

## 1. Übungsblatt

Ausgabe: XX.XX.2022

Prof. H. Jelger Risselada

Abgabe: XX.XX.2022 20 Uhr

---

## Aufgabe 0: Verständnisfragen Blatt 0

0 Punkte

- 1) Was bedeutet numerische *Stabilität* und wieso tritt sie auf?
- 2) Wieso kann eine höhere *Genauigkeit* (beispielsweise durch eine feinere Diskretisierung) zu numerischer Instabilität führen?

## Musterlösung:

- 1) Stabilität beschreibt, wie robust ein Algorithmus gegen kleiner Fehler ist. Diese Fehler können aus verschiedenen Quellen stammen und die Eingangsdaten oder Zwischenergebnisse betreffen. Da physikalische Näherungen als Fehlerquellen sehr schwierig zu untersuchen sind, sind für die Numerik relevant
  - (a) **Rundungsfehler:** Aus der Fließkommazahldarstellung (häufig) oder Integerdarstellung (seltener). Hierdurch entsteht in der Regel ein numerisches Rauschen, das in einem instabilen Algorithmus zu signifikanten Fehlern anwachsen kann. Daher gilt es hier, entsprechende Rechenschritte geschickt so zu formulieren, dass beispielsweise Auslöschung möglichst nicht auftritt.
  - (b) **Abbruchfehler:** In vielen Verfahren wird eine  $N$ -malige Iteration/Rekursion durchgeführt oder ein Kontinuum auf  $\Delta x$  diskretisiert, wobei numerisch zwangsläufig  $N < \infty$  und  $\Delta x > 0$  gilt. Dadurch kann eine mathematische Methode numerisch oft nur näherungsweise implementiert werden.
- 2) Diese Fehler können sich gegenseitig bedingen, wenn beispielsweise zur Verringerung des Abbruchfehlers einer Integration eine sehr feine Diskretisierung gewählt wird. Hierbei werden viele kleine Teilergebnisse auf das Gesamtintegral addiert, wodurch der Rundungsfehler ansteigt. Eine zu feine Diskretisierung kann das Ergebnis dadurch sogar verschlechtern.

Auch in Aufgabe 3 sehen wir eine scheinbare Verbesserung der Genauigkeit des Euler-Verfahrens durch eine symmetrische Ableitung, die für die abfallende Exponentialfunktion jedoch zu einer Instabilität des Verfahrens führt.

## Aufgabe 1: Verständnisfragen Blatt 1

0 Punkte

- 1) Welches der in Aufgabe 2) und 3) genannten Integrationsverfahren hat die höchste Genauigkeit? Welches ist am besten?
- 2) Welchen Grad darf ein Polynom maximal haben, um mit der Simpsonregel exakt integrierbar zu sein?

### Musterlösung:

- 1) Am exaktesten ist natürlich die Simpsonregel. Die Antwort auf die zweite Frage ist jedoch situationsabhängig. Für einzelne, komplexere Integrale ist die Simpsonregel am besten, sollten jedoch sehr viele Integrationen durchgeführt werden müssen, kann es zu Gunsten der Laufzeit allerdings auch ratsam sein auf Integrationsverfahren mit geringerem numerischen Aufwand zurückzugreifen.
- 2) Die Simpsonregel selber ist exakt für Polynome vom Grad  $n = 3$ . Die Genauigkeit einer jeden Newton-Cotes-Formel, zu welchen die Simpsonregel gehört, lässt sich dabei anhand der Anzahl der Stützstellen angeben. Eine ungerade Anzahl an Stützstellen ist dabei immer zu bevorzugen, da die Quadratur mit einer Stützstelle mehr immer auf den gleichen Exaktheitsgrad führt!

- 1) Was versteht man unter symplektischen Integrationsverfahren? Welche Vorteile bieten sie?
- 2) Was versteht man unter Hauptwertintegralen?

**Musterlösung:**

- 1) Symplektischen Integrationsverfahren sind numerische Integrationsverfahren, welche für Systeme genutzt werden, die durch den Hamilton-Formalismus beschrieben werden können. Sie bieten den Vorteil, dass jeder Zeitschritt einer kanonischen Transformation entspricht und somit, wie von dem Liouville-Theorem gefordert, das Phasenraumvolumen stets erhalten bleibt. Ein Beispiel dafür ist der Verlet-Algorithmus, siehe Aufgabe 2.
- 2) Man spricht in der Mathematik von einem Hauptwertintegral, wenn über eine Singularität einer Funktion integriert wird, bspw.

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

mit  $a < z < b$  und sich die divergenten Teile mit verschiedenem Vorzeichen wegheben und somit dem Integral trotz Singularität ein definierter Wert zugewiesen werden kann. Numerisch kann ein Hauptwertintegral berechnet werden, indem die Umgebung um die Singularität isoliert wird und analytisch in eine Form gebracht wird, die keine Singularität mehr besitzt. Somit kann dieser Anteil, wie auch der nicht-singuläre Anteil, wieder mit einem beliebigen numerischen Integrationsverfahren ausgewertet werden, siehe Aufgabe 1.

- 1) Worauf müssen Sie bei der Initialisierung einer MD-Simulation achten? Warum ist dies notwendig?  
*Tipp:* Was würde passieren, wenn zwei Teilchen exakt den gleichen Ort haben?
- 2) Was beschreibt die Paarkorrelationsfunktion? Welche Eigenschaften hat diese und was können Sie anhand dieser über das zugehörige System aussagen?
- 3) In welchem Ensemble wird bei der MD-Simulation gearbeitet? Was sind die natürlichen Größen? Was ändert ein Thermostat an diesen Tatsachen?

**Musterlösung:**

- 1)
  - Zu Beginn sollte zwischen den Teilchen ein ausreichend großer Abstand sein, damit die Kräfte nicht zu groß werden. Sonst könnten die Teilchen bei ungünstig gewähltem Integrationsschritt explosionsartig auseinanderfliegen, was die Äquilibration deutlich erschwert.
  - Es gilt Energieerhaltung, also wird die Gesamtenergie bei der Initialisierung schon festgelegt.
  - Die Geschwindigkeiten müssen so gewählt werden, dass die Schwerpunktschwindigkeit verschwindet, damit das System nicht driftet und die Schwerpunktsbewegung auch nicht unphysikalisch zur Temperatur beiträgt. Interessant sind ohnehin nur die Relativbewegungen der Teilchen.
- 2) Die Paarkorrelationsfunktion beschreibt die Struktur des Systems. Anhand dieser lassen sich Aussagen über Nachbarschaftsverhältnisse zwischen den Teilchen und über die Anordnung der Teilchen in den verschiedenen Phasen treffen. Eigenschaften:
  - $g(r) \approx 1$  für  $r \rightarrow \infty$ , d.h. es gibt keine Korrelationen zwischen unendlich weit entfernten Teilchen.
  - $g(r) = 1$  für ideales Gas, da es dort keine Korrelationen gibt und alle Teilchen unabhängig voneinander sind.
  - Bei "harten", Teilchen mit Teilchendurchmesser  $D$  können sich keine Teilchen im Abstand  $r < D$  aufhalten, deshalb gilt  $g(r) = 0$  für  $r < D$ .
  - Gasphase:  $g(r)$  hat typischerweise nur ein Maximum bei einem  $r_{\max}$ , das den wahrscheinlichsten Teilchenabstand anzeigt. Dieses  $r_{\max}$  sollte ca. im Minimum des anziehenden Potentials  $V(r)$  liegen, aber oberhalb des Teilchendurchmessers liegen.
  - Flüssigkeit:  $g(r)$  hat typischerweise mehrere Maxima bei  $r$ , die Vielfache des Potentialminimums sind und deren Höhen mit zunehmendem  $r$  monoton abnehmen.  $g(r)$  oszilliert dabei um den Wert 1, der für  $r \rightarrow \infty$  angenommen wird. Ursache: Die typische Nahordnung der Teilchen in einer Flüssigkeit, bei der sich Teilchen in "Schalen", anordnen.
  - Kristalline Phase:  $g(r)$  hat unendlich viele Maxima aufgrund der Kristallordnung.

(Für Beispiele zu den verschiedenen Phasen siehe Kierfeldskript Abbildungen 5.6 und 5.7)

- 3) Ohne ein Thermostat wird bei der MD-Simulation im mikrokanonischen Ensemble (mit erhaltener Gesamtenergie  $E$ ) gearbeitet. Führt man ein Thermostat ein, um die Temperatur  $T$  konstant zu halten, wird im kanonischen Ensemble gearbeitet. Natürliche Größen im mikrokanonischen Ensemble sind die Gesamtenergie  $E$ , das Volumen  $V$  und die Teilchenzahl  $N$ . Im kanonischen Ensemble wird die Gesamtenergie  $E$  durch die Temperatur  $T$  als natürliche Größe ersetzt.

#### Aufgabe 4: Verständnisfragen Ergänzung aus Vorjahr

0 Punkte

1. Welche Arten der Schrittweitenanpassung kennen Sie?
2. Ist eine Schrittweitenanpassung immer sinnvoll?

#### Musterlösung:

1. Arten
  - brute force, mit Vorwissen die Schrittweiten anpassen.
  - 2 Schritte mit  $h$ , mit einem Schritt mit  $2h$  vergleichen  $\rightarrow$  Fehlerabschätzung
  - $m$ te Ordnung berechnen,  $m > m'$ te Ordnung „kostenlos“ mitbestimmen  $\rightarrow$  Fehlerabschätzung
2. Nein, wenn eine Funktion nur leicht schwankt bringt der zusätzliche Rechenaufwand kaum eine vergleichbare bessere Genauigkeit.

- 1) Welche Verfahren zum Lösen einer zeitunabhängigen partiellen Differentialgleichung kennen Sie? Welche Vorteile bieten diese jeweils?
- 2) Wie unterscheidet sich das FTCS-Schema von dem Crank-Nicolson-Schema? Was müsste man bei einer Implementierung des Crank-Nicolson-Schemas zusätzlich beachten?

**Musterlösung:**

- 1) Um eine zeitunabhängige partielle Differentialgleichung numerisch zu lösen, gibt es unterschiedliche Verfahren. Eine Möglichkeit ist ein iteratives Fixpunkt Verfahren zu nutzen wie z.B. die Jacobi-Iteration oder die Gauß-Seidel-Iteration. Durch den Banachschen Fixpunktsatz kann gezeigt werden, dass die beiden Verfahren stets konvergieren. Der entscheidende Unterschied liegt in der Rekursionsformel. Im Gegensatz zur Jacobi Iteration nutzt die Gauß-Seidel-Iteration teilweise die Werte der Nachbarn für den  $(n + 1)$ -ten Schritt, die bereits berechnet worden sind, um den eigenen  $(n + 1)$ -ten Schritt zu berechnen, was eine schnelleren Konvergenz zur Folge hat. Eine weitere Möglichkeit bietet das Relaxationsverfahren, welches die Lösung einer Jacobi/Gauß-Seidel-Iteration für einen Schritt mit einem Parameter  $\sigma \in [0, 2]$  modifiziert, was zu einer besseren Konvergenz führen kann.
- 2) Der Hauptunterschied ist, dass das Crank-Nicolson-Schema im Gegensatz zum FTCS-Schema implizit ist. Das heißt, dass zur Berechnung eines neuen Zeitschritts teilweise bereits Werte zu diesem neuen Zeitschritt benötigt werden. Als Konsequenz muss in jedem Zeitschritt ein tridiagonales lineares Gleichungssystem gelöst werden, was einen höheren technischen und numerischen Aufwand mit sich bringt. Ein klarer Vorteil des Crank-Nicolson-Schemas ist allerdings, dass es für beliebige Schrittweiten  $\Delta t$  stabil bleibt.

- 1) Welche Bedeutung hat die Feigenbaumkonstante und in welchen Beispielen tritt sie auf?
- 2) Welche Iterationsverfahren kennen Sie zur Lösung nichtlinearer Gleichungen?

**Musterlösung:**

- 1) Die **Feigenbaum-Konstante** ist eine wichtige universelle Konstante in der Chaos-Forschung bei der Betrachtung **Nicht-Linearer Systeme**. Nicht-Lineare Systeme sind dadurch gekennzeichnet, dass Sie in geeigneten Parameter-Bereichen **Fixpunkte** und oder sogenannte **Orbits** besitzen und sich in anderen Bereichen chaotisch verhalten. Am Übergang zwischen diesen Bereichen tritt das Phänomen der sogenannten **Bifurkation** oder **Periodenverdopplung** auf, d.h., bei Veränderung eines geeigneten Parameters  $r$  verdoppeln sich die Orbitgrößen immer weiter. Die Feigenbaum-Konstante trifft sozusagen eine Aussage über die Geschwindigkeit dieser Verdopplung. Diese ist nämlich unabhängig vom zugrundeliegenden Problem im Limes hoher Orbitgrößen  $n$  durch den konstanten Wert

$$\delta = \lim_{n \rightarrow \infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} = 4.669... \quad (1)$$

gegeben. Es sei erwähnt, dass dieser Wert ziemlich hoch ist, d.h., sobald die ersten Bifurkationen auftreten wird der chaotische Bereich sehr rasch erreicht. Hat man ein hinreichend großes  $n$  erreicht, so kann man mit der Feigenbaum-Konstante und dem Abstand  $\Delta r$  der aktuellen Bifurkation zur vorherigen Bifurkation den verbleibenden Abstand zum chaotischen Bereich abschätzen durch die geometrische Reihe

$$\sum_{k=1}^{\infty} \frac{\Delta r}{\delta^k} = \frac{\Delta r}{\delta - 1} \approx \frac{\Delta r}{3.669}. \quad (2)$$

- 2) Gesucht sind Iterationsverfahren, die in der Lage sind eine Lösung  $x_N$  zu einer nichtlinearen Gleichung (hier eindimensional)

$$g(x) = h(x) \quad \text{bzw.} \quad f(x) := g(x) - h(x) = 0 \quad (3)$$

zu finden. Dabei ist der Ausdruck “in der Lage” wörtlich zu nehmen, denn ob ein Wert gefunden wird und welcher Wert gefunden wird hängt von der Wahl der Startwerte ab.

Eines der einfachsten Verfahren ist das **Intervallhalbierungsverfahren**, bei dem durch Vergleichen von Funktionswerten wie der Name schon sagt Intervalle immer weiter halbiert werden und sich so einem Fixpunkt angenähert wird.

Ein etwas schnelleres Verfahren ist das sogenannte **Regula Falsi Verfahren**. Dieses ist ähnlich zum Intervallhalbierungsverfahren, jedoch werden die Intervalle nicht naiv in ihrer Mitte unterteilt sondern an einer durch lineare Interpolation der Randpunkte geschickter gewählten Stelle.

Beim **Newton-Raphson-Verfahren** wird im Gegensatz zu den Verfahren zuvor zusätzlich die Ableitung der Funktion genutzt und so die nächste Position mit Hilfe der Tangente der Funktion bestimmt. Problematisch bei dieser Methode sind Startwerte, die in der Nähe von Sattel oder Extrempunkten liegen, da dort  $f'(x) \approx (0)$  ist. Vorteil dieser Methode ist unter anderem, dass sie im Gegensatz zu den vorher genannten Methoden durch Einführen der Jacobi-Matrix auf mehrere Dimensionen verallgemeinert werden kann.

- 1) Welche Diagonalisierungsalgorithmen kennen Sie und von welcher Ordnung ist ihre Laufzeit?
- 2) Diagonalisierung wird in der Physik häufig verwendet um die Eigenenergien eines Hamiltonoperators  $H$  zu bestimmen. Dazu muss  $H$  in einer geeigneten Basis formuliert werden. Eine Basis die immer verwendet werden kann, ist die vollständige Zustandsbasis von  $H$ . Warum ist die direkte Diagonalisierung größerer Systeme in der vollständigen Zustandsbasis oftmals nicht sinnvoll?

**Musterlösung:**

- 1) LU-Zerlegung  $\mathcal{O}(N^3)$ , Jacobi-Rotationen  $\mathcal{O}(N^3)$ , Potenzmethode  $\mathcal{O}(mN^2)$  ( $m$  ist die benötigte Anzahl an Iterationen bis zur Konvergenz), Lanczos-Verfahren  $\mathcal{O}(N^3)$ , Householder Algorithmus + QR Iteration  $\mathcal{O}(N^3)$

Anders als vielleicht zunächst zu erwarten, haben all diese Algorithmen eine  $\mathcal{O}(N^3)$ -Abhängigkeit (wenn wir mal ganz dreist  $m \approx N$  annehmen;  $m \gg N$  ist jedoch die bessere Abschätzung). Der Unterschied besteht lediglich in den Vorfaktoren. Gerade für kleine Matrizen können bei entsprechenden Vorfaktoren jedoch auch die  $\mathcal{O}(N^2)$ -Term, welche vor allem bei den komplexeren Algorithmen (wie beispielsweise Householder) etwas größer sind, eine entscheidende Rolle spielen. Als Faustregel gilt, kleine Matrizen  $N \lesssim 10$  sind einfache Verfahren besser. Erst für größere Matrizen können die mächtigeren Algorithmen wie Householder oder Lanczos ihr volles Potential entfalten. (Auch die Parallelisierbarkeit der Algorithmen kann durchaus eine Rolle spielen. In diesem Rahmen macht es jedoch wenig Sinn das weiter zu betrachten.)

- 2) Betrachten wir ein einfaches Modell, so wie beispielsweise das Isingmodell der Größe  $N$ , so kann jeder Spin 2 Zustände annehmen. Daraus folgt, dass die Zustandsbasis eine Größe von  $2^N$  Zuständen hat. Die zu diagonalisierende Matrix wird also sehr schnell sehr groß, was sowohl Speicher- als vor allem auch Laufzeittechnisch zu Problemen führen kann.



## Aufgabe 8: Verständnisfragen Ergänzung aus Vorjahr

0 Punkte

- 1) Welchen Vorteil bieten Pseudo-Zufallszahlen? Wann ist es sinnvoll, „echte“ Zufallszahlen zu verwenden?
- 2) Welche Anforderungen werden an gute Zufallszahlen gestellt? Wie können diese getestet werden?
- 3) Welche Verfahren kennen Sie, um Zufallszahlen zu generieren? Welche Verteilungen können Sie damit erzeugen?

### Musterlösung:

- 1) Das gute an Pseudozufallszahlen ist die Wiederholbarkeit von numerischen Experimenten. Echte Zufallszahlen sind sinnvoll falls diese Wiederholbarkeit nicht benötigt wird.
- 2) Guter Zufall bedeutet zu Einen eine geringe Korrelation aufeinander folgender Zufallszahlen (siehe auch Aufgabe 1 d) und e)). Zum anderen sollte die Folge auch möglichst lang laufen bevor sie sich wiederholt. Getestet werden kann dies z.B. durch plotten Paaverteilung aufeinanderfolgender Zufallszahlen.
- 3) Zum generieren: Linear Kongruenter Generator, Mersenne Twister und XOR shift um Gleichverteilungen zu erzeugen. Zu umwandeln: Rejection Sampling → alles Mögliche, Transformationsmethode → alles Mögliche, central limit Theorem → Normalverteilung, Box Muller → Normalverteilung.

- 1) Welche Verfahren zur Minimierung eindimensionaler Funktionen kennen Sie? Diskutieren Sie deren Vor- und Nachteile.
- 2) Nennen Sie Ihnen bekannte Verfahren zur Minimierung von Funktionen der Form  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  mit  $N > 1$ . Beschreiben Sie das generelle Vorgehen bei derartigen Minimierungsaufgaben und diskutieren Sie Vor- bzw. Nachteile der jeweiligen Verfahren.

**Musterlösung:**

- 1) Verfahren aus der Vorlesung:

- Intervallhalbierung mit goldenem Schnitt. Vorteil: Es wird immer ein lokales Minimum gefunden, wenn es sich in den Klammern befindet. Nachteil: Langsamer als andere Verfahren.
- Newton-Verfahren. Vorteil: Konvergiert sehr schnell, wenn der Startwert nah genug am Minimum gewählt wird. Nachteil: Es ist nicht garantiert, dass das Verfahren konvergiert.

Man kann die Vorteile beider Verfahren auch kombinieren. Man startet mit einer Klammerung und berechnet in jedem Schritt den nächsten Wert sowohl nach Intervallhalbierung als auch nach Newton-Verfahren. Falls das Ergebnis des Newtonverfahrens in der Klammer liegt, wird dieses genommen, ansonsten nimmt man das Ergebnis der Intervallhalbierung. Eine besonders geschickte Variante dieser Idee wird im Brent-Verfahren umgesetzt. Alle Verfahren haben den gemeinsamen Nachteil, dass nur ein lokales Minimum gefunden wird.

- 2) Wir machen uns erst einmal klar, dass die in der Aufgabe beschriebene Klasse von Funktionen  $f$  aus denjenigen Funktionen besteht, die von mehr als einer Variablen abhängen, ihrerseits aber skalare Ergebnisse liefern.

Der Grundgedanke derartiger Funktionen ist, einen Startpunkt  $\vec{x} \in \mathbb{R}^N$  zu wählen, um von diesem Startpunkt aus sukzessive Minimierung nacheinander in verschiedene Richtungen durchzuführen. Liegt die Richtung einmal fest, so handelt es sich um eine eindimensionale Minimierung, die schon aus den vorherigen Blättern bekannt ist. Wie die Richtungen vom jeweiligen Startpunkt aus zu wählen sind wie auch die Frage, wann die Iteration abubrechen ist, ist Gegenstand des speziell ausgewählten Verfahrens. Es gibt bspw.:

- (a) Powell-Verfahren (Vorteil: Keine Gradientenbestimmung und damit für Funktionen einsetzbar, die nicht analytisch ableitbar sind),
- (b) Gradienten Methode (explizite Nutzung von Ableitungen; Vorteil: Konvergiert stets zu einem Minimum; Nachteil: Konvergenz ist langsam),
- (c) Methode konjugierter Gradienten (systematische Erweiterung, bei der alle Richtungen zueinander konjugiert sind und eine deutlich schnellere Konvergenz vorliegt),...
- (d) Genetische Algorithmen und Schwarmoptimierung (Vorteil: Kann gut aus lokalen Minima entkommen, gut für komplizierte und nicht-differenzierbare Funktionen, Nachteil: Nutzt nicht direkt Gradientenrichtungen und kann dadurch in ungünstige Richtungen laufen, redundante Rechenarbeit wenn viele Teilchen nahe beieinander)

- 1) Worin liegt der Unterschied zwischen der Molekuldynamik(MD)-Simulation und der Monte-Carlo(MC)-Simulation für molekulare Systeme? Wann eignet sich welche Methode?
- 2) Was sind die grundlegenden Eigenschaften eines Markov-Prozesses?

**Musterlösung:**

- 1) Bei der sogenannten **Molekuldynamik-Simulation** versucht man die Dynamik eines makroskopischen Systems durch explizite Betrachtung der involvierten Teilchen, d.h. durch explizite Berechnung von Kräften und Trajektorien, zu simulieren. Makroskopische Größen werden dabei durch Mittelungen über den betrachteten Phasenraum, d.h., im mikrokanonischen Ensemble berechnet.

Die Idee der sogenannten **Monte-Carlo-Simulation** ist es, hochdimensionale Probleme durch "Sampling" also durch das Ziehen zufälliger Stichproben bzw. Stützstellen zu bewältigen. Damit ist sie allgemein ein sehr praktisches Werkzeug in der Vielteilchenphysik in der man es mit hochdimensionalen Phasenräumen und dementsprechend hochdimensionalen Integralen zur Berechnung von Erwartungswerten zu tun hat. Dort arbeitet man dann im kanonischen Ensemble und gewichtet die thermodynamischen Zustände, also letztlich die Monte-Carlo-Samples, mit dem Boltzmann-Faktor, siehe. z.B. Metropolis-Algorithmus.

Tendenziell werden für Modelle mit mehr Freiheitsgraden (z.B. gasförmig) eher Monte-Carlo-Simulationen und für Systeme mit weniger Freiheitsgraden (z.B. fest) eher MD-Simulationen genutzt, denn: desto hochdimensionaler ein System ist, desto weniger Teilchen können durch MD exakt numerisch simuliert werden.

- 2) Ein Markov-Prozess ist im Wesentlichen dadurch gekennzeichnet, dass er kein "inneres Gedächtnis" hat. Das heißt, wenn ein System über mehrere stochastische Zwischenprozesse (kontinuierlich oder diskret) von einem Anfangszustand  $(i, t)$  in einen Endzustand  $(j, t + \Delta t)$  übergeht, dann darf die Übergangswahrscheinlichkeit/-rate für den Gesamtprozess nicht von den Zwischenprozessen abhängen.