11.10. Während also im unendlich großen System (im thermodynamischen Limes) nahe am kritischen Punkt $m \propto \xi^{-\beta/\nu}$ gelten sollte nach (11.60), gilt im endlichen System $\langle |m| \rangle(t=0) = L^{-\beta/\nu}$ am kritischen Punkt. Die Magnetisierung am kritischen Punkt ist also nicht mehr Null ($\xi \rightarrow \infty$), sondern nimmt mit der Systemgröße ab, siehe Abb. 11.11(A). Entsprechend hat die spezifische Wärme keine Divergenz $c \propto \xi^{\alpha/\nu}$ mehr am kritischen Punkt ($\xi \rightarrow \infty$), sondern die Divergenz wird von der Systemgröße "abgeschnitten" bei $c(t=0) \propto L^{\alpha/\nu}$ (im 2D Ising Modell ist $\alpha = 0$, d.h. die Divergenz ist logarithmisch $c \propto \ln L$), siehe Abb. 11.11(E).

Genauer kann dies durch

$$\langle |m| \rangle = L^{-\beta/\nu} f_m \left((L/\xi)^{1/\nu} \right) = L^{-\beta/\nu} f_m \left(L^{1/\nu} |t| \right) \quad (11.62)$$

mit einer **Skalenfunktion**

$$f_m(x) \approx \begin{cases} \text{const} & x \ll 1 (\xi \gg L) \\ x^\beta & x \gg 1 (\xi \ll L) \text{ und } t < 0 \\ x^{\beta-1} & x \gg 1 (\xi \ll L) \text{ und } t > 0 \end{cases} \quad (11.63)$$

beschrieben werden. Ein analoges Skalengesetz kann man auch für die spezifische Wärme formulieren. Die Form der Skalenfunktion für die Magnetisierung ergibt daraus, dass (i) bei $\xi \gg L$ (kleine Systeme oder nah am kritischen Punkt) das finite-size Verhalten $\langle |m| \rangle = L^{-\beta/\nu}$ vorliegt, (ii) in der Tieftemperaturphase $T < T_c$ und bei $\xi \ll L$ (große Systeme) wieder $\langle |m| \rangle \propto \xi^{-\beta/\nu}$ gelten sollte und L aus der Relation (11.62) herausfallen muss, (iii) in der Hochtemperaturphase $T > T_c$ und bei $\xi \ll L$ (große Systeme) die Magnetisierung wie $\langle |m| \rangle \propto L^{-1} \rightarrow 0$ verschwinden sollte (nach zentralem Grenzwertsatz für unabhängig fluktuierende Spins sollte $\langle |m|^2 \rangle \propto N^{-1/2} \propto L^{-1}$ gelten).

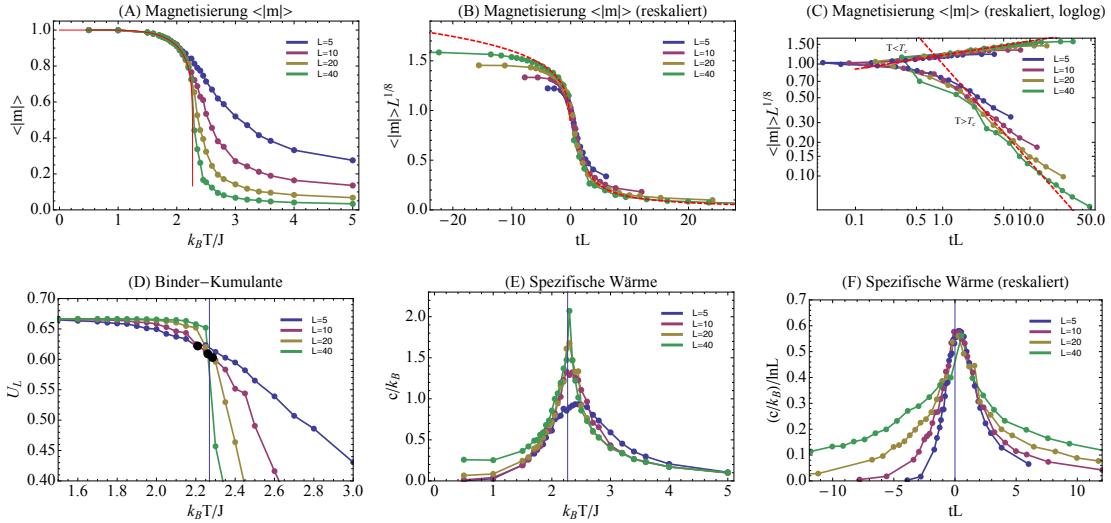


Abbildung 11.11: Metropolis Monte-Carlo Ergebnisse für (A,B,C) Magnetisierung $\langle |m| \rangle$, (D) Binder-Kumulante $U_L(T)$ und (E,F) spezifische Wärme c im 2D Ising-Modell. Die rote Linie in (A) ist das exakte Ergebnis $m = (1 - \sinh^{-4}(2J/k_B T))^{1/8}$ von Onsager für die Magnetisierung, aus der mit $m(T_c) = 0$ die exakte kritische Temperatur $k_B T_c/J = 2/\ln(1 + \sqrt{2}) = 2.269$ folgt. Simulationen sind für 4 Systemgrößen $L = 5, 10, 20, 40$ ($N = L^2$) über 10^4 MC sweeps mit 10^3 warmup MC sweeps gefahren, und es wurde alle 20 MC sweeps gemessen. In (B,C,F) sind Magnetisierung $\langle |m| \rangle L^{1/8}$ und spezifische Wärme $c / \ln L$ reskaliert gegen die reskalierte reduzierte Temperatur tL aufgetragen ($\beta = 1/8$, $\nu = 1$, $\alpha = 0$ im 2D Ising Modell), siehe Gl. (11.64) für die Magnetisierung. In der Nähe des kritischen Punktes fallen alle Daten auf eine Masterkurve. Die roten gestrichelten Linien in (B,C) entsprechen Potenzgesetzen $f_m(x) \propto x^{1/8}$ für $T > T_c$ und $f_m(x) \propto x^{-7/8}$ für $T < T_c$, siehe Gl. (11.63). Die Punkte in Abb. (D) mit der Binder-Kumulante zeigen Schnittpunkte für aufeinanderfolgende Systemgrößen, die Binder-Kumulanten für $L = 20$ und 40 schneiden sich bei $k_B T_c/J \simeq 2.269$.

Daher muss man dann beispielsweise Simulationsdaten für die Magnetisierung in endlichen Systemen mit Hilfe eines sogenannten **Finite-Size-Scaling** für **verschiedene Systemgrößen** untersuchen. Das heißt, man verwendet (11.62) in der Form

$$\langle |m| \rangle L^{\beta/\nu} = f_m(L^{1/\nu}|t|) \quad (11.64)$$

und plottet dann Kurven von $\langle |m| \rangle L^{\beta/\nu}$ für verschiedene Systemgrößen L gegen $x \equiv L^{1/\nu}|t|$. Dabei sollte sich im Idealfall gemäß (11.64) ein **Datenkollaps** aller Daten auf **eine** Kurve $f_m(x)$ ergeben, siehe Abb. 11.11(B,C). Allerdings gelingt dieser Kollaps nur mit den *richtigen* Exponenten β , und ν und der *richtigen* kritischen Temperatur T_c (in t).

Daraus ergibt sich als Methode, um Exponenten und kritische Temperatur zu bestimmen, dass man β , ν und T_c solange variiert, bis der Datenkollaps möglichst gut ist.

Um T_c mit Hilfe von (11.64) zu bestimmen, benutzt man auch die Beobachtung, dass Kurven $mL^{\beta/\nu}$ für verschiedene L einen **gemeinsamen Schnittpunkt** genau bei $T = T_c$ haben sollten (weil dann $|t| = 0$ auf der rechten Seite und die L -Abhängigkeit dort herausfällt). Diese Beobachtung kann selbst noch nicht direkt verwendet werden, um T_c zu bestimmen, es sei denn, β und ν sind

bereits bekannt. Eine Bestimmung der kritischen Temperatur des Ising-Modells *ohne* seine kritische Exponenten bereits zu kennen, gelingt dann über die sogenannte **Binder-Kumulante** [13]

$$U_L(T) = 1 - \frac{\langle m^4 \rangle_L(T)}{3\langle m^2 \rangle_L^2(T)}. \quad (11.65)$$

Für $T \ll T_c$ gilt $U_L(T) \approx 2/3$, da dort im geordneten Zustand $\langle m^4 \rangle_L \approx 1$ und $\langle m^2 \rangle_L \approx 1$ gilt. Für $T \gg T_c$ wird $U_L(T) \approx 0$, da dort die Magnetisierung gaußverteilt ist, so dass $\langle m^4 \rangle_L \approx 3\langle m^2 \rangle_L^2$ (für eine Gaußverteilung verschwinden alle Kumulanten höherer Ordnung als 2, insbesondere auch die 4te Kumulante). Die Temperaturverläufe der Binderkumulante zwischen diesen beiden festen Grenzwerten sind jedoch L -abhängig. Allerdings gilt eine ähnliche Finite-Size Skaleneigenschaft wie (11.64) für die Magnetisierung auch für die höheren Momente der Magnetisierung $\langle m^n \rangle_L \sim L^{-n\beta/\nu} f_n(L^{1/\nu}|t|)$. Dann schneiden sich aber *alle* Kurven $U_L(T)$ für verschiedene L bei $t = 0$ oder $T = T_c$ in einem gemeinsamen Schnittpunkt $U_L(T_c) = 1 - f_4(0)/3f_2^2(0)$. Daher kann $T = T_c$ über den Schnittpunkt der Binderkumulanten $U_L(T)$ für verschiedene L bestimmt werden, siehe Abb. 11.11(D).

11.6 Cluster-Algorithmen

Wir erklären das Phänomen des “critical slowing down” und diskutieren den Wolff-Algorithmus, der ganze Spin-Cluster flippt und kaum “critical slowing down” zeigt, für das Ising-Modell.

Ein Problem des in Kapitel 11.3 vorgestellten klassischen Metropolis MC-Algorithmus mit Einzelspin-Flips ist, dass er sich stark verlangsamt in der Nähe des kritischen Punktes. Dieses Phänomen wird auch als “critical slowing down” bezeichnet.

Was ist der Grund dafür? Am kritischen Punkt werden die magnetischen Domänen sehr groß (ξ divergiert). Zwei aufeinanderfolgende Zustände sind aber nur als zeitlich dekorreliert und unabhängig anzusehen, wenn diese Domänen hin- und her geflippt wurden. Dieser Vorgang sollte eine MC-Zeit

$$t_{MC} \propto \xi^z \quad (11.66)$$

kosten, die im Wesentlichen von der Domänengröße ξ abhängt, wobei z der sogenannte **dynamische Exponent** ist, der für die jeweilige Simulationsdynamik charakteristisch ist. Für die Einzelspin-Flip Dynamik werden wir $z \approx 2$ zeigen. Mit einem relativ großen Exponenten von $z \approx 2$ wird die benötigte Zeit t_{MC} in (11.66) dann schnell groß bei $\xi \rightarrow \infty$ am kritischen Punkt. Dies ist das “critical slowing down”.

Wir wollen nun plausibel machen, dass $z \approx 2$ für die Einzelspin-Flip Dynamik gilt: In der Einzelspin-Flip Dynamik werden Domänen vom Rand her mit Hilfe einzelner Spin-Flips gedreht. Pro MC-sweep macht der Domänenradius R dabei näherungsweise einen Random-Walk-Schritt, d.h. es gilt $\langle (R - R_0)^2 \rangle \propto t_{MC}$ bei t_{MC} MC-sweeps. Das heißt nach t_{MC} MC-sweeps bringt die Einzelspin-Flip Dynamik eine Dekorrelation über eine Länge $\propto \sqrt{t_{MC}}$. Damit eine ganze Domäne der Größe ξ dekorreliert, muss also gelten

$$\sqrt{t_{MC}} \propto \xi \Rightarrow t_{MC} \propto \xi^2, \text{ also } z = 2 \text{ in (11.66).}$$

Um Gleichgewichtsmittelwerte schneller zu berechnen, brauchen wir neue MC-moves, in denen ganze Domänen bzw. “Cluster” von Spins gleicher Richtung *auf einmal* geflippt werden. Genau dies leisten die sogenannten **Cluster-Algorithmen**. Prominente Beispiele sind der **Swendsen-Wang-Algorithmus** von 1987 [14], der viele Cluster konstruiert pro MC-Schritt und der **Wolff-Algorithmus** von 1988 [15], der nur einen Cluster pro MC-Schritt konstruiert.

Wir werden uns hier mit dem **Wolff-Algorithmus** befassen, der folgendermaßen arbeitet:

- 1) Zufällige Auswahl eines Spins.
- 2) Konstruktion eines Clusters C von Spins gleicher Richtung, der diesen Spin enthält (dies legt die Vorschlagswahrscheinlichkeit V_{ij} des MC-moves fest).
- 3) Flippen des **ganzen** Clusters (mit Akzeptanzwahrscheinlichkeit $A_{ij} = 1$).

Der entscheidende Schritt ist offensichtlich Schritt 2. Der konstruierte Cluster ist dabei etwas kleiner als die ganze magnetische Domäne, der der ausgewählte Spin angehört. Die Nachbarspins gleicher Richtung werden jeweils nur mit einer Wahrscheinlichkeit p_c hinzugefügt. Das genaue Verfahren zur Cluster-Konstruktion in 2) ist folgendes:

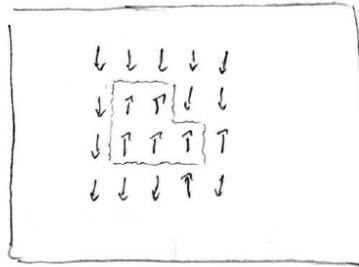
- 2a) Wir gehen zu jedem Spin $s_n \in C$, der noch nicht "besucht" wurde, alle Nachbarn $s_n \notin C$ durch (also die, die noch nicht im Cluster sind).
- 2b) Wenn ein Nachbar s_m in die andere Richtung zeigt, wird er *nicht* zum Cluster hinzugefügt (also $\rightarrow s_m \notin C$).
Wenn dagegen der Nachbar in die gleiche Richtung zeigt, wird er mit der Wahrscheinlichkeit p_c zum Cluster hinzugefügt ($\rightarrow s_m \in C$) und zunächst als "unbesucht" markiert.
- 2c) Dann wird s_n als "besucht" markiert. Danach gehen wir wieder zu Schritt 2a) mit dem nächsten "unbesuchten" Spin aus C , bis es keine "unbesuchten" Spins mehr gibt.

Um dieses Verfahren zu implementieren, müssen wir die "unbesuchten" Spins $\in C$ in einen Puffer speichern (entweder einen Stapel oder FIFO (first in first out) oder eine Schlange/Queue oder LIFO (last in first out)).

Es bleibt die wichtige Frage, wie die Wahrscheinlichkeit p_c zu wählen ist, damit der Wolff-Algorithmus detailed balance erfüllt, wenn Cluster-Flips mit Sicherheit akzeptiert werden sollen ($A_{ij} = 1$). Dann müssen wir nach (11.42) erfüllen:

$$e^{-\beta(E(i)-E(j))} = \frac{p_B(i)}{p_B(j)} \stackrel{!}{=} \frac{V_{ji}A_{ji}}{V_{ij}A_{ij}} = \frac{V_{ji}}{V_{ij}}. \quad (11.67)$$

Wir betrachten dazu den Rand eines Clusters:



Hier gibt es $N = 10$ Bonds am Rand. Davon gehen $m = 2$ Bonds zu Spins mit der gleichen Richtung wie der Cluster.

- Der Cluster habe N Bonds zu Nachbarspins, die nicht mehr zum Cluster gehören.
- m von diesen Bonds seien zu Spins **gleicher** Richtung, dann sind $N - m$ Bonds zu Spins mit **umgekehrter** Richtung.

Die Wahrscheinlichkeit, einen solchen Cluster im Wolff-Algorithmus für den Flip-move $i \rightarrow j$ auszuwählen, ist nach Definition des Algorithmus $= p_I(1 - p_c)^m = V_{ij}$, wobei p_I die Wahrscheinlichkeit bezeichnet, die Spins entlang der Bonds im Innern des Clusters mit der Wahrscheinlichkeit von jeweils p_c ausgewählt zu haben, dass der Cluster entsteht.

- Nach dem Cluster-Flip sind wir in einem Zustand j mit:
 m Bonds am Rand in **umgekehrter** Richtung (diese ehemals ferromagnetischen Bonds werden beim Flip "gebrochen") und $N - m$ Bonds in **gleicher** Richtung (diese ferromagnetischen Bonds entstanden neu beim Flip).

- Die Wahrscheinlichkeit in Zustand j den **gleichen** Cluster für den Rückmove auszuwählen ist entsprechend $= p_I(1 - p_c)^{N-m} = V_{ji}$. Die Auswahlwahrscheinlichkeit p_I der Cluster-Spins entlang der Bonds im Inneren ist die gleiche wie bei dem Hinmove.

Damit gilt also

$$\boxed{\frac{V_{ji}}{V_{ij}} = (1 - p_c)^{N-2m}} \quad (11.68)$$

- Die Energiedifferenz zwischen Zustand i und Zustand j nach dem Clusterflip beträgt

$$\begin{aligned} E(j) - E(i) &= 2J \times (\text{Zahl gebrochener Bonds}) - 2J \times (\text{Zahl neuer Bonds}) \\ &= 2Jm - 2J(N - m) \\ &= -2J(N - 2m). \end{aligned} \quad (11.69)$$

Setzen wir nun (11.68) und (11.69) in die detailed balance Bedingung (11.67) ein, finden wir

$$e^{-2\beta J(N-2m)} = \frac{p_B(i)}{p_B(j)} = \frac{V_{ji}}{V_{ij}} = (1 - p_c)^{N-2m}.$$

Auflösen nach p_c zeigt, dass wir detailed balance mit der Wahl

$$\boxed{p_c = 1 - e^{-2\beta J}} \quad (11.70)$$

immer erfüllen!

Wir schließen mit einigen **Bemerkungen** zum Wolff-Algorithmus

- Der Parameter p_c enthält die Temperatur: Bei hohen T ist p_c klein und der zu flippende Cluster viel kleiner als die eigentliche magnetische Domäne. Dies erfasst genau den Effekt thermischer Fluktuationen, die Spins unabhängig flippen lassen.
- Der Wolff-Algorithmus ist ein Beispiel eines **nicht-Metropolis** MC-Algorithmus mit Markov-Sampling, der natürlich trotzdem detailed balance erfüllt.
- Der dynamische Exponent ist $z_{\text{Wolff}} \approx 0,15$, also **viel** kleiner als beim Einzelspin-Flip. Damit wird die Cluster-Dynamik auch viel schneller an T_c und zeigt kaum critical slowing down.
- Die Cluster-Dynamik ist natürlich völlig “unphysikalische” und ausschließlich zur schnellen Berechnung von Gleichgewichtsmittelwerten gemacht und geeignet.

11.7 Literaturverzeichnis Kapitel 11

- [1] D. Frenkel und B. Smit. *Understanding Molecular Simulation*. 2nd. Orlando, FL, USA: Academic Press, Inc., 2001.
- [2] D. P. Landau und K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2005.
- [3] W. Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Statistical, Computational, and Theoretical Physics. Oxford University Press, 2006.
- [4] W. Kinzel und G. Reents. *Physics by Computer*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
- [5] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [6] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.

- [7] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [8] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [9] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller und E. Teller. *Equation of State Calculations by Fast Computing Machines*. J. Chem. Phys. **21** (1953), 1087–1092.
- [11] E. Ising. *Beitrag zur Theorie des Ferromagnetismus*. Z. Phys. **31** (1925), 253–258.
- [12] R. J. Glauber. *Time-Dependent Statistics of the Ising Model*. J. Math. Phys. **4** (1963), 294.
- [13] K. Binder. *Finite Size Scaling Analysis of Ising Model Block Distribution Functions*. Z. Phys. B: Condens. Matter **43** (1981), 119.
- [14] R. Swendsen und J.-S. Wang. *Nonuniversal critical dynamics in Monte Carlo simulations*. Phys. Rev. Lett. **58** (Jan. 1987), 86–88.
- [15] U. Wolff. *Collective Monte Carlo Updating for Spin Systems*. Phys. Rev. Lett. **62** (Jan. 1989), 361–364.

11.8 Übungen Kapitel 11

1. Diskreter Random Walk

Wir wollen einen “Random Walk” auf einem 2-dimensionalen Quadratgitter $\vec{r}_{nm} = n\vec{e}_x + m\vec{e}_y$ mit diskreten Zeitschritten $t = 0, 1, 2, \dots$ simulieren: In jedem Zeitschritt springt der Random Walker von $\vec{r}(t)$ mit gleicher Wahrscheinlichkeit auf einen benachbarten Gitterplatz.

- a) Wir starten im Ursprung bei $n = m = 0$. Wie können Sie das zufällige Auswählen eines der vier Nachbarn mit Hilfe einer gleichverteilten Zufallszahl aus $[0, 1]$ realisieren? Simulieren Sie 1000 (oder mehr) Random Walks und berechnen Sie $\langle \vec{r}(t) \rangle$ und $\langle r^2(t) \rangle$ für $t = 10, 50, 100, 500, 1000$. Dabei bedeutet $\langle \dots \rangle$ eine Mittelung über die verschiedenen Random Walks.
- b) Welchem Gesetz folgt die t-Abhängigkeit in a)? Können Sie ihr Resultat aus a) analytisch begründen?
- c) Berechnen Sie numerisch für $t = 10, 50, 100, 500, 1000$ auch die Aufenthaltswahrscheinlichkeit $P(\vec{r}, t)$, dass der Random Walker zur Zeit t am Ort \vec{r} zu finden ist. Vergleichen Sie ihr Ergebnis mit der Lösung der Diffusionsgleichung für eine Diffusionskonstante $D = 1/2$.

2. Monte-Carlo Integration

Wir berechnen Integrale durch Monte-Carlo Integration, d.h. indem wir die Integrale als Mittelwert bezüglich einer Zufallsvariablen x mit Verteilung $p(x)$ umschreiben,

$$\langle f \rangle = \int dx p(x) f(x) = \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (11.71)$$

und das Integral durch N-maliges ziehen einer Zufallszahl aus der Verteilung $p(x)$ berechnen. Wir wollen hier $f(x)$ so wählen, dass wir Standard-Zufallszahlgeneratoren für $p(x)$ verwenden können, die Gleichverteilungen $p(x) = 1$ im Intervall $x \in [0, 1[$ generieren.

- a) Das klassische Beispiel ist die Berechnung von π :

$$\int_{|\vec{r}| < 1} d^2\vec{r} \quad (11.72)$$

Ziehen Sie N in $[0, 1]^2$ gleichverteilte Zufallszahlpaare (x_i, y_i) und zählen Sie, wie oft $x_i^2 + y_i^2 < 1$ gilt. Wie berechnet sich daraus das Integral (11.72) und welcher Wahl von $f(x)$ entspricht dieses Vorgehen.

- b) Sie kennen das in a) gesuchte Ergebnis exakt. Berechnen Sie den Fehler als Funktion von N für $N = 10^k$ mit $k = 1, \dots, 6$ und plotten Sie ihr Ergebnis doppelt-logarithmisch. Welche N -Abhängigkeit sollten Sie finden?

Berechnen Sie 1000mal das Integral (11.72) mit $N = 1000$ und plotten Sie ein Histogramm der Verteilung der Ergebnisse. Wie sollte die Verteilung aussehen?

- c) Schreiben Sie eine Monte-Carlo Integrationsroutine zur Berechnung des Flächeninhalts einer Ellipse

$$\int_{(x/a)^2 + (y/b)^2 < 1} dx dy \quad (11.73)$$

als Funktion der Parameter a und b .

- d) Erweitern Sie die Routine auf die Monte-Carlo Integration eines Integrals

$$\int_{(x/a)^2 + (y/b)^2 < 1} f(x, y) dx dy \quad (11.74)$$

einer Funktion $f(x, y)$ über einer Ellipse. Berechnen Sie damit

$$\int_{x^2/2+y^2<1} dx dy e^{-x^2} \quad (11.75)$$

(Kontrolle: 2.993).

3. Einfaches Sampling, Importance-Sampling, Markov-Sampling eines Integrals

Wir berechnen das Integral

$$I = \int_0^\infty dx e^{-\beta x} x^2$$

auf drei Arten:

- a) Einfaches Sampling gleichverteilter Stützstellen $x_i \in [0, \infty[$.
- b) Importance-Sampling der Verteilung $p(x) = \beta e^{-\beta x}$. Samplen Sie diese Verteilung direkt mit Hilfe der Transformationsmethode, siehe Gl. (10.8). Das Integral ergibt sich als Mittelwert einer
- c) Markov-Importance-Sampling der gleichen Verteilung $p(x) = \beta e^{-\beta x}$ mit Hilfe des Metropolis-Algorithmus.

Vergleichen Sie Ergebnisse und Konvergenzgeschwindigkeit der drei Methoden für $\beta = 0.01, 1, 100$.

4. Metropolis Sampling einer Gaußverteilung

Generieren Sie gaußverteilte Zufallszahlen mit Hilfe einer Metropolis MC-Simulation eines einzelnen Teilchens, dass sich in einem Federpotential $V(x) = \frac{1}{2}kx^2$ thermisch bewegt. Wie hängen die Temperatur $k_B T$ des Bades und die Varianz der gesampleten Gaußverteilung zusammen?

5. Monte-Carlo Simulation eines einzelnen Spins

Simulieren Sie mit Hilfe des Metropolis-Algorithmus einen einzelnen Spin $s = \{+1, -1\}$ im äußeren Magnetfeld H mit Energie

$$\mathcal{H} = -sH \quad (11.76)$$

Bieten Sie im Metropolis-Algorithmus Spin-Flips an.

- a) Berechnen Sie die Magnetisierung

$$m = \langle s \rangle \quad (11.77)$$

als Funktion des Magnetfeldes H bei Temperatur $k_B T = 1$, d.h. führen Sie Monte-Carlo Simulationen für verschiedene H durch.

- b) Führen Sie die entsprechende analytische Rechnung durch und vergleichen Sie Ihre Ergebnisse.
- c) Statt Markov- und Importance-Sampling mit dem Metropolis-Algorithmus können Sie einen einzelnen Spin natürlich auch direkt samplen (d.h. direkt Werte $s = \pm 1$ ziehen), und zwar mit und ohne Importance-Sampling mit der Boltzmann-Verteilung. Realisieren Sie auch diese beiden Varianten der Simulation.

6. Monte-Carlo Simulation des 2-dimensionalen Ising-Modells

Simulieren Sie mit Hilfe des Metropolis-Algorithmus das zweidimensionale Ising-Modell (bei $H = 0$, d.h. ohne Magnetfeld),

$$\mathcal{H} = -J \sum_{i,j \text{ n.N.}} s_i s_j. \quad (11.78)$$

Setzen Sie $J = 1$. Verwenden Sie ein Quadratgitter, das mindestens die Größe $N = 10^2$ (besser $N = 100^2$) haben sollte. Sie können bei der Implementierung eine der folgenden Randbedingungen

wählen oder auch mehrere ausprobieren: a) periodische Randbedingungen, b) alle Randspins zeigen fest in eine Richtung, c) den Randspins fehlt ein nächster Nachbar und damit eine Kopplung in (11.78).

Bieten sie im Metropolis-Algorithmus Spin-Flips zufällig ausgewählter Spins an. Wählen Sie als Anfangsbedingungen zufällige Spins oder völlig geordnete Spins. Nach einer Aufwärmphase sollten Sie versuchen $10^3 - 10^4$ MC-Sweeps durchzuführen, in denen im Mittel jedem Spin einmal ein Flip angeboten wird.

- a) Generieren Sie graphische Momentaufnahmen des Systems.
- b) Untersuchen Sie zuerst die Äquilibrierungsphase. Wählen Sie dazu als Anfangsbedingungen (i) zufällig ausgerichtete Spins oder (ii) völlig geordnete Spins und messen Sie die mittlere Energie pro Spin $e = E/N = \langle \mathcal{H} \rangle / N$ als Funktion der Simulationszeit $e = e(t)$. Wie lange müssen Sie warten, bis das Ergebnis unabhängig von den Anfangsbedingungen wird (für $k_B T = 1, 2.25, 3$)?
- c) Berechnen Sie die Mittelwerte der Energie $e = E/N\langle \mathcal{H} \rangle / N$, die Magnetisierung $\langle m \rangle = \langle (\sum_i s_i) / N \rangle$ bzw. den gemittelten Betrag der Magnetisierung $\langle |m| \rangle = \langle |\sum_i s_i| / N \rangle$ pro Spin für verschiedene Temperaturen $k_B T$, mindestens für $k_B T = 1, 2.25, 3$. Die kritische Temperatur (im thermodynamischen Limes) ist $k_B T_c = 2/\ln(1 + \sqrt{2}) \simeq 2.27$. Wie unterscheidet sich das Verhalten oberhalb und unterhalb von T_c ? Wie verhält sich die Magnetisierung $m = m(t)$ als Funktion der Simulationszeit bei den verschiedenen Temperaturen?
- d) Berechnen Sie die spezifische Wärme pro Spin (aus den Energiefluktuationen) für die gleichen Temperaturen wie in b).
- e) Bestimmen Sie die Binder-Kumulante

$$U_L(T) = 1 - \frac{\langle m^4 \rangle}{3\langle m^2 \rangle^2}$$

im Temperaturbereich $T = 1, \dots, 5$ für verschiedene Systemgrößen $N = L \times L$ (z.B. $L = 5, 10, 20$). Die Kurven $U_L(T)$ sollten sich alle bei der kritischen Temperatur $T = T_c$ schneiden. Welchen Wert erhalten Sie für T_c ? (exakter Wert siehe oben)

12 Perkolation

Literatur zu diesem Teil:

Ein Standardwerk ist das Buch von Stauffer [1] (oder auch [2]). Eine gute Einführung ist auch im Schwabl [3] zu finden. Das Material zum Potts-Modell stammt aus [4] und [5].

12.1 Site- und Bond-Perkolation

Wir definieren das Perkolationsproblem in seinen zwei Ausprägungen Bond- und Site-Perkolation.

Bei der **Perkolation** (von lat. *percolare* = durchsickern lassen, auswaschen motiviert durch Flüssigkeiten in porösen Medien) betrachten wir **Gitter** (Quadrat, Dreieck, kubisch) in D Dimensionen mit N **Gitterplätzen (Sites)** und N_b **Bonds** (Kanten) zwischen nächsten Nachbarn. Ein kubisches Gitter der **Größe** L hat $N = L^D$ Sites. Die Zahl N der Sites und N_b der Bonds hängt über $N_b = \frac{z}{2}N$ zusammen, wobei z wieder die **Koordinationszahl**, d.h. die Zahl der nächsten Nachbarn ist. Für kubische Gitter ist $z = 2D$.

12.1.1 Site-Perkolation

Bei der **Site-Perkolation (Knotenperkolation, Platzperkolation)** ist jeder Gitterplatz entweder besetzt oder unbesetzt. Dabei werden Gitterplätze *zufällig* mit der Wahrscheinlichkeit p besetzt. Dadurch entstehen **Cluster** von besetzten Plätzen, die durch nächste Nachbar Bonds verbunden sind, siehe Abb. 12.1.

Offensichtlich wächst die Clustergröße an, wenn p wächst. Bei der Perkolation stellt man die Frage, ob es einen **perkolierenden Cluster** gibt, d.h. einen Cluster, der von einem Ende des Systems (z.B. oben oder links) zum anderen (z.B. unten oder rechts) reicht. Im thermodynamischen Limes $N, L \rightarrow \infty$ ist ein perkolierender Cluster gleichbedeutend mit einem **unendlich großen Cluster**.

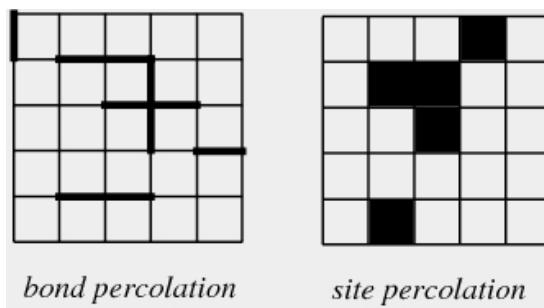


Abbildung 12.1: Bond- und Site-Perkolation.

12.1.2 Bond-Perkolation

Bei der **Bond-Perkolation** (**Kantenperkolation**, **Bindungsperkolation**) ist jeder Gitterbond entweder besetzt oder unbesetzt. Dabei werden Bonds zufällig mit einer Wahrscheinlichkeit p besetzt. Auch hier entstehen **Cluster** von Plätzen, die durch besetzte Bonds verbunden sind, siehe Abb. 12.1.

12.1.3 Geschichte und Anwendungen

Die Geschichte der Perkolation begann um 1940 mit Arbeiten von Flory und Stockmayer zu Makromolekülen wie Gummi, die sich durch chemische Vernetzung (bei Gummi Vulkanisierung durch Schwefelbrücken) von kleineren Molekülen bilden. Bei solchen Gelationsprozessen kommt es bei zunehmender Vernetzung irgendwann zu einem sogenannten **sol-gel Übergang**, wo aus einer Lösung kleiner Moleküle ein großes zusammenhängendes Makromolekül mit neuen mechanischen Eigenschaften wird. Dieser Übergang kann als eine Art Perkolationsübergang verstanden werden.



Abbildung 12.2: Links: Paul John Flory (1910-1985), amerikanischer Chemiker. Nobelpreis 1974 “for his fundamental achievements, both theoretical and experimental, in the physical chemistry of macromolecules”. Mitte: Walter H. Stockmayer (1914-2004), amerikanischer Chemiker. Rechts: Sol-gel Übergang.

Die Bezeichnung “Perkolation” wurde zum erstenmal 1957 von Broadbent und Hammersley verwendet, die **Flüssigkeiten in einem porösem Medium** untersucht haben. Der zugängliche Porenraum lässt sich als Cluster von Plätzen deuten, und die Flüssigkeit “perkoliert” (sickert durch), wenn ein perkolierender Cluster existiert. Broadbent und Hammersley erkannten auch als erste, dass die Perkolation eine ideale Anwendung für Computer darstellt.

Im Laufe der Zeit haben sich viele andere Anwendungen ergeben. Perkolation spielt eine wichtige Rolle bei der Beschreibung **“verdünnter” (oder “ungeordneter”) Magneten**, d.h. einem Gitter von wechselwirkenden Spins, wo nicht auf allen Gitterplätzen Spins sind, sondern Gitterplätze mit einer Wahrscheinlichkeit p leer bleiben. Andere Anwendungen sind Modelle für eher biologische Ausbreitungsprozesse, z.B. die Ausbreitung von **Waldbränden**: Damit ein Waldbrand sich ausbreiten kann, muss ein perkolierender Baumbestand existieren. Mit ganz ähnlichen Modellen lässt sich auch die Ausbreitung von **Epidemien** durch Ansteckung beschreiben. Bei diesen Ausbreitungsprozessen stehen natürlich nicht nur geometrische, sondern auch dynamische Fragestellungen im Vordergrund.

12.2 Perkolation als Phasenübergang

Wir zeigen, dass Perkolation als geometrischer kontinuierlicher Phasenübergang an der Perkolationsschwelle aufgefasst werden kann und definieren kritische Exponenten.

12.2.1 Perkolationsschwelle

Man kann zeigen, dass eine kritische Wahrscheinlichkeit p_c existiert, so dass für $p > p_c$ Perkolation vorliegt, d.h. es existiert ein unendlich großer Cluster. Dies kann auch mathematisch streng bewiesen werden. Der kritische Wert p_c wird auch als **Perkolationsschwelle** bezeichnet.

Eine wichtige Frage bei der Perkolation ist dann die exakte Bestimmung dieser Perkolationsschwelle p_c . Einige wichtige Ergebnisse dazu sind in Tabelle 12.1 zusammengestellt.

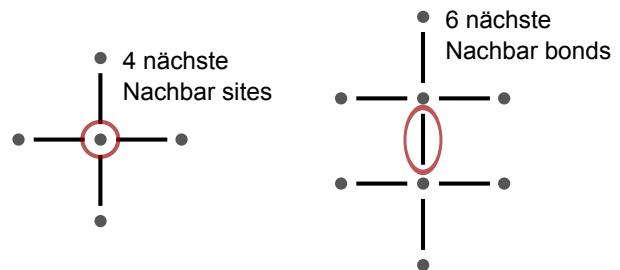
Dimension	Gitter	Site	Bond
D=1		1	1
D=2	Quadrat	0.593	1/2
	Dreieck	1/2	0.347=2 sin($\pi/18$)
D=3	kubisch	0.312	0.249
	BCC	0.246	0.180
D=4	kubisch	0.197	0.160

Tabelle 12.1: Perkolationsschwellen p_c für Site- und Bondperkolation auf verschiedenen Gittern und in verschiedenen Dimensionen.

Mehrere Dinge fallen auf an Tabelle 12.1:

- $p_c = p_c(D)$ ist fallend als Funktion der Dimension D , weil die Koordinationszahl $z = z(D)$ wächst als Funktion von D , wie man z.B. an den kubischen Gittern mit $z(D) = 2D$ sieht. Mehr nächste Nachbarn bedeuten aber mehr potentielle Möglichkeiten einen Cluster zu verbinden und daher eine kleinere Perkolationsschwelle p_c .
- Aus dem gleichen Grund gilt auch $p_{c,\text{site}} > p_{c,\text{bond}}$, wenn man Perkolationsschwellen von Site- und Bond-Perkolation vergleicht. Die Zahl der nächste Nachbar Sites ist immer kleiner als die Zahl der nächste Nachbar Bonds.

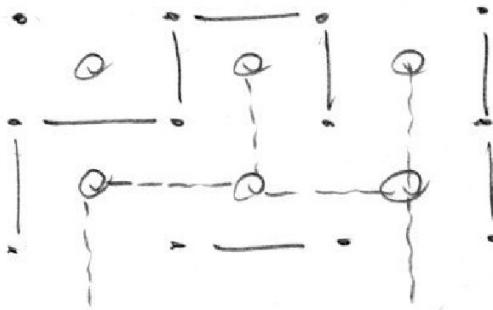
Dies sieht man z.B. auch an einem Quadratgitter in $D = 2$, wo $z_{\text{site}} = 2D = 4$ gilt, aber $z_{\text{bond}} = 6$ gilt.



Das Ergebnis

$$p_{c,\text{bond}} = 1/2 \text{ für } D = 2 \text{ Quadratgitter}$$

konnte mathematisch streng von Kesten 1980 gezeigt werden. Wir wollen hier kurz das Argument skizzieren.



Dazu führen wir das **duale Gitter** der Quadratmittelpunkte ein. Das duale Gitter ist für ein Quadratgitter identisch zum ursprünglichen Gitter. Zu jeder Bondkonfiguration im ursprünglichen Gitter definieren wir eine Bondkonfiguration im dualen Gitter durch die Regel, dass wir einen Bond im dualen Gitter setzen, wenn er *keinen* Bond im ursprünglichen Gitter schneidet. Dies führt zum Setzen von Bonds mit Wahrscheinlichkeit

$$p_d = 1 - p,$$

da dann jeder Bond in einem der Gitter belegt ist per Konstruktion.

Kesten konnte nun zeigen:

$$p < p_c \iff p_d > p_{d,c}.$$

Wenn das ursprüngliche Gitter unterhalb der Perkolationsschwelle ist, gibt es einen perkolierenden Cluster aus unbesetzten Bonds, so dass dieser Cluster auch im dualen Gitter perkoliert (dieses einleuchtende Resultat mathematisch streng zu zeigen, ist nicht einfach und die Leistung Kestens). Dies impliziert, dass wenn für das ursprüngliche Gitter $p = p_c$ gilt, auch für das duale Gitter genau $p_d = p_{d,c}$ gelten muss. Daher gilt auch $p_{d,c} = 1 - p_c$ für die Perkolationsschwellen. Da das quadratische Gitter **selbstdual** ist, gilt aber auch $p_c = p_{d,c}$. Beide Gleichungen zusammen implizieren

$$p_c = p_{d,c} = 1/2.$$

12.2.2 Cluster-Observablen und kritische Exponenten

Wir wollen nun einige wichtige Cluster-Observablen definieren, die es erlauben werden, den Übergang in die perkolierende Phase zu charakterisieren.

Wir beginnen mit

$P_\infty(x) \equiv$ Wahrscheinlichkeit, dass ein besetzter Site/Bond x zum unendlichen Cluster gehört.

(12.1)

(In einigen Büchern wird stattdessen eine Wahrscheinlichkeit M_∞ betrachtet, dass ein Site/Bond x zum unendlichen Cluster gehört – unabhängig davon, ob er besetzt ist oder nicht. Beide Definitionen unterscheiden sich nur um einen Faktor p , $M_\infty = pP_\infty$.) Diese Größe sollte im thermodynamischen Limes translationsinvariant und damit unabhängig von x werden. P_∞ charakterisiert den **mittleren Anteil des unendlichen Clusters am besetzten Teil des Gitters**, $M_\infty = pP_\infty$ ist der mittlere Anteil des unendlichen Clusters am gesamten Gitter. Man findet für P_∞ (und M_∞) folgendes Verhalten:

$$P_\infty \propto \begin{cases} 0 & p < p_c \text{ (kein unendl. Cluster)} \\ (p - p_c)^\beta & p > p_c \end{cases} \quad (12.2)$$

Das Verhalten an der Perkolationsschwelle definiert einen **kritischen Exponenten** β . Die Bezeichnung ist die gleiche wie für den kritischen Exponenten der Magnetisierung im Ising-Modell; der Grund dafür ist eine Analogie, die durch eine Abbildung auf das Potts-Modell zustande kommt und in Kapitel 12.4 diskutiert wird.

Eine weitere wichtige Observable ist die **mittlere Clustermasse** der endlichen Cluster, die man als

$$\boxed{\begin{aligned} S &\equiv \text{mittlere Zahl von Plätzen in einem endlichen Cluster} \\ &\propto |p - p_c|^{-\gamma} \end{aligned}} \quad (12.3)$$

definieren kann. Man beachte, dass S so definiert ist, dass der unendliche Cluster für $p > p_c$ nicht berücksichtigt wird. An der Perkolationsschwelle divergiert S dann mit einem Exponenten γ . Diese Bezeichnung ist die gleiche wie der Exponent der Suszeptibilität eines magnetischen Systems.

Neben der Clustermasse interessiert noch die **mittlere Clusterausdehnung**, die durch eine **Korrelationslänge** ξ gemessen wird:

$$\boxed{\begin{aligned} \xi &= \text{mittlerer Abstand zweier Punkte im endlichen Cluster} \\ &\propto |p - p_c|^{-\nu}. \end{aligned}} \quad (12.4)$$

Auch hier werden nur Beiträge von endlichen Clustern berücksichtigt. Der Exponent trägt die gleiche Bezeichnung wie der Korrelationslängenexponent im Ising-Modell.

Die Korrelationslänge ergibt sich aus der **Korrelations- oder Paarfunktion**

$$\boxed{\begin{aligned} g(r) &\equiv \text{Wahrscheinlichkeit, dass Punkt im Abstand } r \text{ von besetztem Punkt} \\ &\quad \text{zum selben endlichen Cluster gehört} \\ &\propto e^{-r/\xi} \quad (\text{für } p \neq p_c). \end{aligned}} \quad (12.5)$$

Diese Korrelationsfunktion fällt exponentiell ab für $p \neq p_c$ und die Zerfallslänge definiert die Korrelationslänge.

Wir sehen also:

- Charakteristische Clustergrößen gehorchen Skalengesetzen mit kritischen Exponenten wie bei einem kontinuierlichen Phasenübergang.
- Die Exponenten β , ν und γ hängen tatsächlich *nur* von der Dimension D des Gitters ab und nicht davon, ob wir Site- oder Bond-Perkolation betrachten oder von der Art des Gitters (z.B. Quadrat oder Dreieck). Dies ist Ausdruck einer **Universalität** an der Perkolationsschwelle.
- Für $D = 2$ sind die Exponenten beispielsweise exakt bekannt: $\beta = 5/36$, $\nu = 4/3$ und $\gamma = 43/18$ und *verschieden* von den Exponenten des Ising-Modells ($\beta = 1/8$, $\nu = 1$, siehe Tabelle in Kapitel 11.5). Ising-Modell und Perkolation liegen daher in verschiedenen Universalitätsklassen.

Insgesamt stellen wir fest:

Perkolation kann als geometrischer kontinuierlicher Phasenübergang aufgefasst werden, also als kritischer Punkt bei $p = p_c$.

12.3 Perkolation in D=1

Wir definieren Clusterzahlen zur Charakterisierung der Clusterverteilung. Dann leiten wir einige exakte Resultate für den Fall der Perkolation in einer Raumdimension $D = 1$ her.

12.3.1 Clusterzahlen

Bevor wir uns dem Spezialfall $D = 1$ zuwenden, führen wir noch Clusterzahlen für Site-Perkolation ein, die allgemein gelten. Viele der oben eingeführten Perkolationsobservablen lassen sich durch die

Clusterzahlen n_s für endliche **s-Cluster**, d.h. Cluster mit $s (< \infty)$ Plätzen, ausdrücken:

$$n_s \equiv \frac{\text{Anzahl der s-Cluster}}{N}. \quad (12.6)$$

Wir verwenden also auf die Gesamtplatzzahl normierte Clusterzahlen.

Dann gilt

$$p = \sum_{s=1}^{\infty} sn_s = \frac{\text{Anzahl besetzter Plätze (in endl. Clustern)}}{N} \quad \text{für } p < p_c. \quad (12.7)$$

Für $p > p_c$ ist $p > \sum_{s=1}^{\infty} sn_s$, da sich dann ein endlicher Anteil von Plätzen im unendlichen Cluster befindet. Die normierte **Gesamtzahl an endlichen Clustern** n_c ist

$$n_c = \sum_{s=1}^{\infty} n_s = \frac{\text{Anzahl endl. Cluster}}{N}. \quad (12.8)$$

Da jeder Platz (i) zu einem unendlichen oder (ii) zu einem endlichen Cluster gehört oder (iii) gar nicht besetzt ist, gilt

$$N = \underbrace{NpP_{\infty}}_{\infty \text{ Cluster}} + \underbrace{N \sum_{s=1}^{\infty} sn_s}_{\text{endl. Cluster}} + \underbrace{N(1-p)}_{\text{unbesetzt}} \quad (12.9)$$

(dabei ist pP_{∞} die Wahrscheinlichkeit, dass ein Platz besetzt ist *und* zum unendlichen Cluster gehört). Damit lässt sich P_{∞} durch die Clusterzahlen ausdrücken:

$$P_{\infty} = 1 - \frac{1}{p} \sum_{s=1}^{\infty} sn_s. \quad (12.10)$$

Schließlich wollen wir noch die mittlere Clustermasse S der endlichen Cluster aus (12.3) durch die Clusterzahlen n_s ausdrücken. Dazu wählen wir zufällig einen besetzten Platz P in einem endlichen Cluster und wollen die Wahrscheinlichkeit p_s berechnen, dass P zu einem s-Cluster gehört. Dazu benötigen wir (i) die Wahrscheinlichkeit sn_s , dass ein *beliebiger* (besetzter oder unbesetzter) Platz zu einem s-Cluster gehört und (ii) die Wahrscheinlichkeit $\sum_{s=1}^{\infty} sn_s$, dass ein beliebiger Platz zu einem endlichen Cluster gehört. Dann gilt $sn_s = p_s(\sum_{s=1}^{\infty} sn_s)$, also

$$p_s = \text{Wahrscheinlichkeit, dass P zu einem s-Cluster gehört} = \frac{sn_s}{\sum_{s=1}^{\infty} sn_s}. \quad (12.11)$$

Mit der Wahrscheinlichkeit p_s bilden wir nun die mittlere Clustermasse

$$\begin{aligned} S &= \text{mittlere Größe des endlichen Clusters, zu dem P gehört} \\ &= \sum_{s=1}^{\infty} sp_s = \frac{\sum_{s=1}^{\infty} s^2 n_s}{\sum_{s=1}^{\infty} sn_s} \quad \text{für } \underline{p < p_c} \quad \frac{1}{p} \sum_{s=1}^{\infty} sn_s. \end{aligned} \quad (12.12)$$

12.3.2 Perkolation in D=1

Für Site-Perkolation in einer Dimension $D = 1$ lassen sich einige Exponenten exakt berechnen. In $D = 1$ sollte allerdings einfach $p_c = 1$ gelten, da auch nur ein einziger unbesetzter Platz einen unendlichen perkolierenden Cluster zerreißen.

Wir betrachten ein eindimensionales ‘‘Gitter’’ mit L Plätzen. Die Wahrscheinlichkeit, dass ein beliebiger Punkt dieses Gitters ein linker Endpunkt eines s -Clusters ist, beträgt $p^s(1-p)^2$ und ergibt sich aus den Wahrscheinlichkeiten, dass genau s Plätze (nach rechts und einschließlich des gewählten Gitterpunktes) jeweils mit Wahrscheinlichkeit p besetzt sein müssen und die beiden Enden des Clusters jeweils unbesetzt. Durch die Fixierung auf den linken Rand vermeiden wir zusätzliche kombinatorische Faktoren. Da jeder Punkt des Gitters als linker Rand fungieren kann, ist die Gesamtzahl an s -Clustern $Lp^s(1-p)^2$ und damit

$$n_s = \frac{\text{Anzahl der } s\text{-Cluster}}{L} = p^s(1-p)^2. \quad (12.13)$$

Dieses Resultat können wir nun nutzen, um P_∞ und S nach (12.10) bzw. (12.12) zu berechnen. Dabei benötigen wir

$$\sum_{s=1}^{\infty} sn_s = (1-p)^2 \sum_{s=1}^{\infty} sp^s = (1-p)^2(p\partial_p) \frac{p}{1-p} = p,$$

was für alle $p < 1$ gilt, und erhalten nach (12.10)

$$P_\infty = 1 - \frac{p}{p} = 0 \quad \text{für } p < 1.$$

Für $p = 1$ gilt natürlich $P_\infty = 1$. Das heißt aber nach (12.2), dass

$$p_c(D = 1) = 1$$

wie wir das erwartet haben. Außerdem stellen wir fest, dass der Exponent β unter diesen Umständen nicht wohldefiniert ist.

Wir fahren fort und berechnen die mittlere Clustermasse S nach (12.12):

$$S = \frac{1}{p} \sum_{s=1}^{\infty} s^2 n_s = \frac{1}{p} (1-p)^2 \sum_{s=1}^{\infty} s^2 p^s,$$

was nach einigen Umformungen mit geometrischen Reihen schließlich auf

$$S = \frac{1+p}{1-p} \propto (1-p)^{-1}$$

führt. Das bedeutet, dass S in der Tat an $p_c = 1$ divergiert, und wir lesen gemäß (12.3)

$$\gamma = 1$$

ab.

Weiter können wir auch sofort

$$g(r) = 2p^r$$

angeben in $D = 1$. Dies ist die Wahrscheinlichkeit nach links oder nach rechts (Faktor 2) r besetzte Plätze zu finden. Daraus lässt sich dann auch die Korrelationslänge ξ berechnen

$$\xi = \frac{\sum_{r=1}^{\infty} rg(r)}{\sum_{r=1}^{\infty} g(r)} = \frac{\sum_{r=1}^{\infty} rp^r}{\sum_{r=1}^{\infty} p^r} = \frac{1}{1-p}.$$

Das bedeutet, dass auch ξ in der Tat bei $p_c = 1$ divergiert und wir lesen nach (12.4)

$$\nu = 1$$

ab.

Das heißt wir können für den einfachen Fall $D = 1$ in der Tat die oben postulierten Skalengesetze explizit berechnen und Exponenten angeben.

12.4 Potts-Modell und Perkolation

Wir zeigen, dass das Q -Zustands Potts-Modell im Limes $Q \rightarrow 1$ äquivalent zum Perkolationsproblem ist. Diese Abbildung liefert auch die Idee zu Cluster-Algorithmen für das Ising-Modell. Mit Hilfe der Mean-Field Theorie des Potts-Modells leiten wir dann einige Resultate zum Perkolationsübergang her.

In diesem (etwas technischen) Kapitel werden wir eine Verbindung zwischen dem Perkolationsproblem und dem sogenannten Q -Zustands Potts-Modell im Limes $Q \rightarrow 1$ aufzeigen. Dies wird eine Verbindung herstellen zwischen dem geometrischen Perkolationsübergang und den thermischen Phasenübergängen, wie wir sie bereits aus dem mit dem Potts-Modell eng verwandten Ising-Modell kennen. Dies zeigt auch wieder, dass der Perkolationsübergang als Phasenübergang zu verstehen ist und seine Analogie zu magnetischen Phasenübergängen.

12.4.1 Q-Zustands Potts-Modell

Das **Q-Zustands Potts-Modell** ist eine Verallgemeinerung des Ising-Modells auf Spin-Variablen s_i ($i = 1, \dots, N$ Gitterplatzindex), die nun Q Zustände $s_i \in \{1, \dots, Q\}$ annehmen können. Dabei wollen wir wieder magnetische Wechselwirkungen auf nächste Nachbarn, also Bonds, beschränken mit

$$\text{Energie eines Bonds } <ij> = \begin{cases} -QJ & s_i = s_j \\ 0 & s_i \neq s_j \end{cases}.$$

Außerdem führen wir ein Magnetfeld H ein, das in einer Art Zeeman-Wechselwirkung die Energie des Spins s_i absenkt, sofern sich dieser im Zustand $s_i = 1$ befindet, also

$$\text{Energie pro site } i = \begin{cases} -QH & s_i = 1 \\ 0 & s_i \neq 1 \end{cases}.$$

Mit diesen beiden Beiträgen ist die Gesamthamiltonfunktion des Potts-Modells definiert:

$$\mathcal{H}_{\text{Potts}} = - \sum_{\text{bonds } ij} JQ\delta_{s_i s_j} - \sum_{\text{sites } i} HQ\delta_{s_i, 1}. \quad (12.14)$$

Bei tiefen Temperaturen kann sich das Potts-Modell genau wie das Ising-Modell ordnen in einen der Q gleichberechtigten Spinzustände. Ist ein infinitesimal kleines Magnetfeld $H = 0^+$ angeschaltet, ordnet sich das Potts-Modell in den 1-Zustand. Man definiert dann als **Ordnungsparameter** eine **Magnetisierung** m , die diese Ordnung in den Zustand 1 misst:

$$m = \frac{\langle Q\delta_{s_i, 1} - 1 \rangle}{Q - 1} = \frac{Q\langle \delta_{s_i, 1} \rangle - 1}{Q - 1}. \quad (12.15)$$

Wir überzeugen uns, dass eine so definierte Magnetisierung alle Eigenschaften eines Ordnungsparameters hat: In der vollständig ungeordneten Phase sind alle Q Zustände gleich wahrscheinlich und wir erwarten $\langle \delta_{s_i, 1} \rangle = 1/Q$, was zu $M = 0$ führt. In der vollständig geordneten Phase sind alle Spins im Zustand 1, was auf $\langle \delta_{s_i, 1} \rangle = 1$ und damit $M = 1$ führt.

Wir überzeugen uns auch, dass der **Spezialfall** $Q = 2$ des Potts-Modells wieder das **Ising-Modell** (11.40) ergibt. Beim Ising-Modell hat jeder Spin $Q = 2$ Einstellungen mit einer Differenz in der Bondenergie von $QJ = 2J$ und einer Differenz in der Zeeman-Energie von $QH = 2H$ genau wie in (11.40).

12.4.2 Abbildung auf Perkolation im Limes $Q \rightarrow 1$

Im Folgenden sind wir am Limes $Q \rightarrow 1$ des Potts-Modells interessiert, für den wir zeigen wollen:

$$\lim_{Q \rightarrow 1} \text{Potts-Modell} \iff \text{Bond-Perkolation mit } p = 1 - e^{-\beta J}. \quad (12.16)$$

Dazu stellen wir zunächst fest, dass der Fall $Q = 1$ eigentlich trivial ist: Dann hat jeder Spin nur genau eine Einstellungsmöglichkeit und wir haben $\mathcal{H}_{\text{Potts}} = \text{const}$ und eine Zustandssumme über genau einen Zustand mit $Z_{\text{Potts}} = e^{-\beta \text{const}}$. Die Information über die Perkolation steckt also nicht im Wert bei genau $Q = 1$, sondern im führenden Term von $\ln Z_{\text{Potts}}$ bei einer *Entwicklung* um $Q = 1$.

Zunächst vereinfachen wir die Notation etwas:

$$\beta \mathcal{H}_{\text{Potts}} = - \sum_{\text{bonds } ij} K(\delta_{s_i s_j} - 1) - \sum_{\text{sites } i} L(\delta_{s_i,1} - 1) \quad (12.17)$$

mit $K \equiv QJ/k_B T$ und $L \equiv QH/k_B T$. Außerdem haben wir zwei konstante Terme abgezogen. Die Zustandssumme des Potts-Modells lautet dann

$$\begin{aligned} Z_{\text{Potts}} &= \sum_{s_1=1}^Q \sum_{s_2=1}^Q \dots \sum_{s_N=1}^Q e^{\sum_{ij} K(\delta_{s_i s_j} - 1) + \sum_{i=1}^N L(\delta_{s_i,1} - 1)} \\ &= \sum_{\{s\}} \left(\prod_{\text{bonds } ij} e^{K(\delta_{s_i s_j} - 1)} \right) \left(\prod_{i=1}^N e^{L(\delta_{s_i,1} - 1)} \right). \end{aligned}$$

Nun schreiben wir die exp-Funktionen etwas um. Wegen

$$\begin{aligned} e^{K(\delta_{s_i s_j} - 1)} &= \begin{cases} 1 & s_i = s_j \\ e^{-K} & s_i \neq s_j \end{cases} & e^{L(\delta_{s_i,1} - 1)} &= \begin{cases} 1 & s_i = 1 \\ e^{-L} & s_i \neq 1 \end{cases} \\ &= \delta_{s_i s_j} \underbrace{(1 - e^{-K})}_{\equiv p} + \underbrace{e^{-K}}_{\equiv 1 - p} & &= \delta_{s_i,1} (1 - e^{-L}) + e^{-L} \end{aligned}$$

gilt

$$Z_{\text{Potts}} = \sum_{\{s\}} \left(\prod_{\text{bonds } ij} [p\delta_{s_i s_j} + (1 - p)] \right) \left(\prod_{i=1}^N [\delta_{s_i,1} (1 - e^{-L}) + e^{-L}] \right).$$

Nun wollen wir das Produkt über die Bonds $<ij>$ ausmultiplizieren. Dabei entsteht eine Summe von Produkttermen, die man sich grafisch veranschaulichen kann: In jedem Produktterm ist auf jedem Bond *entweder* ein Faktor $p\delta_{s_i s_j}$ – dann wollen wir den Bond als “besetzt” bezeichnen – *oder* ein Faktor $(1 - p)$ – dann bezeichnen wir den Bond als “unbesetzt”. In jedem Produktterm ist jeder Bond entweder als besetzt oder unbesetzt zu kennzeichnen; dies ergibt eine **Bondkonfiguration** B ; ist ein Bond $<ij>$ in dieser Bondkonfiguration besetzt, schreiben wir $<ij> \in B$. Außerdem bezeichnen wir mit $|B|$ die Anzahl aller besetzten Bonds, wobei N_b die Gesamtzahl aller möglichen Bonds ist. In der Bondkonfiguration B sind also $|B|$ Bonds besetzt und $N_b - |B|$ Bonds unbesetzt. Wir können dann das Produkt $\prod_{\text{bonds } ij} [...]$ als Summe über alle möglichen Bondkonfigurationen B schreiben:

$$\prod_{\text{bonds } ij} [p\delta_{s_i s_j} + (1 - p)] = \sum_{\text{Bondkonfigurationen } B} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) p^{|B|} (1 - p)^{N_b - |B|}.$$

Die Wahrscheinlichkeiten $p^{|B|}(1-p)^{N_b-|B|}$ sind nun aber genau die Wahrscheinlichkeit, diese Bondkonfiguration B im Bond-Perkolationsproblem mit Wahrscheinlichkeit $p = 1 - e^{-K}$ zu finden. Daher lassen sich Summen

$$\sum_{\text{Bondkonfigurationen } B} \dots p^{|B|}(1-p)^{N_b-|B|} = \langle \dots \rangle_{\text{Perk}}$$

auch als **Mittelwerte über alle Bondkonfigurationen im Bond-Perkolationsproblem** auffassen. damit ist die Verbindung zum Perkolationsproblem hergestellt. Wir können dann die Zustandssumme weiter umschreiben:

$$\begin{aligned} Z_{\text{Potts}} &= \sum_{\{s\}} \sum_{\text{Bondkonf. } B} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) \left(\prod_{i=1}^N [\delta_{s_i,1}(1 - e^{-L}) + e^{-L}] \right) p^{|B|}(1-p)^{N_b-|B|} \\ &= \left\langle \sum_{\{s\}} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) \left(\prod_{i=1}^N [\delta_{s_i,1}(1 - e^{-L}) + e^{-L}] \right) \right\rangle_{\text{Perk}}. \end{aligned}$$

Um die multiple Spinsumme $\sum_{\{s\}} = \sum_{s_1} \sum_{s_2} \dots \sum_{s_N}$ unter der Perkolations-Mittelung zu berechnen, muss man sich den Effekt des Produkts der $\delta_{s_i s_j}$ -Funktionen auf den Bonds klarmachen: Bei einer Summation über alle Spins tragen auf Grund dieser $\delta_{s_i s_j}$ -Funktionen immer nur Konfigurationen bei, wo Spins die durch einen Bond B verbunden sind, den *gleichen* Zustand haben. Eine Gruppe von Spins die jeweils durch Bonds verbunden sind, bildet aber gerade einen zusammenhängenden **Cluster C** . Für alle Spins innerhalb eines Clusters C bleibt daher bei der multiplen Spinsummation $\sum_{\{s\}}$ nur *eine* freie Spinsummation übrig. Wir teilen die Bondkonfiguration B daher in $N_C(B)$ Cluster $C \in B$, die jeweils G_C Plätze umfassen. Innerhalb jedes Clusters stimmen dann die Spins überein ($s_i = s_C$ für $i \in C$) und wir behalten in der multiplen Spinsumme nur noch eine Summation $\sum_{s_C=1}^Q$ pro Cluster, also

$$\sum_{\{s\}} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) \dots = \left(\prod_{i=1}^N \sum_{s_i=1}^Q \right) \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) \dots = \left(\prod_{C \in B} \sum_{s_C=1}^Q \right) \dots$$

Es gilt dann:

$$\begin{aligned} \sum_{\{s\}} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) &= \prod_{C \in B} \left(\sum_{s=1}^Q 1 \right) = Q^{N_C(B)} \\ \sum_{\{s\}} \left(\prod_{\text{bonds } ij \in B} \delta_{s_i s_j} \right) \left(\prod_{i=1}^N [\delta_{s_i,1}(1 - e^{-L}) + e^{-L}] \right) &= \left(\prod_{C \in B} \sum_{s_C=1}^Q \right) \left(\prod_{C \in B} [\delta_{s_C,1}(1 - e^{-L}) + e^{-L}]^{G_C} \right) \\ &= \prod_{C \in B} \left(\sum_{s=1}^Q [\delta_{s,1}(1 - e^{-L}) + e^{-L}]^{G_C} \right) \\ &= \prod_{C \in B} (1 + (Q - 1)e^{-LG_C}). \end{aligned}$$

Damit erhalten wir das zentrale Zwischenergebnis

$$Z_{\text{Potts}} = \left\langle \prod_{C \in B} (1 + (Q - 1)e^{-LG_C}) \right\rangle_{\text{Perk}}, \quad (12.18)$$

wo die Bond-Perkolationsmittelwerte mit der Bondwahrscheinlichkeit $p = 1 - e^{-K} = 1 - e^{-\beta JQ}$ genommen werden.

Wenn wir genau $Q = 1$ setzen, erhalten wir wie erwartet ein triviales Ergebnis, da dann jeder Faktor im verbleibenden Produkt =1 ist, also auch $Z_{\text{Potts}}|_{Q=1} = 1$. Allerdings enthält der führende Term in einer Entwicklung um $Q = 1$ alle Information über das Bond-Perkolationsproblem:

$$\left. \frac{\partial}{\partial Q} \right|_{Q=1} Z_{\text{Potts}} = \left\langle \sum_{C \in B \text{ endl.}} e^{-LG_C} \right\rangle_{\text{Perk}}. \quad (12.19)$$

Für alle positiven Magnetfelder $L > 0$ tragen unendliche Cluster mit $G_C = \infty$ nicht bei in (12.19), da $e^{-LG_C} = 0$ in diesem Fall; daher muss nur über endliche Cluster C summiert werden. Wegen $Z_{\text{Potts}}|_{Q=1} = 1$ gilt auch

$$\left. \frac{\partial}{\partial Q} \right|_{Q=1} Z_{\text{Potts}} = \left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}} = - \left. \frac{\partial}{\partial Q} \right|_{Q=1} (\beta F_{\text{Potts}}),$$

d.h. (12.19) ist tatsächlich der führende Term in der Q-Entwicklung der freien Energie des Potts-Modells um $Q = 1$.

Die Funktion $\left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}}$ ist nun eine **generierende Funktion** für Perkolationseigenschaften: Sie generiert Momente der Clustergrößen G_C bei fortgesetzter Ableitung $\partial/\partial L|_{L=0}$, also bei Taylorentwicklung um $L = 0$:

$$\left. \frac{\partial}{\partial Q} \right|_{Q=1, L=0} \ln Z_{\text{Potts}} = \left\langle \sum_{C \text{ endl.}} 1 \right\rangle_{\text{Perk}} = \langle N_C \rangle_{\text{Perk}} = \text{mittlere Clusterzahl} \quad (12.20)$$

$$\left. \frac{\partial^n}{\partial L^n} \right|_{L=0} \left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}} = (-1)^n \left\langle \sum_{C \text{ endl.}} G_C^n \right\rangle_{\text{Perk}}. \quad (12.21)$$

Bei Bond-Perkolation sind Cluster mit $G_C = 1$ isolierte Punkte. In diesem Sinne gehören alle Gitterplätze zu einem Cluster, und es gilt

$$\sum_{C \text{ endl.}} G_C = N - NP_\infty.$$

Daher erlaubt das erste Clustergrößen-Moment, P_∞ zu berechnen, während das zweite Moment die mittlere Clustermasse S_P angibt (hier die mittlere Clustermasse des Clusters, zum dem ein fester Punkt P, z.B. der Ursprung gehört),

$$P_\infty = \frac{1}{N} \left(1 - \left\langle \sum_{C \text{ endl.}} G_C \right\rangle_{\text{Perk}} \right) = 1 + \frac{1}{N} \left. \frac{\partial}{\partial L} \right|_{L=0} \left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}} \quad (12.22)$$

$$S_P = \left\langle \sum_{C \text{ endl.}} G_C^2 \right\rangle_{\text{Perk}} = \left. \frac{\partial^2}{\partial L^2} \right|_{L=0} \left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}} \quad (12.23)$$

(P kann auf G_C Plätzen liegen in einem Cluster der Größe G_C , daher wird G_C^2 gemittelt).

Auf der anderen Seite generieren Ableitungen $\left. \frac{\partial^n}{\partial L^n} \right|_{L=0} \ln Z_{\text{Potts}}$ im Potts-Modell aber Momente der Magnetisierung. So gilt für die in (12.15) definierte Magnetisierung m

$$\frac{1}{N} \left. \frac{\partial}{\partial L} \right|_{L=0} \ln Z_{\text{Potts}} = \langle \delta_{s_i, 1} \rangle - 1 = (m - 1) \frac{Q - 1}{Q}.$$

Aus dieser Beziehung zusammen mit (12.22) ergibt sich dann ein direkter Zusammenhang zwischen P_∞ in der Perkolation und der Magnetisierung des Q-Zustands Potts-Modells im Limes $Q \rightarrow 1$:

$$\boxed{P_\infty = 1 + \frac{1}{N} \left. \frac{\partial}{\partial L} \right|_{L=0} \left. \frac{\partial}{\partial Q} \right|_{Q=1} \ln Z_{\text{Potts}} = 1 + \left. \frac{\partial}{\partial Q} \right|_{Q=1} ((m-1) \frac{Q-1}{Q}) = m|_{Q=1}. \quad (12.24)}$$

Dies rechtfertigt dann auch die Verwendung der gleichen Bezeichnung β für den Exponenten sowohl von P_∞ als auch der Magnetisierung. Eine ähnliche Beziehung wie in (12.24) zwischen P_∞ und m gilt für die mittlere Clustermasse S und die Suszeptibilität χ des Potts-Modells, die sich als zweites Moment der Magnetisierung schreiben lässt, im Limes $Q \rightarrow 1$.

Damit ist die Äquivalenz (12.16) zwischen Potts-Modell im Limes $Q \rightarrow 1$ und Bond-Perkolation mit $p = 1 - e^{-K} = 1 - e^{-QJ/k_B T} \xrightarrow{Q \rightarrow 1} 1 - e^{-\beta J}$ gezeigt.

Cluster-Algorithmen

Die Abbildung vom Potts-Modell auf das Perkolationsproblem ist auch Grundlage der Cluster-Simulationsalgorithmen für das Ising-Modell aus Kapitel 11.6. Wir wollen uns hier kurz klarmachen, wie diese Abbildung die Idee für Cluster-Algorithmen liefern kann. Dazu betrachten wir nochmal (12.18) für den Fall ohne Magnetfeld $H = L = 0$ und für den Fall des Ising-Modells mit $Q = 2$,

$$\boxed{Z_{\text{Ising}} = \left\langle 2^{N_C(B)} \right\rangle_{\text{Perk}} = \sum_{\text{Bondkonfigurationen } B} \left(\prod_{C \in B} \left(\sum_{s_C=1}^2 1 \right) \right) p^{|B|} (1-p)^{N_b - |B|}, \quad (12.25)}$$

wobei im Ising-Modell mit $Q = 2$

$$p = 1 - e^{-K} = 1 - e^{-2\beta J} = p_c$$

mit p_c wie im Wolff-Algorithmus, siehe (11.70). Die Form (12.25) der Zustandssumme besagt, dass wir statt Spinkonfigurationen zu samplen (wie im Einzelspin-Flip Metropolis-Algorithmus) auch Bondkonfigurationen B wie bei der Perkolation samplen können, d.h. in dem wir zufällig Bonds mit der Wahrscheinlichkeit p_c setzen. Dadurch entstehen Cluster. Dann erhalten wir die zugehörige Spinkonfiguration, indem wir innerhalb jedes Clusters die Spins s_i gleich setzen. Dies ist genau die Grundidee der Cluster-Algorithmen.

Durch zufällige Auswahl eines Clusters (durch Auswahl des ersten Spins und Aufbau eines Clusters um diesen Spin im Wolff-Algorithmus) und den Cluster-Flip aller Spins innerhalb des Clusters arbeiten wir dann genau den Faktor $\prod_{C \in B} (\sum_{s_C=1}^2 1) = 2^{N_C(B)}$ in (12.25) ab.

12.4.3 Mean-Field Theorie des Potts-Modells

Die Äquivalenz (12.16) zwischen Potts-Modell im Limes $Q \rightarrow 1$ und Bond-Perkolation mit $p = 1 - e^{-K} = e^{-\beta J}$ impliziert, dass wir auch kritische Exponenten, den Wert für p_c oder die Ordnung des Perkolationsüberganges aus den entsprechenden Eigenschaften des thermischen Phasenübergangs des Potts-Modells ableiten können. Dazu können wir dann auch Methoden benutzen, die wir in der Theorie der Phasenübergänge kennengelernt haben.

In diesem Kapitel wollen wir die **Bragg-Williams Mean-Field Theorie** des Potts-Modells untersuchen, um im Limes $Q \rightarrow 1$ einige Resultate über den Perkolationsübergang zu erzielen. Wir werden uns dabei auf den Fall $H = L = 0$ ohne Magnetfeld beschränken mit

$$\beta \mathcal{H}_{\text{Potts}} = - \sum_{\langle ij \rangle} K(\delta_{s_i s_j} - 1), \quad \text{mit } K = \frac{QJ}{k_B T}.$$

In der Mean-Field Theorie betrachten wir

$$x_s \equiv \text{Anteil der Spins mit } s_i = s = \langle \delta_{s_i, s} \rangle \quad (s = 1, \dots, Q).$$

Wir nehmen wieder o.B.d.A. an, dass sich das System in der Tieftemperaturphase in den Zustand 1 ordnet. Die Magnetisierung m hängt dann nach (12.15) mit x_1 zusammen,

$$m = \frac{Q\langle \delta_{s_i, 1} \rangle - 1}{Q - 1} = \frac{Qx_1 - 1}{Q - 1},$$

während alle anderen Zustände gleich wahrscheinlich sind,

$$x_2 = \dots = x_Q.$$

Außerdem gilt $\sum_{s=1}^Q x_s = 1$, so dass wir alle x_s durch m ausdrücken können:

$$x_1 = \frac{1}{Q}[1 + (Q - 1)m], \quad x_2 = \dots = x_Q = \frac{1}{Q}(1 - m). \quad (12.26)$$

In der Bragg-Williams Mean-Field Theorie betrachten wir die freie Energie $F = E - TS$ und schätzen E durch eine Mean-Field Näherung ab, während wir S exakt als Mischungsentropie angeben können. In der Mean-Field Näherung für die **mittlere Energie** $E = \langle \mathcal{H}_{\text{Potts}} \rangle$ nähern wir

$$\langle \delta_{s_i s_j} \rangle = \sum_{s=1}^Q \langle \delta_{s_i, s} \delta_{s_j, s} \rangle \approx \sum_{s=1}^Q \langle \delta_{s_i, s} \rangle \langle \delta_{s_j, s} \rangle,$$

so dass

$$E = \langle \mathcal{H}_{\text{Potts}} \rangle \approx -\frac{1}{\beta} N \frac{z}{2} K \left(\sum_{s=1}^Q x_s^2 - 1 \right). \quad (12.27)$$

Die **Mischungsentropie** für ein System aus Q verschiedenen Komponenten mit Anteilen x_s ist (ohne Näherung) durch

$$TS = \frac{1}{\beta} N \sum_{s=1}^Q x_s \ln x_s \quad (12.28)$$

gegeben. Nun bilden wir die freie Energie $F = E - TS$ und verwenden (12.26) für die x_s , um die freie Energie $F = F(m)$ als Funktion der Magnetisierung zu erhalten. Nach einigen Umformungen bekommen wir

$$\begin{aligned} \frac{1}{N} \beta(F(m) - F(0)) &= \frac{1 + (Q - 1)m}{Q} \ln(1 + (Q - 1)m) + \frac{Q - 1}{Q} (1 - m) \ln(1 - m) \\ &\quad - \frac{z}{2} K \left(\frac{1}{Q^2} [1 + (Q - 1)m]^2 + \frac{Q - 1}{Q^2} (1 - m)^2 - \frac{1}{Q} \right). \\ &\quad \underbrace{- \frac{z}{2} K \frac{Q - 1}{Q} m^2}_{} \end{aligned}$$

Taylorentwicklung um $m = 0$ liefert dann schließlich eine freie Energie

$$\frac{1}{N(Q - 1)} \beta(F(m) - F(0)) \approx \frac{1}{2Q} (Q - zK)m^2 - \frac{1}{6} (Q - 2)m^3 + \frac{1}{12} (Q^2 - 3Q + 3)m^4 + \dots, \quad (12.29)$$

die die Form einer **Landau freien Energie** hat, in der allerdings ein m^3 -Term auftritt, der im allgemeinen Q-Zustands Potts-Modell auch nicht durch Symmetrien verboten ist. Im Ising-Spezialfall $Q = 2$ verschwindet dieser Term, was auch durch die $m \leftrightarrow -m$ Z_2 -Symmetrie des Ising-Modells widerspiegelt.

Aus diesem Ergebnis können wir mehrere Schlussfolgerungen bezüglich des Perkolationsübergangs ziehen, wenn wir den Limes $Q \rightarrow 1$ betrachten:

- Eine Diskussion der freien Energie als Funktion von m zeigt, dass in Anwesenheit eines m^3 -Terms mit *negativem* Vorfaktor, wie er für $Q > 2$ vorliegt, der Phasenübergang *diskontinuierlich* wird mit einem Sprung in der Magnetisierung m am Übergang. Wir stellen allerdings fest, dass für $Q \leq 2$ der Vorfaktor des m^3 -Terms *positiv* ist. Dann ist der Phasenübergang *kontinuierlich* und findet statt, wenn der Vorfaktor $a(T) \propto (Q - zK) = Q(1 - zJ/k_B T)$ des m^2 -Terms sein Vorzeichen wechselt. Daher finden wir bei $Q = 1$ einen **kontinuierlichen Perkolationsübergang**. Insgesamt finden wir im Q-Zustands Potts-Modell:

$Q > 2$: Phasenübergang 1. Ordnung
$Q = 2$ Ising : kontinuierlicher Phasenübergang
$Q = 1$ Perkolation : kontinuierlicher Phasenübergang

- Der kritische Punkt für $Q \leq 2$ ist gegeben durch den Punkt, wo der Vorfaktor $a(T) \propto (Q - zK)$ sein Vorzeichen wechselt, also gilt $K_c = Q/z$. Für die Perkolation bei $Q = 1$ und mit $p = 1 - e^{-K}$ erhalten wir dann eine Mean-Field Perkolationsschwelle

$$p_c = 1 - e^{-K_c(Q=1)} = 1 - e^{-1/z},$$

die nur von der Koordinationszahl z des Gitters abhängt. Das Ergebnis zeigt das richtige Verhalten $\lim_{z \rightarrow \infty} p_c = 0$, dass im Falle unendlich vieler Nachbarn das System sofort perkoliert (und umgekehrt $\lim_{z \rightarrow 0} p_c = 1$ im wenig realistischen Limes $z \rightarrow 0$). Wir können auch mit einigen exakten Werten vergleichen (siehe auch Tabelle 12.1) für Bond-Perkolation vergleichen: Für ein Quadratgitter in $D = 2$ mit $z = 4$ ergibt die Mean-Field Theorie $p_c = 1 - e^{-1/4} \approx 0.22$, während der exakte Wert $p_c = 1/2$ deutlich größer ist. Für $D = 4$ mit $z = 8$ ergibt die Mean-Field Theorie $p_c = 1 - e^{-1/8} \approx 0.12$, während hier der exakte Wert $p_c \approx 0.16$ nur noch wenig größer ist. In der Tat nähert sich die Mean-Field Theorie für kubische Gitter mit $z = 2D$ in hohen Dimensionen $p_c \approx 1/z \approx 1/2D$. Diese Asymptotik ist in der Tat korrekt [2]. Wir stellen also fest, dass die Mean-Field Theorie in hohen Dimensionen gute Resultate liefert. Dies ist allgemein der Fall, das Mean-Field Approximationen gut werden im Limes vieler Nachbarn, also hoher Dimensionen.

12.5 Simulationsmethoden

Perkolation muss in endlichen Systemen simuliert werden. Daher werden Ergebnisse mit Hilfe von Finite-Size-Scaling analysiert. Wesentlich sind dabei Algorithmen zur Identifizierung der Cluster wie der Hoshen-Kopelman Algorithmus.

In diesem Kapitel werden wir noch zwei wichtige Aspekte von Computersimulationen der Perkolation diskutieren, Finite-Size-Scaling und Algorithmen zur Clusterbestimmung.

12.5.1 Finite-Size-Scaling

Zunächst stellen wir fest, dass wir bei der Untersuchung von Perkolation in einem endlichen System immer eine **Mittelung über viele Realisationen** $\langle \dots \rangle$ (die Mittelung $\langle \dots \rangle_{\text{Perk}}$ aus Abschnitt 12.4)

durchführen müssen, wenn wir mittlere Clustereigenschaften wie P_∞ oder S berechnen wollen. Hätten wir ein echt unendliches System zur Verfügung könnte die räumliche Mittelung innerhalb des Systems diesen Mittelung ersetzen.

Außerdem bekommen wir im endlichen System Finite-Size Effekte, sobald die mittlere Clustergröße ξ größer als die Systemgröße L wird. Als Beispiel betrachten wir

$$q_L \equiv \text{Wahrscheinlichkeit, einen perkolierenden Cluster zu finden} \quad (12.30)$$

in einem System der Größe L . Wir können q_L auch als eine Mittelung über viele Realisationen auffassen, wenn wir die Observable

$$\gamma_L = \begin{cases} 1 & \exists \text{ perkolierender Cluster} \\ 0 & \text{sonst} \end{cases}$$

einführen, mit der

$$q_L = \langle \gamma_L \rangle$$

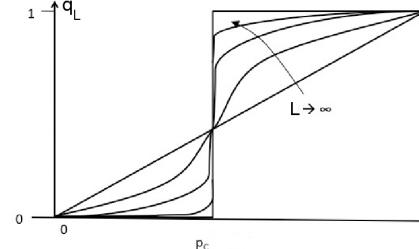
bei Mittelung über viele Realisationen gilt.

In einem unendlichen System sollten wir eine echte Stufenfunktion

$$q_\infty(p) = \begin{cases} 1 & p > p_c \\ 0 & p < p_c \end{cases} \quad (12.31)$$

finden. Im endlichen System wird q_L allerdings vom Verhältnis ξ/L von mittlerer Clustergröße $\xi \sim |p - p_c|^{-\nu}$ und Systemgröße L abhängen, also von der Variablen $(p - p_c)L^{1/\nu}$ ($= (L/\xi)^{1/\nu}$):

$$q_L(p) = f((p - p_c)L^{1/\nu}). \quad (12.32)$$



Die Skalenfunktion $f(x)$ sollte die Grenzwerte $f(-\infty) \approx 0$ und $f(\infty) \approx 1$ besitzen, um im Limes eines unendlichen Systems $L \rightarrow \infty$ wieder (12.31) zu bekommen.

Ähnlich erwarten wir für den mittleren Anteil des perkolierenden Clusters am besetzten Teil des Gitters $P_\infty \propto (p - p_c)^\beta \propto \xi^{-\beta/\nu}$, dass im endlichen System ξ durch L "abgeschnitten" wird, also dass

$$P_L = L^{-\beta/\nu} f((L/\xi)^{1/\nu}) = L^{-\beta/\nu} f((p - p_c)L^{1/\nu}) \quad (12.33)$$

gilt, analog zur Magnetisierung im Ising-Modell, siehe (11.62) in Kapitel 11.5.2. Die Analyse solcher **Finite-Size-Effekte** erfolgt dann auch ganz analog mit den in Kapitel 11.5.2 diskutierten Methoden, siehe auch Übungsaufgabe 1 am Kapitelende.

Insbesondere ist die Größe $q_L(p)$ sehr gut geeignet, um die Perkolationsschwelle zu bestimmen. Nach Gl. (12.32) sollte bei $p = p_c$ für $q_L(p)$ -Kurven für verschiedene L gelten $q_L(p_c) = f(0)$, d.h. alle diese Kurven sollten sich in einem Punkt schneiden (siehe auch Abbildung rechts). Damit lässt sich die Perkolationsschwelle p_c ähnlich gut über den Schnittpunkt der Kurven $q_L(p)$ bestimmen, wie die kritische Temperatur T_c des Ising-Modells über den Schnittpunkt der Binder-Kumulant $U_L(T)$, siehe Gl. (11.65), wenn man einen Algorithmus hat, der schnell bestimmt, ob es einen perkolierenden Cluster gibt, z.B. der Leath- oder Hoshen-Kopelman Algorithmus aus dem nächsten Abschnitt. In Abbildung 12.3 ist das Ergebnis einer Simulation gezeigt, wo der Hoshen-Kopelman Algorithmus zur Bestimmung des perkolierenden Clusters (Messung von q_L) und seiner Größe (Messung von P_L) genutzt wurde.

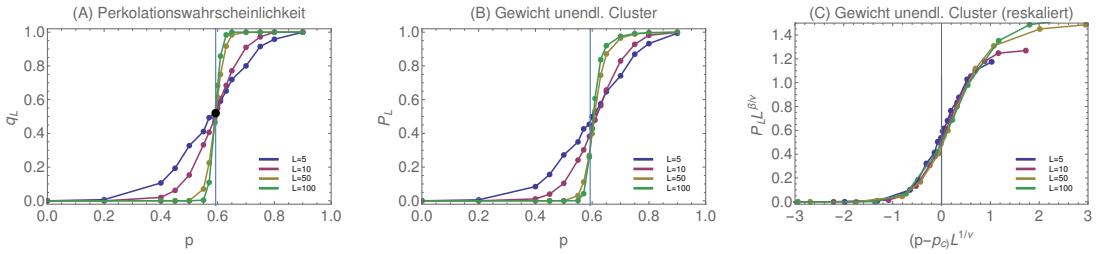


Abbildung 12.3: Site-Perkolation auf dem 2D Quadratgitter, siehe auch Abbildung 12.4 weiter unten. Simulationsergebnisse für (A) die Perkolationswahrscheinlichkeit q_L (einen Cluster zu finden, der vom oberen zum unteren Ende reicht) und (B,C) das Gewicht P_L des perkolierenden Clusters (seine Größe im Verhältnis zur Gesamtzahl besetzter Plätze) als Funktion von p . Der Literaturwert für die Perkolationsschwelle im thermodynamischen Limes ist $p_c \simeq 0.593$ (siehe Tabelle 12.1). Simulationen sind für 4 Systemgrößen $L = 5, 10, 50, 100$ ($N = L^2$) über 1000 Realisationen gemittelt; Cluster und ihre Größen wurden mit dem Hoshen-Kopelman Algorithmus ermittelt, siehe auch Abbildung 12.4. In (C) ist $P_L L^{\beta/\nu}$ (mit $\beta = 5/36$ und $\nu = 4/3$) reskaliert gegen die reskalierte reduzierte Wahrscheinlichkeit $(p - p_c)L^{1/\nu}$ aufgetragen, siehe Gl. (12.33). In der Nähe der Perkolationsschwelle fallen alle Daten auf eine Masterkurve. Der Punkt in Abb. (A) zeigt den Schnittpunkt von $q_L(p)$ für die beiden größten Systeme $L = 50$ und 100 und liegt bei $p_c \simeq 0.593$ und $q_L(p_c) = 0.52$ (in Übereinstimmung mit theoretischen Vorhersagen [6]).

12.5.2 Hoshen-Kopelman Algorithmus

Ein zentrales Computerproblem bei der Perkolation ist die Identifizierung der Cluster, um sie dann weiter nach Größe und Masse analysieren zu können. Dazu gibt es verschiedene Algorithmen.

In der Übungsaufgabe 1 am Kapitelende ist ein einfacher Algorithmus skizziert, der auch als **Leath Algorithmus** bezeichnet wird, der ganz ähnlich funktioniert wie der Aufbau eines zu flippenden Clusters im Wolff-Algorithmus in Kapitel 11.6. Der Algorithmus startet von einem besetzten Platz und findet alle Plätze, die zum gleichen Cluster gehören. Dazu werden die Punkte, die bereits zum Cluster gehören als “besucht” oder “unbesucht” markiert, indem eine Liste (LIFO,FIFO) verwaltet wird mit noch abzusuchenden “unbesuchten” Clusterplätzen zusätzlich zur Liste, die alle “besuchten” Clusterplätze enthält. Wird ein Platz aus der Liste der “unbesuchten” Clusterplätze abgearbeitet, werden alle Nachbarn dieses Punktes, die zum Cluster gehören und noch nicht in der Liste “besuchter” Clusterplätze sind, der Liste der “unbesuchten” Clusterplätze hinzugefügt; der so abgearbeitete Clusterplatz wird in die Liste der “besuchten” Clusterplätze transferiert. Der Algorithmus arbeitet so lange, bis die Liste der “unbesuchten” Clusterplätze abgearbeitet ist. Dann ist die dazu parallel aufgebaute Liste der “besuchten” Clusterplätze vollständig.

Der **Hoshen-Kopelman Algorithmus** zur effektiven Identifizierung der Cluster in einer Konfiguration des Systems wurde in [7] im Jahr 1976 im Kontext der Perkolation vorgestellt. Er identifiziert in einem Durchlauf durch das Gitter *alle* Cluster und ermittelt gleichzeitig die Clustermassen. Die Identifizierung von Clustern ist ein weitaus allgemeineres Problem, das in vielen Bereichen der Informatik wichtig ist. Ganz allgemein definiert die Clusterzugehörigkeit eine **Äquivalenzrelation** für die Plätze bzw. Bonds. Die Cluster sind dann **Äquivalenzklassen**. In der Informatik werden zur Einteilung von Elementen in Äquivalenzklassen sogenannte **Union-Find Algorithmen** verwendet. Der Hoshen-Kopelman Algorithmus ist im Prinzip ein spezieller Union-Find Algorithmus für das Site-Perkolationsproblem, der alle Cluster (Äquivalenzklassen) identifiziert und auch die Clustermassen (Größer der Äquivalenzklassen) ermittelt.

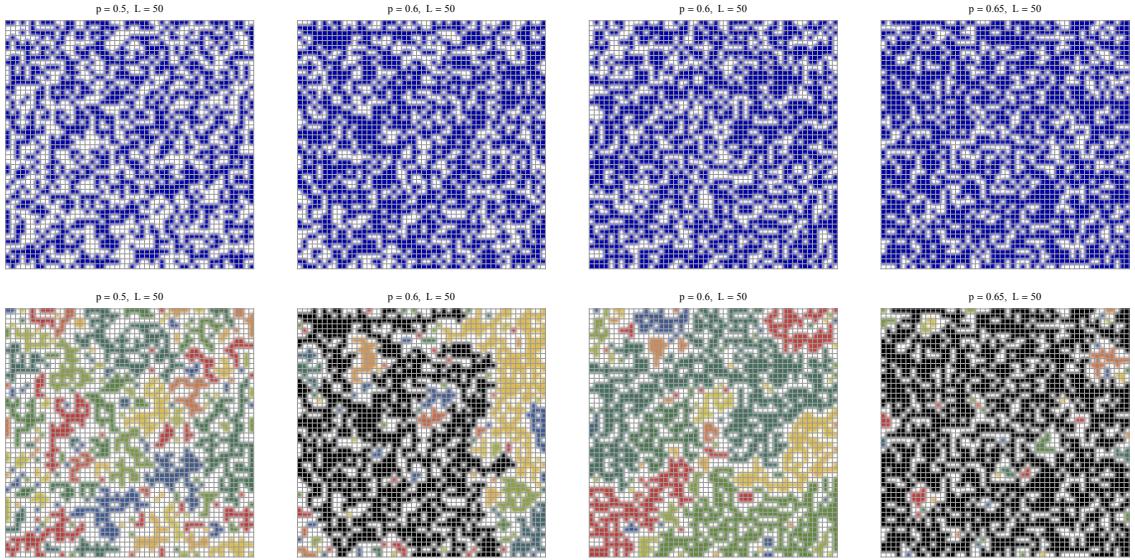


Abbildung 12.4: Site-Perkolation auf dem 2D Quadratgitter. Oben: Typische Realisationen für $L = 50$ für $p = 0.5, 0.6, 0.65$ (Perkolationsschwelle $p_c \simeq 0.593$), blaue Plätze sind besetzt, weiße unbesetzt. Unten: Für diese Realisationen wurden die zusammenhängenden Cluster mit dem Hoshen-Kopelman Algorithmus identifiziert und eingefärbt. Perkolierende Cluster (die vom oberen zum unteren Ende reichen) sind schwarz eingefärbt.

Im Hoshen-Kopelman Algorithmus gibt es

- (i) ein **Array** l_i (von der Größe und Struktur des Gitters) mit einem **Clusterlabel** (positive Zahlen $1, 2, \dots$) für jeden Gitterplatz i . Besetzte Plätze, die zum selben Cluster gehören, sollten mit äquivalenten Labels versehen werden;
- (ii) einen **Vektor** c_j , der die äquivalenten **Cluster-Klassen** und die **Cluster-Größen** angibt. Zu jedem Label $j = 1, 2, \dots$ gibt c_j entweder ein äquivalentes Cluster-Label k an (mit einer negativen Zahl $-k$) oder die **Cluster-Größe** des Clusters mit dem Label j (eine positive Zahl). Eine negative Zahl $c_j = -k$ bedeutet, dass Elemente mit Cluster-Label j eigentlich zum Cluster mit Label k gehören, also das Cluster j und k äquivalent sind. Eine positive Zahl $c_j > 0$ gibt die Gesamtgröße G_j der Cluster mit Label j und aller äquivalenten Cluster an.
- (iii) Dadurch gibt es zu jedem Clusterlabel k ein **eigentliches Clusterlabel** \tilde{k} , das man erhält, indem man im Klassenvektor den negativen Zeigern „nachgeht“, bis $c_{\tilde{k}} > 0$ wird: Wenn z.B. $c_1 = 4, c_2 = -1, c_3 = -2$ und wir das eigentliche Clusterlabel $\tilde{3}$ zu Clusterlabel 3 suchen, finden wir $\tilde{3} = 1$ wegen $3 \rightarrow -c_3 = 2 \rightarrow -c_2 = 1$ und $c_1 > 0$. Clusterlabel 3 gehört also eigentlich zu Cluster 2, der wiederum zu Cluster 1 und $c_{\tilde{3}} = c_1 = 4$ gibt schließlich die Größe des gesamten Clusters an, der aus allen Plätzen mit äquivalenten Labels 1, 2 und 3 besteht.

Der Hoshen-Kopelman Algorithmus baut nun Label-Array und Klassen-Vektor auf, während er vollständig durch das Gitter läuft. Wir beschreiben den Algorithmus für $D = 2$:

- Zu Beginn wird das zweidimensionale Label-Array initialisiert mit dem Label 0 auf jedem unbesetzten Gitterplatz und -1 auf jedem besetzten, aber noch nicht klassifizierten Platz. Die Werte -1 werden im weiteren Verlauf sequentiell durch Clusterlabels (positive Zahlen

$1, 2, \dots$) ersetzt.

Label-Array:	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>-1</td><td>0</td><td>-1</td><td>0</td><td>-1</td></tr> <tr><td>-1</td><td>0</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>0</td><td>-1</td><td>-1</td><td>0</td></tr> <tr><td>...</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>	-1	0	-1	0	-1	-1	0	-1	-1	-1	-1	0	-1	-1	0	...										$J = 0$
-1	0	-1	0	-1																							
-1	0	-1	-1	-1																							
-1	0	-1	-1	0																							
...																											

- Wir gehen dazu sequentiell durch das Label-Array, d.h. von links nach rechts durch jede Zeile und von oben nach unten durch aufeinanderfolgende Zeilen. Das “aktuelle” als nächstes zu vergebene Label sei J mit $J = 0$ am Anfang.
- In der ersten Zeile wird jeder Platz $l_i = -1$, der als linken Nachbarn eine 0 hat, als Anfang eines neuen Clusters gewertet. Dann wird J um eins hochgezählt und der Platz mit dem neuen Label $l_i = J$ versehen. Im Klassen-Vektor wird ein $c_J = 1$ eingetragen für die Anfangsgröße 1 des neuen Clusters.

Jeder Platz $l_i = -1$, der als linken Nachbarn keine 0 hat, gehört zu dem gleichen Cluster wie dieser Nachbar. Daher wird er mit dem Label $l_i = J$ versehen, dass dann bereits links von ihm steht und im Klassen-Vektor wird die Cluster-Größe $c_J \rightarrow c_J + 1$ hochgezählt (der Zeiger J bleibt unverändert).

Label-Array:	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <tr><td>1</td><td>0</td><td>2</td><td>0</td><td>3</td></tr> <tr><td>-1</td><td>0</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>0</td><td>-1</td><td>0</td><td>-1</td><td>0</td></tr> <tr><td>...</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> </table>	1	0	2	0	3	-1	0	-1	-1	-1	0	-1	0	-1	0	...										Klassen-Vektor, $J = 3$:	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th>Label j</th> <th>c_j</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td></tr> <tr><td>3</td><td>1</td></tr> </tbody> </table>	Label j	c_j	1	1	2	1	3	1
1	0	2	0	3																																
-1	0	-1	-1	-1																																
0	-1	0	-1	0																																
...																																				
Label j	c_j																																			
1	1																																			
2	1																																			
3	1																																			

- 1) In den weiteren Zeilen wird jeder Platz $l_i = -1$, der als oberen *und* linken Nachbarn eine 0 hat, wieder als Anfang eines neuen Clusters gewertet. Dann wird wieder J um eins hochgezählt und der Platz mit dem neuen Label $l_i = J$ versehen. Im Klassen-Vektor wird ein $c_J = 1$ eingetragen für die Anfangsgröße 1 des neuen Clusters.
- 2) Ein Platz $l_i = -1$, der *entweder* als oberen *oder* als linken Nachbarn keine 0 hat, gehört zu dem gleichen Cluster wie dieser Nachbar und bekommt den gleichen Label j wie dieser Nachbar, $l_i = j$. Im Klassen-Vektor wird die entsprechende Cluster-Größe $c_j \rightarrow c_j + 1$ hochgezählt (der Zeiger J bleibt unverändert). Sind bereits einige Cluster als äquivalent klassifiziert worden (siehe Schritt 3) ist zu beachten, dass man bei dem Update $c_j \rightarrow c_j + 1$ die *eigentlichen* Clusterlabels verwenden muss, da unter c_j die positiven Gesamt-Clustergrößen aller äquivalenter Cluster abgelegt sind.
- 3) Ein Platz $l_i = -1$, der *weder* als linken *noch* als oberen Nachbarn eine 0 hat, bekommt den *kleineren* der beiden Nachbarlabel j als neuen Label, $l_i = j$. Der *größere* der beiden Nachbarlabels sei k , also $j \leq k$.

Sind die Nachbarlabel j und k verschieden, also $j < k$, werden die zwei Cluster j und k durch diesen Platz verbunden und sind damit äquivalent. Im Klassen-Vektor wird diese Äquivalenz dadurch angezeigt, dass $c_k = -j$ gesetzt wird. Dies hat die Funktion eines Zeigers von k auf den äquivalenten Cluster j , dem ein negativer Wert gegeben wird, um Zeiger von Clustergrößen sofort unterscheiden zu können.

Die *Gesamtgröße* der äquivalenten Cluster j und k wird dann in c_j abgelegt als $c_j \rightarrow c_j + 1 + c_k$. Hier ist wieder zu beachten, dass man die *eigentlichen* Clusterlabels verwendet, um die positiven Gesamt-Clustergrößen zu addieren. (Achtung: die Ersetzung $c_j \rightarrow c_j + 1 + c_k$ wird tatsächlich gemacht, *bevor* $c_k = -j$ gesetzt wird).

Ist das linke Nachbarlabel j gleich dem oberen Nachbarlabel k , also $j = k$, fügt sich der Platz in den Cluster j ein und wir setzen einfach $c_j \rightarrow c_j + 1$ (der Zeiger J bleibt unverändert),

wobei auch wieder das *eigentliche* Clusterlabel verwendet werden muss.

Label-Array:	<table border="1"><tr><td>1</td><td>0</td><td>2</td><td>0</td><td>3</td></tr><tr><td>1</td><td>0</td><td>2</td><td>2</td><td>-1</td></tr><tr><td>0</td><td>-1</td><td>0</td><td>-1</td><td>0</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	1	0	2	0	3	1	0	2	2	-1	0	-1	0	-1	0	...									
1	0	2	0	3																						
1	0	2	2	-1																						
0	-1	0	-1	0																						
...																										

Klassen-Vektor, $J = 3$:

Label j	c_j
1	2
2	3
3	1

Label-Array:	<table border="1"><tr><td>1</td><td>0</td><td>2</td><td>0</td><td>3</td></tr><tr><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr><tr><td>0</td><td>-1</td><td>0</td><td>-1</td><td>0</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	1	0	2	0	3	1	0	2	2	2	0	-1	0	-1	0	...									
1	0	2	0	3																						
1	0	2	2	2																						
0	-1	0	-1	0																						
...																										

Klassen-Vektor, $J = 3$:

Label j	c_j
1	2
2	5
3	-2

Label-Array:	<table border="1"><tr><td>1</td><td>0</td><td>2</td><td>0</td><td>3</td></tr><tr><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td></tr><tr><td>0</td><td>4</td><td>0</td><td>2</td><td>0</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	1	0	2	0	3	1	0	2	2	2	0	4	0	2	0	...									
1	0	2	0	3																						
1	0	2	2	2																						
0	4	0	2	0																						
...																										

Klassen-Vektor, $J = 4$:

Label j	c_j
1	2
2	6
3	-2
4	1

Die Abbildung 12.4 zeigt typische Konfigurationen bei Site-Perkolation auf dem 2D Quadratgitter, wo die Cluster mit Hilfe des Hoshen-Kopelman Algorithmus identifiziert wurden.

12.6 Literaturverzeichnis Kapitel 12

- [1] D. Stauffer und A. Aharony. *Introduction To Percolation Theory*. Taylor & Francis, 1994.
- [2] D. Stauffer. *Scaling theory of percolation clusters*. Phys. Rep. **54** (1979), 1–74.
- [3] F. Schwabl. *Statistische Mechanik*. Springer-Lehrbuch. Springer, 2006.
- [4] F. Y. Wu. *The Potts model*. **54** (1982), 235–268.
- [5] F. Y. Wu. *Percolation and the Potts model*. J. Stat. Phys. **18** (1978), 115–123.
- [6] M. E. J. Newman und R. M. Ziff. *Fast Monte Carlo algorithm for site or bond percolation*. Phys. Rev. E **64** (Juni 2001), 016706.
- [7] J. Hoshen und R. Kopelman. *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm*. Phys. Rev. B **14** (1976), 3438–3445.

12.7 Übungen Kapitel 12

1. Site-Perkolation auf dem $D = 2$ Quadratgitter

Ziel dieser Aufgabe ist die Bestimmung von p_c für die Site-Perkolation auf dem Quadratgitter sowie die Bestimmung von kritischen Exponenten des Perkolationsübergangs. Die Aufgabenstellung soll Ihnen bei der Erstellung der Simulation helfen.

a) Initialisierung: Erzeugen Sie eine Datenstruktur (etwa ein Array) für ein Gitter mit $L \times L$ Plätzen und erzeugen Sie R Realisationen, indem Sie jeweils jeden Platz mit Wahrscheinlichkeit p besetzen. Ihr Programm sollte in der Lage sein, die Systemgröße L , die Besetzungswahrscheinlichkeit p und die Zahl an Realisationen R als Parameter zu verarbeiten.

b) Cluster finden: Schreiben Sie eine Routine, die in der Lage ist, alle Cluster aus besetzten Punkten zu finden. Eine Strategie (die Ihnen vom Wolff-Algorithmus bekannt sein sollte) ist es, eine Liste (LIFO,FIFO) zu verwalten, mit noch abzusuchenden Plätzen und immer die Nachbarn eines in den Cluster aufgenommen Punktes der Liste hinzuzufügen. Diese Suche startet bei einem besetzten Platz und lässt den Cluster dann solange wachsen, bis es keine benachbarten besetzten Plätze mehr gibt. Das wird solange wiederholt, bis alle Plätze entweder besucht wurden oder unbesetzt sind. Sie können bei der Implementation Datenstrukturen aus einer Bibliothek (etwa STL Container) verwenden. Sie sollten danach für jeden besetzten Platz wissen, in welchem Cluster er sich befindet. Dies ist der Leath-Algorithmus. Alternativ können Sie auch den Hoshen-Kopelman-Algorithmus implementieren.

Visualisieren Sie das Ergebnis der Clustersuche für jeweils eine Realisation mit $N = 50$ für $p = 0.1, 0.5, 0.9$ zum Beispiel in dem Sie die Plätze entsprechend ihrer Clusterzugehörigkeit einfärben.

c) Überprüfen Sie, ob es einen perkolierenden Cluster gibt und bestimmen Sie die relative Perkolationshäufigkeit $q_L(p)$ für mindestens $L = 10, 50, 100$. Bestimmen Sie zuvor anhand von $q_{10}(p)$ einen geeigneten Wert für R , indem Sie für R im Bereich von $R = 10$ bis $R = 10^4$ variieren.

d) Begründen Sie, warum $q_L(p_c)$ unabhängig von L sein sollte, und bestimmen Sie damit p_c mit möglichst hoher Genauigkeit.

e) Ändern Sie Ihre Clustersuche so ab, dass Sie auch die Größe des größten Clusters bestimmen. Der Anteil am gesamten Gitter, der durch den größten Cluster belegt wird, wird mit M_∞ bezeichnet ($M_\infty = pP_\infty$). Es gilt

$$M_\infty(p) \sim |p - p_c|^{-\beta}$$

mit einem Exponenten β . Versuchen Sie β durch Messung von $M_\infty(p)$ in Systemen **einer** möglichst großen Größe L sowie durch Finite-Size-Scaling

$$M_\infty(p)L^{\beta/\nu} \sim f((p - p_c)^{1/\nu}L)$$

zu bestimmen. Variieren Sie dazu β und ν so, dass die Datenpunkte möglichst gut auf eine Funktion f kollabieren (wiederum min. für $L = 10, 50, 100$).

13 Simulation stochastischer Bewegungsgleichungen

Literatur zu diesem Teil:

Numerical Recipes [1, 2], Thijssen [3], Landau und Binder [4], Gould/Tobochnik [5]. Für eine Einführung in stochastische Bewegungsgleichungen siehe auch Schwabl [6].

Neben der Molekulardynamik (MD) Simulation aus Kapitel 5 und der Monte-Carlo (MC) Simulation aus Kapitel 11 gibt es eine weitere Möglichkeit ein klassisches System aus einem oder vielen Teilchen mit thermischen Fluktuationen zu simulieren: Wir erweitern die Newtonschen Bewegungsgleichungen um eine **Reibungskraft** und eine **stochastische thermische Kraft**, die beide durch die Kopplung an ein umgebendes Fluid mit Temperatur T zustande kommen, und erhalten **stochastische Bewegungsgleichungen**, und zwar entweder die **Langevin-Gleichung** (mit Inertialterm) oder die **Brownsche Dynamik** (ohne Inertialterm). Die Dynamik kann dann durch Lösung dieser stochastischen gewöhnlichen Differentialgleichungen mit den Methoden aus Kapitel 4 simuliert werden.

13.1 Brownsche Bewegung, Langevin-Gleichung

Die Brownsche Bewegung von Teilchen in Fluiden lässt sich durch die Langevin-Gleichung beschreiben, die eine stochastische Bewegungsgleichung mit Reibung und stochastischer thermischer Kraft darstellt. Wir betrachten Geschwindigkeitskorrelationen und leiten das Fluktuations-Dissipations-Theorem ab für den Zusammenhang zwischen der stochastischen Kraft, Temperatur und Reibung. Wir betrachten auch die mittlere quadratische Auslenkung und leiten die Einstein-Relation für die Diffusionskonstante her. Die Brownsche Dynamik stellt den überdämpften Limes der Langevin-Gleichung dar.

13.1.1 Ein Teilchen

Wir betrachten zunächst *ein* Teilchen, das in Kontakt mit einem Wärmebad steht, das in Form eines umgebenden Fluids (Gas oder Flüssigkeit) aus vielen, noch kleineren Teilchen realisiert sein soll, bei einer Temperatur T , die durch das Wärmebad festgelegt wird. Das Teilchen bewege sich zudem in einem **äußeren Potential** $U(\vec{r})$, dass eine **Kraft** $\vec{F} = -\vec{\nabla}U$ erzeugt, und habe eine Masse m . Im Vakuum wäre die Newtonsche Bewegungsgleichung des Teilchens also

$$m\ddot{\vec{r}} = -\vec{\nabla}U(\vec{r}) \quad (13.1)$$

In einem umgebenden Fluid (unserem Wärmebad) gibt es dagegen zusätzlich eine **Reibungskraft**. Diese kommt dadurch zustande, dass Impuls und kinetische Energie vom Teilchen auf das Bad übertragen und dort in Wärme (d.h. ungeordnete Bewegung der Bad-Teilchen, die wir nicht im Detail verfolgen werden) umgewandelt werden. In einer **viskosen Flüssigkeit** beispielsweise (also eine Flüssigkeit mit kleiner Reynoldszahl $Re = \rho v L / \eta$, wo ρ die Flüssigkeitsdichte, v die typische

Geschwindigkeit von Flüssigkeitsteilchen, L eine typische Längenskala und η die Viskosität sind) gilt die **Stokes-Reibung**

$$\vec{F}_R = -\Gamma \dot{\vec{r}} \quad (13.2)$$

mit einem **Stokes-Reibungskoeffizienten**

$$\Gamma = 6\pi\eta R \quad (13.3)$$

für eine Kugel mit Radius R in einer Flüssigkeit mit dynamischer Viskosität η (Wasser hat $\eta = 10^{-3}$ kg/ms). Für hinreichend kleine Geschwindigkeiten $\dot{\vec{r}}$ sollte die Reibungskraft \vec{F}_R des Bades als linearer Response immer linear sein.

Das Wärmebad hat aber noch einen anderen, komplementären Effekt: Es überträgt durch Stöße der Bad-Teilchen auch Kräfte auf unser Teilchen. Dies führt zu einer **stochastischen Kraft** $\vec{\zeta}(t)$, dem **“thermischen Rauschen”** oder der **thermischen Kraft** auf das Teilchen. Diese Stöße erfolgen zufällig (wir wollen keine Information über die ungeordnete Bewegung der Bad-Teilchen nachverfolgen), daher führt die stochastische Kraft zu einer Zufallskomponente in der Geschwindigkeit unseres Teilchens. Experimentell wurde genau das von dem Botaniker Robert Brown im Jahr 1827 beobachtet an der Bewegung von Blütenpollen unter dem Mikroskop [7], der den Teilchen damals fälschlicherweise eine gewisse eigene “Lebendigkeit” oder “Aktivität” zuschrieb. Diese sogenannte **Brownsche Bewegung** kleiner in einem Fluid suspendierter Teilchen wurde 1905 von Einstein erklärt durch die Stöße mit den (unsichtbaren) Bad-Teilchen [8].

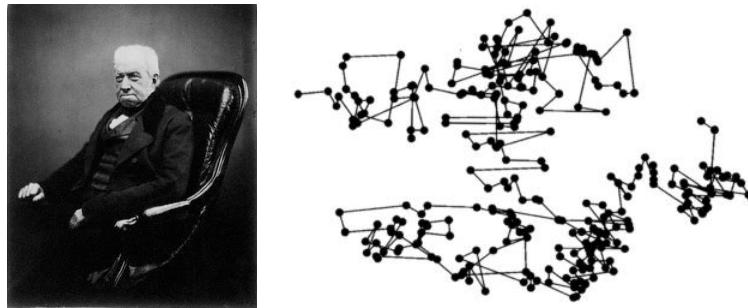


Abbildung 13.1: Links: Robert Brown (1773-1869), Botaniker. (Quelle: Wikipedia). Rechts: Brownsche Bewegung eines Teilchens in zwei Raumdimensionen.

Wir betrachten ein **Beispiel**: Eine Kugel bewege sich in einer viskosen Flüssigkeit im Schwerefeld, siehe Abb. 13.2. Eine makroskopische Kugel mit Radius $R \sim \text{m}$ im Meterbereich wird mit konstanter Geschwindigkeit $v = mg/\Gamma$ gerade nach unten sinken. Mikroskopische Teilchen mit einem Durchmesser im Mikrometerbereich $R \sim \mu\text{m}$ werden dagegen eine überlagerte Zufallskomponente zeigen, da die stochastische thermische Kraft $\vec{\zeta}(t)$ vergleichbar mit $mg \propto R^3$ (wegen Masse $\propto R^3$ bei fester Dichte) oder $\Gamma v \sim R$ (Stokesreibung) wird.

Mit der Reibungskraft und der stochastischen Kraft $\vec{\zeta}(t)$ wird aus der Newtonschen Bewegungsgleichung (13.1) die **Langevin-Gleichung** (Langevin 1908)

$$m\ddot{\vec{r}} = -\Gamma \dot{\vec{r}} - \vec{\nabla}U(\vec{r}) + \vec{\zeta}(t) \quad (13.4)$$

Wir wollen uns nun klarmachen, welche Eigenschaften eine stochastische Kraft, die durch Stöße mit den Bad-Teilchen hervorgerufen wird, besitzen muss:

- Die Eigenschaften einer solchen Kraft können nur “im Mittel” bekannt sein. Im Folgenden sei $\langle \dots \rangle = \text{Mittel über viele Trajektorien } \vec{r}(t) \text{ mit verschiedenen Realisationen von } \vec{\zeta}(t)$.

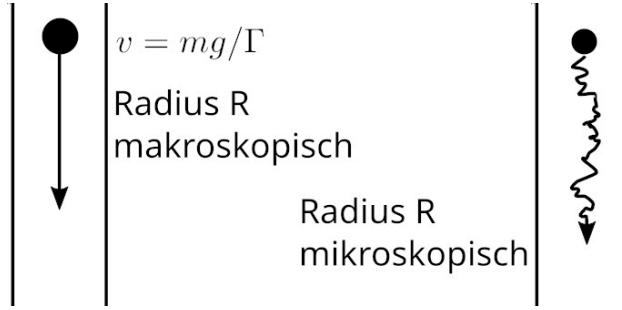


Abbildung 13.2: Teilchen in einer viskosen Flüssigkeit im Schwerefeld. Makroskopisch große Teilchen sinken mit konstanter Geschwindigkeit. Mikroskopische Teilchen zeigen eine überlagerte Brownsche Zufallsbewegung.

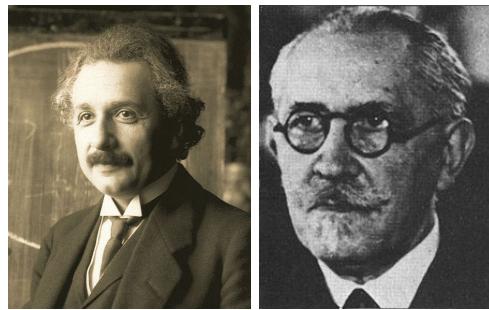


Abbildung 13.3: Links: Albert Einstein (1879-1955). Rechts: Paul Langevin (1872-1946), französischer Physiker. (Quelle: Wikipedia).

Diese Mittelung sollte gleich dem thermodynamischen Mittel im kanonischen Ensemble aus der statistischen Physik sein (wie die identische Schreibweise andeutet), da die umgebende Flüssigkeit ja auch unser Wärmebad ist.

- Die Kollisionen sind **isotrop**, also

$$\langle \vec{\zeta}(t) \rangle = 0 \quad (13.5)$$

Andernfalls würde sich ein Teilchen auf Grund von thermischen Stößen in eine *gerichtete* Bewegung versetzen lassen, was 1. und 2. Hauptsatz der Thermodynamik verletzen würde.

- Die Kollisionen sind **schnell** (Kollisionszeit τ) und **statistisch unabhängig**, also sollte $\langle \zeta_i(t)\zeta_j(t') \rangle = 0$ für $|t - t'| > \tau$ gelten. Für $t = t'$ wird $\langle \zeta^2(t) \rangle$ proportional zur Stärke der stochastischen Kräfte sein. Wir schreiben insgesamt

$$\langle \zeta_i(t)\zeta_j(t') \rangle = \lambda \delta_{ij} \delta_\tau(t - t') \quad (13.6)$$

wobei der Parameter λ die **Stärke** der stochastischen Kraft beschreibt. Aus der Isotropie der Kräfte folgt hier der Faktor δ_{ij} , d.h. die statistische Unabhängigkeit der räumlichen Komponenten der Kraft. Das Subskript τ an der δ -Funktion verdeutlicht die zeitliche Breite der δ -Funktion, die durch die kleine Kollisionszeit τ gegeben ist. Wir betrachten im folgenden den Limes $\tau \approx 0$; eine stochastische Kraft mit δ -Zeitkorrelationen (13.6) springt dann instantan zwischen beliebig kleinen Zeitabständen auf jeweils neue zufällige Werte und wird damit **unstetig in der Zeit**.

- Die letzte wichtige Eigenschaft ist die Tatsache, dass die Kraft $\zeta(t)$ die Summe vieler unabhängiger Kollisionskräfte darstellt. Dann können wir aber den **zentralen Grenzwertsatz** (siehe auch Kapitel 10.2 oder 11.1, Formel 11.10) anwenden und folgern, dass die Kraft $\zeta(t)$ **gaußverteilt** sein muss. Dann reichen die beiden ersten Momente 13.5 und 13.6 aber bereits aus, um die Verteilungsfunktion vollständig anzugeben:

$$P[\zeta(t)] = \mathcal{N} \exp \left(- \int_{-\infty}^{\infty} dt \frac{1}{2\lambda} \zeta^2(t) \right) \quad (13.7)$$

Dies ist die Wahrscheinlichkeit, dass ein ganzer Funktionsverlauf $\vec{\zeta}(t)$ auftritt, also eine Wahrscheinlichkeitsdichte für eine *Funktion*. Den Normierungsfaktor \mathcal{N} werden wir nicht berechnen.

Uns ist auch intuitiv klar, dass die Zufallskraft $\vec{\zeta}(t)$ etwas mit der Temperatur T des Bades zu tun haben muss, da die Mittelung $\langle \dots \rangle$ ja am Ende das thermodynamische Mittel im kanonischen Ensemble sein sollte. Sie sollte auch etwas mit dem Reibungskoeffizienten Γ zu tun haben, weil ja auch die Reibung im gleichen umgebenden Fluidbad zustande kommt. Diese Zusammenhänge werden wir noch klären.

Dazu lösen wir die Langevin-Gleichung 13.4 zunächst für ein **freies Teilchen**, also $U = 0$. Dabei müssen wir erst die Lösung, also die Teilchentrajektorie $\vec{r}(t)$, bestimmen für eine gegebene beliebige Realisation $\zeta(t)$ der stochastischen Kraft und dann eine Mittelung $\langle \dots \rangle$ durchführen über alle Realisationen von $\zeta(t)$ mit Hilfe von 13.5 und 13.6 oder der Verteilung 13.7.

Zuerst lösen wir also die Bewegungsgleichung für die Geschwindigkeit $\vec{v} = \dot{\vec{r}}$ für eine gegebene Realisation $\zeta(t)$ der stochastischen Kraft

$$\begin{aligned} m\dot{\vec{v}} &= -\Gamma\vec{v} + \vec{\zeta}(t) \\ \dot{\vec{v}} &= -\gamma\vec{v} + \vec{\eta}(t) \quad \text{mit } \gamma \equiv \Gamma/m \quad \text{und } \vec{\eta}(t) \equiv \vec{\zeta}(t)/m \end{aligned} \quad (13.8)$$

Dies ist eine lineare Differentialgleichung mit Inhomogenität $\vec{\eta}(t)$, bei der ein Ansatz $\vec{v}(t) = \vec{C}(t)e^{-\gamma t}$ (Variation der Konstanten) zur Lösung führt. Letztlich finden wir für die Geschwindigkeit

$$\vec{v}(t) = e^{-\gamma t} \left[\vec{v}(0) + \int_0^t dt' e^{\gamma t'} \vec{\eta}(t') \right] \quad (13.9)$$

Die Trajektorie $\vec{r}(t)$ bekommt man nach nochmaliger Integration, $\vec{r}(t) = \vec{r}(0) + \int_0^t dt' \vec{v}(t')$.

Nun führen wir die Mittelung $\langle \dots \rangle$ über die Realisationen von $\vec{\zeta}(t)$ durch mit Hilfe von 13.5 und 13.6. Wegen $\langle \vec{\zeta} \rangle = 0$ ist der Mittelwert der Geschwindigkeit dann

$$\langle \vec{v}(t) \rangle = e^{-\gamma t} \vec{v}(0)$$

Die **Geschwindigkeitskorrelationen** sind

$$\begin{aligned} \langle \vec{v}(t_1) \cdot \vec{v}(t_2) \rangle &= \vec{v}^2(0) e^{-\gamma(t_1+t_2)} + e^{-\gamma(t_1+t_2)} \int_0^{t_1} dt \int_0^{t_2} dt' \underbrace{\langle \vec{\eta}(t) \cdot \vec{\eta}(t') \rangle}_{\stackrel{(13.6)}{=} \frac{d\lambda}{m^2} \delta(t-t')} e^{\gamma(t+t')} \\ &\stackrel{t_1 \leq t_2}{=} \vec{v}^2(0) e^{-\gamma(t_1+t_2)} + \frac{d\lambda}{m^2} e^{-\gamma(t_1+t_2)} \int_0^{t_1} dt e^{2\gamma t} \\ &= \left(\vec{v}^2(0) - \frac{d\lambda}{m^2} \frac{1}{2\gamma} \right) e^{-\gamma(t_1+t_2)} + \frac{d\lambda}{m^2} \frac{1}{2\gamma} e^{-\gamma|t_1-t_2|} \end{aligned}$$

wobei d die Raumdimension ist ($\vec{r} \in \mathbb{R}^d$). Mit 13.5 bekommen wir also

$$\langle \vec{v}(t_1) \cdot \vec{v}(t_2) \rangle = \langle \vec{v}(t_1) \rangle \cdot \langle \vec{v}(t_2) \rangle - \frac{d\lambda}{m^2} \frac{1}{2\gamma} \left(e^{-\gamma(t_1+t_2)} - e^{-\gamma|t_1-t_2|} \right) \quad (13.10)$$

γ^{-1} spielt die Rolle einer **Korrelationszeit**. Für $t_1 = t_2 \gg \gamma^{-1}$ erhalten wir die stationären Geschwindigkeitsfluktuationen

$$\langle \vec{v}^2(t) \rangle = \frac{d\lambda}{m^2} \frac{1}{2\gamma} \quad (13.11)$$

In der statistischen Mechanik gilt nach dem Äquipartitionstheorem $\frac{1}{2}m\langle \vec{v}^2 \rangle = d\frac{k_B T}{2}$. Wenn die Mittelung über $\vec{\zeta}(t)$ gleich dem thermodynamischen Mittel sein soll, muss also gelten

$$\underbrace{\lambda}_{\text{Fluktuation}} = 2k_B T m \gamma = 2k_B T \underbrace{\Gamma}_{\text{Dissipation}} \quad (13.12)$$

Dies ist eine Version des **Fluktuations-Dissipations-Theorems**, das in seiner allgemeinen Form aussagt, dass die dissipative Antwort einer Größe auf eine kleine äußere Kraft proportional zu den thermischen (oder auch quantenmechanischen) Fluktuationen dieser Größe ist. In (13.12) ist der Reibungskoeffizient Γ der dissipativen Kraft, der die dissipative Antwort der Geschwindigkeit auf eine äußere Kraft beschreibt, proportional zu den Geschwindigkeitsfluktuationen, die von den Fluktuationen der stochastischen Kraft herrühren, und damit proportional zu λ sind. Das Fluktuations-Dissipations-Theorem zeigt den intuitiv erwarteten Zusammenhang zwischen stochastischer Kraft $\vec{\zeta}(t)$, Reibung Γ und Temperatur T , da alle drei Größen letztlich auf das umgebende fluide Bad zurückgehen.

Aus den Geschwindigkeitskorrelationen (13.10) können wir auch die **mittlere quadratische Schwankung der Teilchenposition** berechnen:

$$\begin{aligned} \langle (\vec{r}(t) - \vec{r}(0))^2 \rangle &= \int_0^t dt_1 \int_0^t dt_2 \langle \vec{v}(t_1) \cdot \vec{v}(t_2) \rangle \\ &= \left(\vec{v}^2(0) - \frac{d\lambda}{m^2} \frac{1}{2\gamma} \right) \frac{1}{\gamma^2} (1 - e^{-\gamma t})^2 + \frac{d\lambda}{m^2} \frac{1}{2\gamma} 2 \underbrace{\int_0^t dt_1 \int_0^{t_1} dt_2 e^{-\gamma(t_1-t_2)}}_{= \frac{t}{\gamma} - \frac{1}{\gamma^2} (1 - e^{-\gamma t})} \end{aligned}$$

Wir können zwei Regimes unterscheiden. Für Zeiten $t \ll \gamma^{-1}$ kleiner als die Korrelationszeit γ^{-1} finden wir sogenanntes **ballistisches Verhalten**

$$\langle (\vec{r}(t) - \vec{r}(0))^2 \rangle \approx \vec{v}^2(0)t^2 \quad \text{für } t \ll \gamma^{-1} \quad (13.13)$$

wo das Teilchen noch in Richtung der Anfangsgeschwindigkeit $\vec{v}(0)$ praktisch "geradeaus" fliegt. Für Zeiten $t \gg \gamma^{-1}$ größer als die Korrelationszeit finden wir **diffusives Verhalten**

$$\langle (\vec{r}(t) - \vec{r}(0))^2 \rangle \approx \frac{d\lambda}{m^2} \frac{1}{\gamma^2} t \quad \text{für } t \gg \gamma^{-1} \quad (13.14)$$

mit $\langle (\vec{r}(t) - \vec{r}(0))^2 \rangle \propto t$. Dies ist genau das **Diffusionsgesetz** $\langle (\vec{r}(t) - \vec{r}(0))^2 \rangle = 2dDt$ mit einer **Diffusionskonstanten** D , für die wir durch Vergleich mit (13.14) dann die **Einstein-Relation** (von 1905 [8]) finden:

$$D = \frac{\lambda}{2m^2\gamma^2} \stackrel{(13.12)}{=} \frac{k_B T}{m\gamma} = \frac{k_B T}{\Gamma} \quad (13.15)$$

die eine andere Form des Fluktuations-Dissipations-Theorems darstellt und die Diffusionskonstante D (die die Fluktuationen von \vec{r} beschreibt) mit dem Reibungskoeffizienten Γ verbindet.

Wenn wir den Inertialterm $m\ddot{\vec{r}}$ in der Langevin-Gleichung (13.4) vernachlässigen, erhalten wir den **überdämpften Limes** der Langevin Gleichung

$$\Gamma \dot{\vec{r}} = -\vec{\nabla}U(\vec{r}) + \vec{\zeta}(t) \quad (13.16)$$

Dieser Limes wird auch als **Brownsche Dynamik** bezeichnet.

Der überdämpfte Limes ist gerechtfertigt, wenn für ein Teilchen, dass sich typischerweise in einer Zeit Δt über eine Distanz δr bewegt,

$$|m\ddot{\vec{r}}| \sim m \frac{\Delta r}{\Delta t^2} \ll \Gamma \frac{\Delta r}{\Delta t} \sim |\Gamma \dot{\vec{r}}| \quad \text{oder}$$

$$\Delta t \gg m/\Gamma = \gamma^{-1} \quad (\text{überdämpfpter Limes})$$

gilt, was gleichbedeutend mit dem diffusiven Limes ist. Es ist instruktiv, einmal die typische Zeitskala γ^{-1} für ein μm großes kugelförmiges Teilchen von einer Dichte vergleichbar mit Wasser abzuschätzen:

$$R = 1\mu\text{m}, \quad \eta_{\text{H}_2\text{O}} = 10^{-3}\text{Pa s} = 10^{-3}\text{kg/ms}$$

$$m = \frac{4\pi}{3} \rho_{\text{H}_2\text{O}} R^3 = \frac{4\pi}{3} 10^{-15} \frac{\text{kg}}{\mu\text{m}^3} (1\mu\text{m})^3 \simeq 4 \cdot 10^{-15}\text{kg}$$

$$\gamma^{-1} = \frac{m}{6\pi\eta R} = \frac{2\rho_{\text{H}_2\text{O}} R^2}{9\eta_{\text{H}_2\text{O}}} \simeq 0.2 \cdot 10^{-6}\text{s}$$

Auf allen Zeitskalen $\Delta t \gg \gamma^{-1} \sim 10^{-6}\text{s}$ kann die Bewegung als überdämpft und diffusiv angesehen werden. Für ein cm großes Teilchen bekommt man dagegen $\gamma^{-1} \simeq 0.2 \cdot 10^2\text{s}$ und die Bewegung kann nur noch auf sehr langen Zeitskalen als überdämpft angesehen werden.

Im überdämpften Limes finden wir aus der Brownschen Dynamik (13.16) eine mittlere quadratische Schwankung der Teilchenposition

$$\langle (\vec{r}(t) - \vec{r}(0))^2 \rangle \stackrel{(13.16)}{=} \frac{1}{\Gamma^2} \int_0^t dt_1 \int_0^t dt_2 \langle \vec{\zeta}(t_1) \cdot \vec{\zeta}(t_2) \rangle$$

$$= \frac{1}{\Gamma^2} d\lambda t$$

im Einklang mit dem Langevin-Ergebnis (13.14) im diffusiven Limes. Allerdings gilt in der Brownschen Dynamik

$$\langle \vec{v}(t_1) \cdot \vec{v}(t_2) \rangle \propto \langle \vec{\zeta}(t_1) \cdot \vec{\zeta}(t_2) \rangle \propto \delta(t_1 - t_2)$$

d.h. die Geschwindigkeiten sind nicht mehr wohldefiniert im Gegensatz zu Formel (13.10) in der Langevin-Dynamik. Dies liegt daran, dass die stochastische Kraft $\vec{\zeta}(t)$ mit δ -Zeitkorrelationen (13.6) nicht mehr stetig ist in der Zeit. In der Brownschen Dynamik führt dies zu Unstetigkeiten bereits in der ersten Ableitung $\dot{\vec{r}} = \vec{v}$ in der Bewegungsgleichung erster Ordnung (13.16), während in der Langevin-Dynamik mit der Bewegungsgleichung zweiter Ordnung (13.4) nur die Beschleunigungen $\ddot{\vec{r}}$ unstetig werden.

13.1.2 N Teilchen

Nun wollen wir die Langevin-Dynamik auf N **wechselwirkende Teilchen** verallgemeinern, die sich in einem umgebenden Fluid mit Temperatur T bewegen. Für N Teilchen sind $\vec{r} = (\vec{r}_1, \dots, \vec{r}_N)$ und $\vec{\zeta}(t) = (\vec{\zeta}_1, \dots, \vec{\zeta}_N)$ entsprechend hoch-dimensionale Vektoren. Mit einer Energie $\mathcal{H} = \mathcal{H}(\{\vec{r}_k\})$ lautet die verallgemeinerte Kraft auf das k -te Teilchen $\vec{F}_k = -\vec{\nabla}_{\vec{r}_k} \mathcal{H}$. Die **Langevin-Gleichung**

für N Teilchen mit Koordinaten \vec{r}_k ($k = 1, \dots, N$) und Energie $\mathcal{H} = \mathcal{H}(\{\vec{r}_k\})$ lautet dann wie die Langevin-Gleichung für ein Teilchen (13.4) mit einem hochdimensionalen Vektor $\vec{r} = (\vec{r}_1, \dots, \vec{r}_N)$

$$m_k \ddot{\vec{r}}_k = -\Gamma_k \dot{\vec{r}}_k - \vec{\nabla}_{\vec{r}_k} \mathcal{H}(\{\vec{r}_k\}) + \vec{\zeta}_k(t) \quad (13.17)$$

mit gaußverteilter stochastischer Kraft $\vec{\zeta}_k(t)$

$$\begin{aligned} \langle \vec{\zeta}_k(t) \rangle &= 0 \\ \langle \zeta_{k,i}(t) \zeta_{l,j}(t') \rangle &= \lambda_k \delta_{kl} \delta_{ij} \delta(t - t') \end{aligned} \quad (13.18)$$

wobei i, j die räumlichen Komponenten der Kraft indizieren. Der Faktor δ_{kl} bedeutet, dass die stochastischen Kräfte **unabhängig** auf jedes Teilchen wirken, mit einer Stärke λ_k , die vom Teilchen abhängen kann. Dies ist plausibel, solange die Fluidteilchen, welche die stochastischen Kräfte auf *verschiedene* Teilchen k und l verursachen, als unkorreliert angesehen werden können. Dies wird verletzt, wenn hydrodynamische Strömungen im Fluid Korrelationen zwischen den Teilchen verursachen, sowohl in den stochastischen Kräften als auch in den Reibungskräften auf die einzelnen Teilchen. Solche **hydrodynamischen Wechselwirkungen** sind hier komplett vernachlässigt.

Das **Fluktuations-Dissipations-Theorem** (13.12),

$$\lambda_k = 2k_B T \Gamma_k \quad (13.19)$$

gilt für jedes Teilchen k einzeln unverändert und führt auch für N freie, nicht wechselwirkende Teilchen mit $\mathcal{H} = 0$ zur Übereinstimmung mit dem Äquipartitionstheorem $\frac{1}{2} m_k \langle \vec{v}_k^2 \rangle = d \frac{k_B T}{2}$ für jedes Teilchen.

Im **überdämpften Limes** erhalten wir wieder die **Brownsche Dynamik** für N Teilchen

$$\dot{\vec{r}}_k = \frac{1}{\Gamma_k} \left(-\vec{\nabla}_{\vec{r}_k} \mathcal{H}(\{\vec{r}_k\}) + \vec{\zeta}_k(t) \right) \quad (13.20)$$

Eine überdämpfte Brownsche Dynamik ist durchaus realistisch für Teilchen im μm -Bereich in wässriger Lösung. Für solch kleine Teilchen sind Inertialeffekte klein. Für annähernd runde Teilchen ist Γ_k dann auch durch die Stokes-Reibung (13.3) gegeben.

Eine weitere Annahme in (13.17) und (13.20) ist, dass die Reibungskräfte und stochastische Kräfte für verschiedene Teilchen unabhängig sind. Dies vernachlässigt eine mögliche **hydrodynamische Wechselwirkung** zwischen den N Teilchen. Dies bedeutet, dass das sich mit Geschwindigkeit \vec{v}_k bewegende Teilchen k ein Geschwindigkeitsfeld im umgebenden Fluid erzeugt, dass dann ein anderes Teilchen l spürt. Dies kann dazu führen, dass die Geschwindigkeit \vec{v}_l auch von den Kräften auf alle anderen Teilchen abhängt, $\vec{v}_k = \sum_l \underline{\mu}_{kl} \vec{F}_l$ mit einem sogenannten **Mobilitätstensor** $\underline{\mu}_{kl}$, der im Rahmen der Hydrodynamik berechnet werden kann. Dies führt auf den sogenannten Oseen Tensor. Wir benutzen in (13.17) und (13.20) nur die einfachste Approximation $\underline{\mu}_{kl} = \underline{\underline{\mu}}_{kl} \frac{1}{\Gamma_k}$, die in der Realität nur im Limes weit entfernter Teilchen (verdünnter Limes) korrekt ist.

13.2 Langevin- und Brownsche Dynamik Simulation

Bei der Langevin-Dynamik Simulation wird die Langevin-Gleichung numerisch gelöst (i.Allg. mit dem Euler-Verfahren), bei der Brownschen Dynamik Simulation wird im überdämpften Limes die Brownsche Dynamik numerisch gelöst.

13.2.1 Langevin-Dynamik Simulation

Wir formulieren die Langevin-Dynamik Simulation direkt allgemein für N Teilchen. Im letzten Kapitel 13.1 haben wir für das mittlere Geschwindigkeitsquadrat $\langle \vec{v}_k^2 \rangle$ eines Teilchens $k = 1, \dots, N$ und für den Spezialfall $\mathcal{H} = 0$ freier Teilchen gezeigt:

Wenn die stochastische Kraft $\vec{\zeta}_k(t)$ die Eigenschaften

$$\begin{aligned}\vec{\zeta}_k(t) &\text{ gaußverteilt mit} \\ \langle \vec{\zeta}_k(t) \rangle &= 0 \\ \langle \zeta_{k,i}(t) \zeta_{l,j}(t') \rangle &= 2k_B T \Gamma_k \delta_{kl} \delta_{ij} \delta(t - t')\end{aligned}\quad (13.21)$$

(i, j räumliche Komponenten der Kraft; k, l Teilchenindizes) besitzt (wir beachten, dass wir in (13.21) bereits das Fluktuations-Dissipations-Theorem (13.19) erfüllt haben), dann gilt

$$\begin{aligned}\langle \dots \rangle &= \text{Mittel über Realisationen von } \vec{\zeta}_k(t) \\ &= \text{Mittel über viele Trajektorien der Langevin-Gleichung} \\ &= \text{thermodynamisches Mittel bei Temperatur } T\end{aligned}\quad (13.22)$$

Wir werden dies im nächsten Kapitel 13.3 (zumindest für ein Teilchen $N = 1$) ganz allgemein für alle Observablen und beliebige Potentiale zeigen.

Wegen (13.22) können thermodynamische Mittelwerte $\langle O \rangle$ von Observablen O bei der Temperatur T aus einer numerischen Lösung der Langevin-Gleichung

$$m_k \ddot{\vec{r}}_k = -\Gamma_k \dot{\vec{r}}_k - \vec{\nabla}_{\vec{r}_k} \mathcal{H}(\{\vec{r}_k\}) + \vec{\zeta}_k(t)\quad (13.23)$$

berechnet werden, wenn $\vec{\zeta}_k(t)$ (13.21) erfüllt. Dies ist die **Langevin-Dynamik Simulation**:

- 1) Wir lösen die gewöhnliche DGL (13.23) numerisch mit einer Methode aus Kapitel 4 (i.Allg. reicht hier ein **Euler-Verfahren**, siehe unten) für eine zufällige Realisation $\vec{\zeta}_k(t)$ der stochastischen Kraft, die (13.21) erfüllt. Daraus erhalten wir eine (in der Zeit diskrete) Trajektorie $\vec{r}_k(t)$ und $\vec{v}_k(t)$ als Lösung.
- 2) Dies ist die Grundlage der Mittelung $\langle \dots \rangle$ einer Observable $O = O(\{\vec{r}_k, \vec{v}_k\})$, die wir auf zwei Arten durchführen können:
 - a) Wir führen S Simulationen 1) mit verschiedenen Realisationen $\vec{\zeta}_k(t)$ der stochastischen Kraft durch. Das Ergebnis sind S Trajektorien $\vec{r}_{k,s}(t)$ und $\vec{v}_{k,s}(t)$ ($s = 1, \dots, S$). Dann ist $\langle \dots \rangle$ das **Mittel über diese Trajektorien**

$$\langle O(\{\vec{r}_k, \vec{v}_k\}) \rangle = \langle O(\{\vec{r}_k(t), \vec{v}_k(t)\}) \rangle = \frac{1}{S} \sum_{s=1}^S O(\{\vec{r}_{k,s}(t), \vec{v}_{k,s}(t)\})$$

Diese Methode erlaubt offensichtlich auch die Mittelung **zeitabhängiger** Observablen $O(\{\vec{r}_k(t), \vec{v}_k(t)\}, t)$ und damit die **Simulation von Dynamik** bzw. von statistischer Physik im **Nicht-Gleichgewicht**.

- 1) Die andere Methode basiert auf der **Ergodizitätsannahme**: **Zeitunabhängige** Observablen $O(\{\vec{r}_k, \vec{v}_k\})$ können auch gemittelt werden, indem wir eine lange Simulation durchführen und eine **Mittelung über die Zeit** $t = n\Delta t$ durchführen:

$$\langle O(\{\vec{r}_k, \vec{v}_k\}) \rangle = \frac{\Delta t}{T} \sum_{n=1}^{T/\Delta t} O(\{\vec{r}_k(n\Delta t), \vec{v}_k(n\Delta t)\})$$

Für zeitunabhängige Observablen sind auch Kombinationen von a) und b) möglich.

Bei der numerischen Integration der gewöhnlichen DGL in 1) genügt ein einfaches **Euler-Verfahren**: Weil die stochastische Kraft $\vec{\zeta}_k(t)$ unstetig in der Zeit ist (wegen der δ -artigen Korrelationen), haben Verfahren höherer Ordnung i.Allg. keinen Vorteil. Die Herleitung dieser Verfahren – wie beispielsweise Runge-Kutta Methoden in Kapitel 4 – beruhten auf der Annahme, dass die rechte Seite der DGL mindestens stetig ist.

Wir formulieren die numerische Euler-Integration für $N = 1$ Teilchen (der Übersichtlichkeit halber, für N Teilchen kommen noch entsprechende Indizes k wieder dazu) einmal explizit. Für Zeitschritte $t_n = n\Delta t$ mit $\vec{r}_n = \vec{r}(n\Delta t)$, $\vec{v}_n = \vec{v}(n\Delta t)$, $\vec{F}_n = -\vec{\nabla}U(\vec{r}_n)$ und $\vec{\zeta}_n = \vec{\zeta}(n\Delta t)$ nimmt das Euler-Verfahren (4.8) folgende Form an:

$$\boxed{\begin{aligned}\vec{r}_{n+1} &= \vec{r}_n + \vec{v}_n \Delta t \\ m\vec{v}_{n+1} &= m\vec{v}_n + \vec{F}_n \Delta t - \Gamma \vec{v}_n \Delta t - \underbrace{\vec{\zeta}_n \Delta t}_{\equiv \vec{R}_n}\end{aligned}} \quad (13.24)$$

Um (13.21) zu erfüllen, muss für die zeitdiskretisierte stochastische Kraft $\vec{\zeta}_n$ dabei gelten

$$\begin{aligned}\vec{\zeta}_{n,i} &\text{ gaußverteilt mit} \\ \langle \zeta_{n,i} \rangle &= 0 \\ \langle \zeta_{n,i} \zeta_{m,j} \rangle &= \lambda \delta_{ij} \delta_{nm} \frac{1}{\Delta t} = \frac{2k_B T \Gamma}{\Delta t} \delta_{ij} \delta_{nm}\end{aligned}$$

Also ziehen wir in jedem Zeitschritt neue **zufällige gaußverteilte** Komponenten $R_{n,i}$, z.B. mit dem Box-Muller Algorithmus (10.11), mit Mittelwert $\langle R_{n,i} \rangle = 0$ und Varianz $\sigma_R = \langle R_{n,i}^2 \rangle = 2k_B T \Gamma \Delta t$.

Für hinreichend kleines Δt addieren sich die zufälligen $R_{n,i}$ im Laufe der Zeit ohnehin zu Gaußverteilungen (zentraler Grenzwertsatz). Daher reicht es normalerweise, in jedem Zeitschritt **gleichverteilte** $R_{n,i}$ aus $[-\sqrt{6k_B T \Gamma \Delta t}, +\sqrt{6k_B T \Gamma \Delta t}]$ zu ziehen, so dass die Varianz wieder $\sigma_R = 2k_B T \Gamma \Delta t$ ist.

13.2.2 Brownsche Dynamik Simulation

Die Brownsche Dynamik Simulation stellt den **überdämpften Limes** der Langevin-Dynamik Simulation dar. Entsprechend vernachlässigen wie die Inertialterme und lösen die Gleichung der Brownschen Dynamik numerisch:

$$\boxed{\dot{\vec{r}}_k = \frac{1}{\Gamma_k} \left(-\vec{\nabla}_{\vec{r}_k} \mathcal{H}(\{\vec{r}_k\}) + \vec{\zeta}_k(t) \right)} \quad (13.25)$$

wobei $\vec{\zeta}_k(t)$ wieder (13.21) erfüllen muss.

Die Vorgehensweisen zur Mittelung von Observablen sind völlig identisch zur Langevin-Dynamik Simulation.

Abschließend wollen wir die Langevin- oder Brownsche Dynamik Simulation mit der MD-Simulation vergleichen. In beiden Methoden wird die Bewegungsgleichung von Teilchen gelöst. Allerdings würden im Rahmen einer MD-Simulation die Teilchen des umgebenden Fluids mit Temperatur T explizit mitsimuliert werden müssen, um die Reibungskraft und die thermische Kraft zu erzeugen. Bei der Langevin-Dynamik Simulation wird das umgebende Fluid nicht explizit mitsimuliert, sondern sein Einfluss nur durch Reibung und stochastische thermische Kraft berücksichtigt. Dies ist natürlich weitaus effektiver, allerdings hat man auch keinerlei Information mehr über die mikroskopische Dynamik der Fluidteilchen im umgebenden Bad.

13.3 Fokker-Planck-Gleichungen

Fokker-Planck-Gleichung (Rayleigh-Gleichung), Klein-Kramers-Gleichung und Smoluchowski-Gleichung sind Bewegungsgleichungen (partielle DGLn) für die Wahrscheinlichkeitsverteilungen, die den stochastischen Langevin-Gleichungen oder Brownscher Dynamik (gewöhnliche DGLn) entsprechen. Das thermodynamisches Mittel entspricht dann einem Mittel mit stationärer Wahrscheinlichkeitsverteilung. Diese partiellen DGLn können mit den Methoden aus Kapitel 6 auch numerisch gelöst werden.

Bei der Beschreibung stochastischer Bewegung oder stochastischer Prozesse gibt es grundsätzlich zwei Möglichkeiten:

1) **Langevin-Gleichungen:**

Hier formulieren wir, wie in den letzten Kapiteln beschrieben, eine **stochastische mikroskopische Bewegungsgleichung**, z.B. für ein Teilchen die Langevin-Gleichung (13.4). Dies ist eine **gewöhnliche DGL** in t für die Trajektorie $\vec{r}(t)$. Um **Mittelungen** (...) analytisch oder numerisch durchzuführen, müssen wir eine Lösung finden für eine beliebige gegebene Realisation der stochastischen Kraft $\vec{\zeta}(t)$ und anschließend über all Realisationen der stochastischen Kraft $\vec{\zeta}(t)$ mitteln.

2) **Fokker-Planck-Gleichungen:**

Hier formulieren wir eine **deterministische Bewegungsgleichung**, allerdings für **Wahrscheinlichkeitsverteilungen**, z.B. für die Wahrscheinlichkeit $P(\vec{r}, \vec{v}, t)$, ein Teilchen zur Zeit t am Ort \vec{r} , mit Geschwindigkeit \vec{v} anzutreffen. Dies ist dann eine **partielle DGL**, allerdings ist die DGL deterministisch, Mittelungen über die stochastische Kraft sind bereits in der DGL selbst ausgeführt worden.

Dies verhält sich analog wie bei Markov-Prozessen, siehe Kapitel 11.2: 1) entspricht einem Markov-Prozess, in dem für jeden Übergang $i \rightarrow j$ eine Übergangswahrscheinlichkeit M_{ij} spezifiziert ist. Er beschreibt eine stochastische Übergangsdynamik im Zustandsraum. 2) entspricht der Master-Gleichung für die Wahrscheinlichkeitsverteilung $p_i(t)$. Hier haben wir jetzt lediglich *kontinuierliche Zustände* im Raum der Orte \vec{r} und Geschwindigkeiten \vec{v} anstatt einem diskreten Zustandsraum.

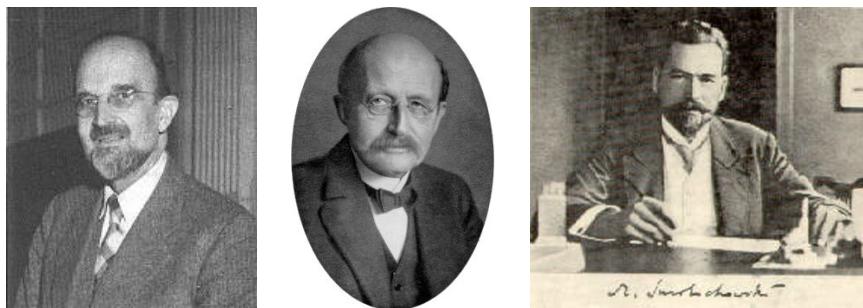


Abbildung 13.4: Links: Adriaan Daniël Fokker (1887-1972), niederländischer Physiker (und Musiker), Cousin des Flugzeugbauers. In seiner Promotion bei Max Planck (1858-1947), mittleres Bild, formulierte er die Fokker-Planck-Gleichung. Rechts: Marian Smoluchowski (1872-1917), polnischer Physiker.

Wir wollen im Folgenden verschiedene Varianten von **Fokker-Planck-Gleichungen** herleiten. Dabei beschränken wir uns auf **ein Teilchen** ($N = 1$). Die Rechnungen für N Teilchen wären wieder analog mit entsprechend höherdimensionalen Vektoren, was mehr Schreibaufwand erfordert und

etwas unübersichtlich würde.

13.3.1 Fokker-Planck-Gleichung (Rayleigh-Gleichung)

Zunächst betrachten wir ein **freies Teilchen** ($U = 0$) mit der **Langevin-Gleichung**

$$m\dot{\vec{v}} = -\Gamma\vec{v} + \vec{\zeta}(t) \quad (13.26)$$

Die **Wahrscheinlichkeitsverteilung** $P(\vec{v}, t)$, das Teilchen zur Zeit t mit einer Geschwindigkeit \vec{v} anzutreffen lässt sich als Mittelwert einer entsprechenden δ -Funktion formulieren:

$$P(\vec{v}, t) = \langle \delta(\vec{v} - \vec{v}(t)) \rangle \quad (13.27)$$

wobei $\vec{v}(t)$ eine stochastische Trajektorie des Teilchens nach der Langevin-Gleichung (13.26) beschreibt. Wir sehen, dass die Mittelung über die Realisationen der stochastischen Kraft unmittelbar in die Definition der Wahrscheinlichkeitsverteilung selbst eingehen:

- Wenn $P(\vec{v}, t)$ bekannt ist, kann jede Mittelung als

$$\langle O(\vec{v}(t), t) \rangle = \int d^d \vec{v} O(\vec{v}(t), t) P(\vec{v}, t)$$

durchgeführt werden.

- Die Formulierung (13.27) stellt sicher, dass $P(\vec{v}, t)$ zu jeder Zeit t normiert sein wird:
 $\int d^d \vec{v} P(\vec{v}, t) = 1$.

Nun werden wir eine partielle DGL für die Verteilung $P(\vec{v}, t)$ herleiten. Dazu schreiben wir die Langevin-Gleichung als

$$\dot{\vec{v}} = -\gamma\vec{v} + \vec{\eta}(t) \quad \text{mit} \quad \gamma \equiv \Gamma/m \quad \text{und} \quad \vec{\eta}(t) \equiv \vec{\zeta}(t)/m \quad (13.28)$$

wie in (13.8) und differenzieren $P(\vec{v}, t)$ an der Stelle $\vec{v} = \vec{v}_0$ nach der Zeit, indem wir zunächst den Differenzenquotienten mit kleinem $\Delta t \rightarrow 0$ bilden

$$\begin{aligned} \partial_t P(\vec{v}_0, t) &\approx \frac{1}{\Delta t} [\langle \delta(\vec{v}_0 - \vec{v}(t + \Delta t)) \rangle - \langle \delta(\vec{v}_0 - \vec{v}(t)) \rangle] \\ &\approx \frac{1}{\Delta t} \left[-\vec{\nabla}_{\vec{v}_0} \cdot \langle \delta(\vec{v}_0 - \vec{v}(t)) \Delta \vec{v}(t) \rangle + \sum_{i,j} \frac{1}{2} \frac{\partial^2}{\partial v_{0,i} \partial v_{0,j}} \langle \delta(\vec{v}_0 - \vec{v}(t)) \Delta v_i(t) \Delta v_j(t) \rangle \right] \end{aligned} \quad (13.29)$$

wo wir in $\Delta \vec{v}(t) = \vec{v}(t + \Delta t) - \vec{v}(t)$ entwickelt haben. Im Limes $\Delta t \rightarrow 0$ überleben nur Terme bis $\mathcal{O}(\Delta t)$ in der Klammer [...], die Frage ist, welche Beiträge das eigentlich sind. Dazu machen wir uns zuerst klar von welcher Ordnung eigentlich $\Delta \vec{v}(t)$ ist:

$$\begin{aligned} \Delta \vec{v}(t) &= \vec{v}(t + \Delta t) - \vec{v}(t) = \int_t^{t+\Delta t} d\tau \dot{\vec{v}}(\tau) \\ &\stackrel{(13.28)}{=} \int_t^{t+\Delta t} d\tau (-\gamma \vec{v}(\tau) + \vec{\eta}(\tau)) \\ &= \underbrace{-\gamma \vec{v}(\tau) \Delta t}_{\mathcal{O}(\Delta t)} + \underbrace{\int_t^{t+\Delta t} d\tau \vec{\eta}(\tau)}_{\mathcal{O}(\Delta t^{1/2})!!} + \mathcal{O}(\Delta t^2) \end{aligned} \quad (13.30)$$

weil

$$\int_t^{t+\Delta t} d\tau_1 \int_t^{t+\Delta t} d\tau_2 \underbrace{\langle \eta_i(\tau_1) \eta_j(\tau_2) \rangle}_{\frac{\lambda}{m^2} \delta_{ij} \delta(\tau_1 - \tau_2)} = \frac{\lambda}{m^2} \delta_{ij} \Delta t = \mathcal{O}(\Delta t) \quad (13.31)$$

Also ist $\Delta \vec{v}(t)$ tatsächlich von der Ordnung $\mathcal{O}(\Delta t^{1/2})$, d.h. *beide* Terme in der Klammer [...] in (13.29) tragen in der Ordnung $\mathcal{O}(\Delta t)$ bei und müssen mitgenommen werden:

$$\begin{aligned} \partial_t P(\vec{v}_0, t) &\stackrel{(13.30)}{\approx} \frac{1}{\Delta t} \left[-\vec{\nabla}_{\vec{v}_0} \cdot \left\langle \delta(\vec{v}_0 - \vec{v}(t)) \left(-\gamma \vec{v}(\tau) \Delta t + \int_t^{t+\Delta t} d\tau \vec{\eta}(\tau) \right) \right\rangle + \right. \\ &+ \sum_{i,j} \frac{1}{2} \frac{\partial^2}{\partial v_{0,i} \partial v_{0,j}} \left\langle \delta(\vec{v}_0 - \vec{v}(t)) \left(\int_t^{t+\Delta t} d\tau_1 \int_t^{t+\Delta t} d\tau_2 \eta_i(\tau_1) \eta_j(\tau_2) \right) \right\rangle + \mathcal{O}(\Delta t^{3/2}) \left. \right] \end{aligned}$$

Auf Grund der *Kausalität* kann $\vec{v}(t)$ nur von $\vec{\eta}(\tau)$ abhängen mit $\tau < t$. Daher sollten $\vec{v}(t)$ und $\int_t^{t+\Delta t} d\tau \vec{\eta}(\tau)$ unkorreliert sein und es folgt:

$$\left\langle \delta(\vec{v}_0 - \vec{v}(t)) \left(\int_t^{t+\Delta t} d\tau \vec{\eta}(\tau) \right) \right\rangle = \langle \delta(\vec{v}_0 - \vec{v}(t)) \rangle \underbrace{\left\langle \left(\int_t^{t+\Delta t} d\tau \vec{\eta}(\tau) \right) \right\rangle}_{=0} = 0$$

und

$$\begin{aligned} &\left\langle \delta(\vec{v}_0 - \vec{v}(t)) \left(\int_t^{t+\Delta t} d\tau_1 \int_t^{t+\Delta t} d\tau_2 \eta_i(\tau_1) \eta_j(\tau_2) \right) \right\rangle \\ &= \langle \delta(\vec{v}_0 - \vec{v}(t)) \rangle \left\langle \left(\int_t^{t+\Delta t} d\tau_1 \int_t^{t+\Delta t} d\tau_2 \eta_i(\tau_1) \eta_j(\tau_2) \right) \right\rangle \\ &\stackrel{(13.31)}{=} \langle \delta(\vec{v}_0 - \vec{v}(t)) \rangle \frac{\lambda}{m^2} \delta_{ij} \Delta t \end{aligned}$$

Damit erhalten wir mit der Definition (13.27) von $P(\vec{v}_0, t)$

$$\partial_t P(\vec{v}_0, t) = \vec{\nabla}_{\vec{v}_0} \cdot (\gamma \vec{v}_0 P(\vec{v}_0, t)) + \frac{1}{2} \vec{\nabla}_{\vec{v}_0}^2 \left(\frac{\lambda}{m^2} P(\vec{v}_0, t) \right)$$

und damit schließlich wieder eine kompakte Gleichung

$$\partial_t P(\vec{v}, t) = \frac{\Gamma}{m} \vec{\nabla}_{\vec{v}} \cdot (\vec{v} P(\vec{v}, t)) + \frac{\lambda}{2m^2} \vec{\nabla}_{\vec{v}}^2 P(\vec{v}, t) \quad (13.32)$$

Dies ist die **Fokker-Planck-Gleichung** oder auch **Rayleigh-Gleichung**. Nach dem Fluktuations-Dissipations-Theorem bzw. der Einstein-Relation (13.15) gilt $\lambda^2/2m^2 = D(\Gamma/m)^2$ für den Koeffizienten im letzten Term.

Wir können die Fokker-Planck-Gleichung (13.32) auch als **Kontinuitätsgleichung** schreiben

$$\begin{aligned} \partial_t P(\vec{v}, t) &= -\vec{\nabla}_{\vec{v}} \cdot \vec{j}(\vec{v}, t) \quad \text{mit} \\ \vec{j}(\vec{v}, t) &= -\frac{\Gamma}{m} \vec{v} P(\vec{v}, t) - \frac{\lambda}{2m^2} \vec{\nabla}_{\vec{v}} P(\vec{v}, t), \end{aligned} \quad (13.33)$$

was einen **Strom** im Geschwindigkeitsraum darstellt.

Im **thermodynamischen Gleichgewicht** sollten alle Ströme verschwinden (detailed balance), d.h. $\vec{j}(\vec{v}, t) = 0$ gelten. Dies führt dann nach der Kontinuitätsgleichung auf eine **stationäre Gleichgewichtsverteilung** $P_{eq}(\vec{v})$ ($\partial_t P_{eq} = 0$) mit

$$\vec{\nabla}_{\vec{v}} P_{eq}(\vec{v}) = -m\vec{v} \frac{2\Gamma}{\lambda} P_{eq}(\vec{v})$$

was durch

$$P_{eq}(\vec{v}) = \mathcal{N} \exp\left(-\frac{1}{2}mv^2 \frac{2\Gamma}{\lambda}\right) \quad (13.34)$$

gelöst wird (\mathcal{N} ist ein Normierungsfaktor). Dies ist aber gerade die **Boltzmannverteilung**, wenn das **Fluktuations-Dissipations-Theorem** (13.12) gilt:

$$\lambda = 2k_B T \Gamma$$

Damit haben wir dann (für $U = 0$) wieder das Fluktuations-Dissipations-Theorem gezeigt und darüberhinaus ganz allgemein, dass $\langle \dots \rangle$ identisch ist mit dem thermodynamischen Mittel mit der kanonischen Boltzmannverteilung.

13.3.2 Klein-Kramers-Gleichung

Für die **allgemeine Langevin-Gleichung** (13.4)

$$m\ddot{\vec{r}} = -\Gamma\dot{\vec{r}} - \vec{\nabla}U(\vec{r}) + \vec{\zeta}(t)$$

mit Kräften $-\vec{\nabla}U \neq 0$ betrachtet man analog eine **Wahrscheinlichkeitsverteilung** $P(\vec{r}, \vec{v}, t)$, das Teilchen zur Zeit t mit einer Geschwindigkeit \vec{v} und am Ort \vec{r} anzutreffen

$$P(\vec{v}, t) = \langle \delta(\vec{r} - \vec{r}(t))\delta(\vec{v} - \dot{\vec{r}}(t)) \rangle \quad (13.35)$$

Eine längere Rechnung nach dem gleichen Schema wie für die Fokker-Planck-Gleichung führt in diesem Fall auf die sogenannte **Klein-Kramers-Gleichung**

$$\partial_t P(\vec{r}, \vec{v}, t) = -\vec{v} \cdot \vec{\nabla}_{\vec{r}} P + \vec{\nabla}_{\vec{v}} \cdot \left(\frac{\Gamma}{m} \vec{\nabla}_{\vec{v}} + \frac{1}{m} \vec{\nabla}_{\vec{r}} U \right) P + \frac{\lambda}{2m^2} \vec{\nabla}_{\vec{v}}^2 P \quad (13.36)$$

wo wir nach dem Fluktuations-Dissipationstheorem $\lambda^2/2m^2 = D(\Gamma/m)^2$ finden für den letzten Koeffizienten.

Wir stellen fest, dass für $P(\vec{v}, t) = \int d^d\vec{r} P(\vec{r}, \vec{v}, t)$ und $\vec{\nabla}U = 0$ sich wieder die Fokker-Planck-Gleichung (13.32) aus der Klein-Kramers-Gleichung (13.36) ergibt.

Im **thermodynamischen Gleichgewicht** gilt auch wieder detailed balance und $\partial_t P_{eq} = 0$ mit einer **stationären Gleichgewichtsverteilung**

$$P_{eq}(\vec{r}, \vec{v}) = \mathcal{N} \exp\left(-\left(\frac{1}{2}mv^2 + U(\vec{r})\right) \frac{2\Gamma}{\lambda}\right) \quad (13.37)$$

(\mathcal{N} ist ein Normierungsfaktor). Dies ist wieder gerade die **Boltzmannverteilung**, wenn das **Fluktuations-Dissipations-Theorem** (13.12) gilt. Damit haben wir dann ganz allgemein gezeigt, dass $\langle \dots \rangle$ identisch ist mit dem thermodynamischen Mittel mit der kanonischen Boltzmannverteilung.

13.3.3 Smoluchowski-Gleichung

Für die Brownsche Dynamik im **überdämpften Limes** ($m\ddot{r} \ll \Gamma\dot{r}$) betrachtet man eine **Wahrscheinlichkeitsverteilung**

$$P(\vec{r}, t) = \langle \delta(\vec{r} - \vec{r}(t)) \rangle \quad (13.38)$$

das Teilchen zur Zeit t am Ort \vec{r} anzutreffen.

Mit analogen Methoden wie für die Fokker-Planck-Gleichung leitet man die **Smoluchowski-Gleichung**

$$\partial_t P(\vec{r}, t) = \frac{1}{\Gamma} \vec{\nabla}_{\vec{r}} \left[(\vec{\nabla}_{\vec{r}} U) P \right] + \frac{\lambda}{2\Gamma^2} \vec{\nabla}_{\vec{r}}^2 P \quad (13.39)$$

her, wobei $\lambda/2\Gamma^2 = D$ im letzten Term. Wir sehen hier, dass im Spezialfall $U = 0$ sich deshalb genau die bekannt **Diffusionsgleichung** $\partial_t P = D\vec{\nabla}_{\vec{r}}^2 P$ ergibt.

Auch die Smoluchowski-Gleichung lässt sich als **Kontinuitätsgleichung** schreiben

$$\begin{aligned} \partial_t P(\vec{r}, t) &= -\vec{\nabla}_{\vec{r}} \cdot \vec{j}(\vec{r}, t) \quad \text{mit} \\ \vec{j}(\vec{r}, t) &= \underbrace{-D\vec{\nabla}_{\vec{r}} P}_{\text{diffusiver Strom}} \quad \underbrace{-\frac{1}{\Gamma}(\vec{\nabla}_{\vec{r}} U) P}_{\text{Drift-Strom}} \end{aligned} \quad (13.40)$$

wobei der Drift-Strom durch die äußere Kraft $-\vec{\nabla}_{\vec{r}} U$ hervorgerufen wird und mit Geschwindigkeit $-\vec{\nabla}_{\vec{r}} U / \Gamma$ strömenden Teilchen entspricht. Im **thermodynamischen Gleichgewicht** sollten alle Ströme verschwinden (detailed balance), d.h. $\vec{j}(\vec{r}, t) = 0$ gelten. Dies führt dann nach der Kontinuitätsgleichung auf eine **stationäre Gleichgewichtsverteilung** $P_{eq}(\vec{r})$ ($\partial_t P_{eq} = 0$) mit

$$P_{eq}(\vec{v}) = \mathcal{N} \exp(-U(\vec{r})/k_B T) \quad (13.41)$$

also genau die **Boltzmannverteilung** im Ortsraum (ohne kinetische Energie in der Exponentialfunktion, da wir den Inertialterm vernachlässigt haben: Teilchen kommen im überdämpften Limes *sofort* zur Ruhe ohne äußere Kraft und haben deshalb keine kinetische Energie), wobei wir das **Fluktuations-Dissipations-Theorem** (13.12), $\lambda = 2k_B T \Gamma$, bzw. die **Einstein-Relation** (13.15), $D = k_B T / \Gamma$ benutzt haben. Damit gilt auch für die Brownsche Dynamik, dass $\langle \dots \rangle$ identisch ist mit dem thermodynamischen Mittel mit der kanonischen Boltzmannverteilung.

13.3.4 Numerische Lösung von Fokker-Planck-Gleichungen

Natürlich können wir auch die Fokker-Planck-, Klein-Kramers- bzw. Smoluchowski-Gleichung numerisch lösen, um die stochastische Dynamik des Systems zu simulieren. Dazu können die Methoden aus Kapitel 6 zur numerischen Lösung von **partiellen DGLn** verwenden. Alle diese Gleichungen sind von der Form von Kontinuitätsgleichungen, wie wir bereits gezeigt haben. Klein-Kramers- und Smoluchowski-Gleichung reduzieren sich auf die Diffusionsgleichung für den freien Fall $U = 0$.

Daher können ähnliche Diskretisierungsmethoden wie bei der Diffusionsgleichung verwendet werden, also ein **explizites FTCS-Schema (forward in time, centered in space)**, das **explizite Lax-Schema** oder das **implizite Crank-Nicolson-Schema**.

Wenn sich eine Fokker-Planck-Gleichung als kontinuierlicher Grenzfall einer auf einem Gitter definierten Master-Gleichung ergibt, sollte auch die diskrete Master-Gleichung selbst als die “physikalische” Diskretisierung des Problems in der numerischen Lösung verwendet werden.

13.4 Literaturverzeichnis Kapitel 13

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [4] D. P. Landau und K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2005.
- [5] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [6] F. Schwabl. *Statistische Mechanik*. Springer-Lehrbuch. Springer, 2006.
- [7] R. Brown. *A brief account of microscopical observations made in the months 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 inorganic bodies*. Philosophical Magazine Series 2 **4** (1828), 161–173.
- [8] A. Einstein. *Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen*. Annalen der Physik **322** (1905), 549–560.

Literaturverzeichnis

- [1] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. 2nd. (2nd edition freely available online). New York, NY, USA: Cambridge University Press, 1992.
- [2] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. 3rd. New York, NY, USA: Cambridge University Press, 2007.
- [3] S. Koonin und D. Meredith. *Computational Physics: Fortran Version*. Redwood City, Calif, USA: Addison-Wesley, 1998.
- [4] W. Kinzel und G. Reents. *Physics by Computer*. 1st. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
- [5] D. Frenkel und B. Smit. *Understanding Molecular Simulation*. 2nd. Orlando, FL, USA: Academic Press, Inc., 2001.
- [6] H. Gould, J. Tobochnik und C. Wolfgang. *An Introduction to Computer Simulation Methods: Applications to Physical Systems (3rd Edition)*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [7] J. Thijssen. *Computational Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2007.
- [8] D. P. Landau und K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. 2nd. New York, NY, USA: Cambridge University Press, 2005.
- [9] J. Stoer, R. Bartels, W. Gautschi, R. Bulirsch und C. Witzgall. *Introduction to Numerical Analysis*. 3rd. Texts in Applied Mathematics. New York, NY, USA: Springer, 2013.
- [10] R. W. Hamming. *Numerical Methods for Scientists and Engineers*. 2nd. New York, NY, USA: Dover Publications, Inc., 1986.
- [11] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Studies in nonlinearity. Westview Press, 2008.
- [12] G. H. Golub und C. F. Van Loan. *Matrix Computations*. 3rd. Johns Hopkins Studies in the Mathematical Sciences. Baltimore, Maryland, USA: Johns Hopkins University Press, 1996.
- [13] M. Hjorth-Jensen. *Computational Physics (Skript)*. Oslo: University of Oslo, 2012.
- [14] R. Fitzpatrick. *Computational Physics (Skript)*. Austin, Texas: The University of Texas at Austin, 2012.
- [15] W. Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Statistical, Computational, and Theoretical Physics. Oxford University Press, 2006.
- [16] C. Moore. *A complex legacy*. Nature Phys. **7** (2011), 828–830.
- [17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller und E. Teller. *Equation of State Calculations by Fast Computing Machines*. J. Chem. Phys. **21** (1953), 1087–1092.
- [18] N. Metropolis. *The beginning of the Monte Carlo method*. Los Alamos Science **15** (1987), 125–130.
- [19] H. L. Anderson. *Metropolis, Monte Carlo, and the MANIAC*. Los Alamos Science (1986), 96–108.

- [20] E. Fermi, J. Pasta und S. Ulam. *Studies of nonlinear problems*. LASL Report LA-1940 (1955).
- [21] T. Dauxois, M. Peyrard und S. Ruffo. *The Fermi–Pasta–Ulam ‘numerical experiment’: history and pedagogical perspectives*. Eur. J. Phys. **26** (2005), S3–S11.
- [22] B. Alder und T. Wainwright. *Phase Transition for a Hard Sphere System*. J. Chem. Phys. **27** (1957), 1208–1211.
- [23] C. Dellago und H. A. Posch. *Realizing Boltzmann’s dream: computer simulations in modern statistical mechanics*. In: *Boltzmann’s Legacy*. Hrsg. von G. Gallavotti, W. Reiter und J. Yngvason. Zuerich, Switzerland: European Mathematical Society Publishing House, Okt. 2008, 171–202.
- [24] G. Uhlenbeck. *Round Table on Statistical Mechanics*. In: *The many-body problem*. Hrsg. von J. Percus. London: Interscience Publishers/John Wiley, 1963. Kap. XXVIII, 493–509.
- [25] F. Benford. *The Law of Anomalous Numbers*. Proceedings of the American Philosophical Society **78** (1938), 551–572.
- [26] N. Hüngerbühler. *Benfords Gesetz über führende Ziffern : Wie die Mathematik Steuersündern das Fürchten lehrt*. 2007.
- [27] T. Hill. *The First Digit Phenomenon*. American Scientist **86** (1998), 358.
- [28] B. F. Roukema. *A first-digit anomaly in the 2009 Iranian presidential election*. J. Appl. Stat. **41** (Jan. 2014), 164–199.
- [29] J. D. Jackson. *Classical electrodynamics*. 3rd ed. New York, NY: Wiley, 1999.
- [30] L. Verlet. *Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules*. Phys. Rev. **159** (1967), 98–103.
- [31] M. Tuckerman, G. J. Martyna und B. J. Berne. *Reversible multiple time scale molecular dynamics*. J. Chem. Phys. **97** (1992), 1990–2001.
- [32] L. Schulman. *Techniques and Applications of Path Integration*. Wiley, 1996.
- [33] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, a. DiNola und J. R. Haak. *Molecular dynamics with coupling to an external bath*. J. Chem. Phys. **81** (1984), 3684–3690.
- [34] S. Nosé. *A unified formulation of the constant temperature molecular dynamics methods*. J. Chem. Phys. **81** (1984), 511–519.
- [35] W. G. Hoover. *Canonical dynamics: Equilibrium phase-space distributions*. Phys. Rev. A **31** (1985), 1695–1697.
- [36] M. J. Feigenbaum. *Quantitative Universality for a Class of Nonlinear Transformations*. J. Stat. Phys. **19** (1978), 25–52.
- [37] H. Korsch und H. Jodl. *Chaos: a program collection for the PC*. Bd. 1. Springer, 1999.
- [38] V. Arnol’d. *Mathematical Methods of Classical Mechanics*. Graduate Texts in Mathematics. Springer, New York, 1997.
- [39] L. Page. *Method for node ranking in a linked database*. US Patent 6,285,999. Sep. 2001.
- [40] E. A. Hylleraas und B. Undheim. *Numerische Berechnung der 2 S-Terme von Ortho- und Par-Helium*. Z. Physik **65** (1930), 759–772.
- [41] L. Noll, R. Mende und S. Sisodiya. *Method for seeding a pseudo-random number generator with a cryptographic hash of a digitization of a chaotic system*. US Patent 5,732,138. März 1998.
- [42] G. Marsaglia. *Random numbers fall mainly in the planes*. Proce. Nat. Acad. Sci. U.S.A. **61** (1968), 25–28.
- [43] E. Ising. *Beitrag zur Theorie des Ferromagnetismus*. Z. Phys. **31** (1925), 253–258.
- [44] R. J. Glauber. *Time-Dependent Statistics of the Ising Model*. J. Math. Phys. **4** (1963), 294.

- [45] K. Binder. *Finite Size Scaling Analysis of Ising Model Block Distribution Functions*. Z. Phys. B: Condens. Matter **43** (1981), 119.
- [46] R. Swendsen und J.-S. Wang. *Nonuniversal critical dynamics in Monte Carlo simulations*. Phys. Rev. Lett. **58** (Jan. 1987), 86–88.
- [47] U. Wolff. *Collective Monte Carlo Updating for Spin Systems*. Phys. Rev. Lett. **62** (Jan. 1989), 361–364.
- [48] D. Stauffer und A. Aharony. *Introduction To Percolation Theory*. Taylor & Francis, 1994.
- [49] D. Stauffer. *Scaling theory of percolation clusters*. Phys. Rep. **54** (1979), 1–74.
- [50] F. Schwabl. *Statistische Mechanik*. Springer-Lehrbuch. Springer, 2006.
- [51] F. Y. Wu. *The Potts model*. **54** (1982), 235–268.
- [52] F. Y. Wu. *Percolation and the Potts model*. J. Stat. Phys. **18** (1978), 115–123.
- [53] M. E. J. Newman und R. M. Ziff. *Fast Monte Carlo algorithm for site or bond percolation*. Phys. Rev. E **64** (Juni 2001), 016706.
- [54] J. Hoshen und R. Kopelman. *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm*. Phys. Rev. B **14** (1976), 3438–3445.
- [55] R. Brown. *A brief account of microscopical observations made in the months 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 inorganic bodies*. Philosophical Magazine Series 2 **4** (1828), 161–173.
- [56] A. Einstein. *Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen*. Annalen der Physik **322** (1905), 549–560.