Universität Stuttgart – Institut für Computerphysik

Probeklausur

Physik auf dem Computer SS 2012

JP Dr. Axel Arnold Dr. Olaf Lenz Florian Fahrenberger
Dominic Röhm

6. August 2012

Name	
Vorname	
Matrikelnummer	

Hinweise

- In der Regel gibt der verfügbare freie Platz einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Lies Dir *alle* Fragen am Anfang durch, bevor Du anfängst, sie zu beantworten.
- Falls der Platz nicht ausreichen sollte, verwende zusätzliche Blätter. Beschrifte diese unbedingt mit Deinem Namen und Matrikelnummer!
- Die Maximalpunktzahl ist 100.
- Zum Bestehen der Klausur sind 50 Punkte notwendig.

Viel Erfolg!

1 Lineare Algebra I (13 Punkte)

Aufgabe 1: (1 Punkt)

Schreibe eine beliebige (5×5)-Drei-Bandmatrix auf.

Antwort:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Aufgabe 2: (5 Punkte)

Löse das folgende lineare Gleichungssystem mit Hilfe der Gaußelimination mit Totalpivotisierung und der Rücksubstitution.

$$\begin{pmatrix} -24 & -6 & -14 \\ 32 & 8 & 16 \\ 8 & 4 & 8 \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} -130 \\ 168 \\ 52 \end{pmatrix}$$

Antwort:

$$\left(\begin{array}{ccc|c} -24 & -6 & -14 & -130 \\ \mathbf{32} & 8 & 16 & 168 \\ 8 & 4 & 8 & 52 \end{array}\right) \quad \text{(Tausche Zeile 1 und 2)}$$

$$\Rightarrow \left(\begin{array}{ccc|c} 32 & 8 & 16 & 168 \\ -24 & -6 & -14 & -130 \\ 8 & 4 & 8 & 52 \end{array}\right) \quad \text{(Eliminiere Zeilen 2 und 3))}$$

$$\Rightarrow \left(\begin{array}{ccc|c} 32 & 8 & 16 & 168 \\ 0 & 0 & -2 & -4 \\ 0 & 2 & \mathbf{4} & 10 \end{array}\right) \quad \text{(Tausche Zeile und Spalte 2 und 3)}$$

$$\Rightarrow \left(\begin{array}{ccc|c} 32 & 16 & 8 & 168 \\ 0 & 4 & 2 & 10 \\ 0 & -2 & 0 & -4 \end{array}\right) \quad \text{(Eliminiere Zeile 3)}$$

$$\Rightarrow \left(\begin{array}{ccc|c} 32 & 16 & 8 & 168 \\ 0 & 4 & 2 & 10 \\ 0 & 0 & 1 & 1 \end{array}\right) \quad \text{(Rücksubstitution und Rücktauschen)}$$

$$\Rightarrow \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} \Rightarrow \mathbf{x} = \begin{pmatrix} 4 \\ 1 \\ 2 \end{pmatrix}$$

Aufgabe 3: (7 Punkte)

Invertiere die folgende Matrix mit Hilfe der Gaußeliminiation.

$$\begin{pmatrix} -1 & 3 & -3 \\ 0 & -6 & 5 \\ -5 & -3 & 1 \end{pmatrix}$$

Antwort:

$$\left(\begin{array}{ccc|ccc} -1 & 3 & -3 & 1 & 0 & 0 \\ 0 & -6 & 5 & 0 & 1 & 0 \\ -\mathbf{5} & -3 & 1 & 0 & 0 & 1 \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} -\mathbf{5} & -3 & 1 & 0 & 0 & 1 \\ 0 & -6 & 5 & 0 & 1 & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{3}{10} & 0 & \frac{1}{10} & -\frac{1}{5} \\ 0 & -\mathbf{6} & 5 & 0 & 1 & 0 \\ -1 & 3 & -3 & 1 & 0 & 0 \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{3}{10} & 0 & \frac{1}{10} & -\frac{1}{5} \\ 0 & -6 & 5 & 0 & 1 & 0 \\ 0 & 3 & -\frac{27}{10} & 1 & \frac{1}{10} & -\frac{1}{5} \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{3}{10} & 0 & \frac{1}{10} & -\frac{1}{5} \\ 0 & -6 & 5 & 0 & 1 & 0 \\ 0 & 0 & -\frac{2}{5} & 2 & \frac{6}{5} & -\frac{2}{5} \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & \frac{3}{10} & 0 & \frac{1}{10} & -\frac{1}{5} \\ 0 & 1 & -\frac{5}{6} & 0 & -\frac{1}{6} & 0 \\ 0 & 0 & -\frac{2}{5} & 2 & \frac{6}{5} & -\frac{2}{5} \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{3}{2} & 1 & -\frac{1}{2} \\ 0 & 1 & 0 & -\frac{25}{6} & -\frac{8}{3} & \frac{5}{6} \\ 0 & 0 & 1 & -5 & -3 & 1 \end{array}\right)$$

2 Darstellung von Funktionen (16 Punkte)

Aufgabe 4: (0 Punkte)
Wie hieß Fourier mit Vornamen?

Antwort:

Joseph

Aufgabe 5: (2 Punkte)
Berechne die Taylorentwicklung der Funktion $f(x) = \sin x$ um den Punkt $x_0 = 0$ bis zur dritten Ordnung (also bis zum $f'''(x)$-Term).

Antwort:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3$$

$$f(x) = x - \frac{1}{6}x^3$$

Aufgabe 6: (1 Punkt)
Wird eine Funktion durch eine immer höhere Anzahl von äquidistanten Stützstellen bei der Polynominterpolation immer besser angenähert? Warum?

Antwort:

Nein. Am Beispiel der Rungefunktion wurde aufgezeigt, dass im Fall der äqudistanten Stützstellen, eine höhere Anzahl von Stützstelle sogar zu einer Verschlechterung führt. Verwendet man Chebyshev-Stützstellen, dann ja.

Aufgabe 7: (1 Punkt)

Wieso ist die Funktion $f(x) = e^{3x}$ schlecht durch eine Taylorentwicklung anzunähern?

Antwort:

Je häufiger die Funktion abgeleitet wird, desto größer wird sie. Die höheren Terme der Taylorentwicklung schrumpfen also nur sehr langsam, es ist also notwendig, viele Terme mitzunehmen, um eine gute Annäherung zu erhalten.

Aufgabe 8: (1 Punkt)

Welche Eigenschaften müssen Funktionen haben, damit sie gut für eine numerische Näherung durch eine Fouriertransformation geeignet sind?

Antwort:

Periodisch, keine Singularitäten, stetig.

Aufgabe 9: (5 Punkte)

Berechne mit Hilfe der Lagrangepolynome das interpolierende Polynom in der Form ax^2+bx+c der Funktion

$$f(x) = \sin x$$

für die Chebyshev-Stützstellen in der nebenstehenden Tabelle.

k	$\mathbf{x_k}$
0	0
1	$\frac{\sqrt{3}}{2}$
2	$-\frac{\sqrt{3}}{2}$

Antwort:

k	$\mathbf{x_k}$	$\mathbf{y_k}$
0	0	0
1	$\frac{\sqrt{3}}{2}$	0,762
2	$-\frac{\sqrt{3}}{2}$	-0,762

$$\begin{aligned} P(x) &= \sum_{k=0}^{2} y_k L_k(x) \\ &= y_0 \left(\frac{4}{3}x^2 - 1\right) + y_1 \left(\frac{2}{3}x^2 + \frac{\sqrt{3}}{3}x\right) + y_2 \left(\frac{2}{3}x^2 - \frac{\sqrt{3}}{3}x\right) \end{aligned}$$

$$P(x) = 0{,}762 \left(\frac{2}{3}\sqrt{3}x\right) = 0{,}880x$$

Aufgabe 10: (1 Punkt)

Nenne einen Nachteil der Splineinterpolation gegenüber der Interpolation mit Polynomen.

Antwort:

Splines sind stückweise definierte Polynome, also keine analytisch geschlossene Funktion. Daher ist z.B. die Berechnung der Nullstellen oder Ableitungen aufwendiger.

Aufgabe 11: (5 Punkte)

Berechne die Koeffizienten des Hornerschemas für ein Polynom 3. Grades mit den Nullstellen 1, 2, 3. Dabei sei der führende Koeffizient $c_3 = 1$.

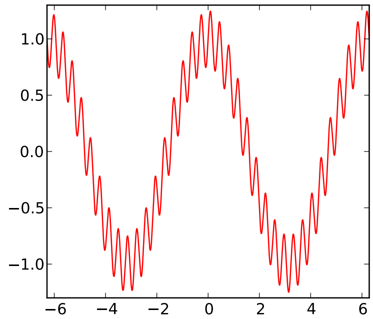
Antwort:

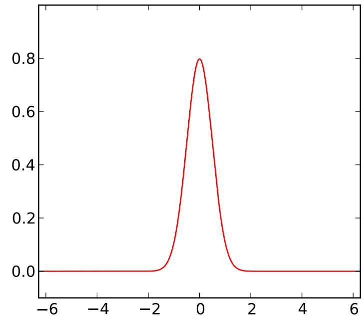
$$
\begin{aligned}
&P(x) \\
&\quad = (x-1)(x-2)(x-3) \\
&\quad = x^3 - 6x^2 + 11x - 6 \\
&\quad \Rightarrow c_0 = -6; c_1 = 11; c_2 = -6; c_3 = 1
\end{aligned}
$$

3 Signalverarbeitung (12 Punkte)

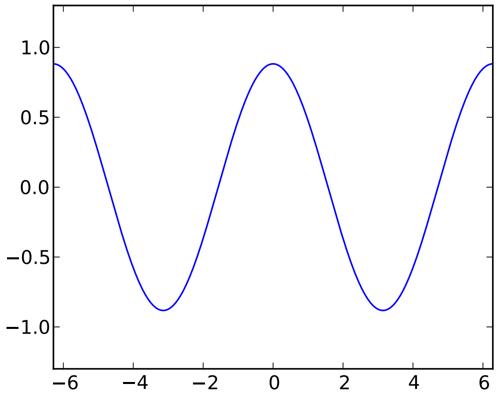
Aufgabe 12: (3 Punkte)

Skizziere die Faltung der in nebenstehender Abbildung skizzierten Funktionen.





Antwort:



Aufgabe 13: (3 Punkte)

Welche Frequenz ist die höchste sinnvoll zu verwendende bei der diskreten Fouriertransformation eines Signals, das mit n Punkten auf dem Intervall $[0, L[$ abgetastet wird?

Antwort:

Die Nyquist-Frequenz $f_{\text{Nyquist}} = \frac{1}{2\Delta} = \frac{n}{2L}$. Dabei ist $\Delta = \frac{n}{L} = \frac{2\pi}{\omega_0}$ die Abtastrate und $\omega_0 = \frac{2\pi n}{L}$ die Abtastkreisfrequenz.
Alle Frequenzen oberhalb dieser Frequenz werden auf Frequenzen unterhalb der Nyquistfrequenz abgebildet und verfälschen das Signal. Sind solchen Frequenzen im Signal vorhanden, müssen diese gefiltert oder das Signal mit einer höheren Rate abgetastet werden.

Aufgabe 14: (1 Punkt)

Von welcher Ordnung in N ist die Geschwindigkeit der diskreten (nicht der schnellen) Fouriertransformation (wobei N die Anzahl der zu transformierenden Punkte ist)?

Antwort:

Von der Ordnung $\mathcal{O}(N^2)$.

Aufgabe 15: (5 Punkte)

Schreibe eine Pythonfunktion `stats(x)`, die den Mittelwert und die Standardabweichung einer Datenreihe berechnet und zurückgibt, ohne dabei die entsprechenden NumPy-Funktionen zu benutzen. Dabei sei x ein eindimensionales Numpy-Array.

Antwort:

```
def stats(x):
  N = len(x)

  mean = x.sum()/N
  mean2 = (x**2).sum()/N

  stddev = (mean2 - mean**2)**0.5

  return mean, stddev
```

4 Nichtlineare Gleichungssysteme (11 Punkte)

Aufgabe 16: (1 Punkt)
Wann verwendet man das gedämpfte Newtonverfahren anstelle des ungedämpften?

Antwort:

Wenn die Umgebung um die Nullstelle, in der das Newtonverfahren konvergiert, sehr klein ist. Das ist häufig in höherdimensionalen Systemen der Fall. Man verhindert so ein versehentliches „Überspringen“ der Nullstelle.

Aufgabe 17: (5 Punkte)
Schreibe ein Pythonskript, das mit Hilfe der (gegebenen) Funktion `newton(f, fp, x0)` zur Nullstellensuche $\arcsin \frac{1}{2}$ berechnet, ohne dabei die entsprechende Pythonfunktion `asin` zu verwenden. Verwende als Startwert 0. `f` sei dabei eine mathematische Funktion, `fp` deren Ableitung, und `x0` der Startwert des Newtonverfahrens. Die Funktion `newton` muss hier *nicht* definiert werden, sondern wird als gegeben vorausgesetzt.

Antwort:

```python
import math # in Klausur nicht notwendig...

def f(x):
  return (math.sin(x) - 0.5)

def fp(x):
  return math.cos(x)

print newton(f, fp, 0.0)
```

Aufgabe 18: (5 Punkte)

Berechne die ersten drei Näherungen des Newtonverfahrens für $f(x) = x^2 - 2$ mit dem Startwert $x_0 = 1$.

Antwort:

Newton-Verfahren: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
Funktion: $f(x) = x^2 - 2$
Ableitung: $f'(x) = 2x$

$$
\begin{aligned}
x_0 &= 1 \\
x_1 &= 1 - \frac{1^2 - 2}{2} = \frac{3}{2} \\
x_2 &= \frac{3}{2} - \frac{(3/2)^2 - 2}{3} = \frac{17}{12} \\
x_3 &= \frac{17}{12} - \frac{(17/12)^2 - 2}{17/6} = \frac{17}{12} - \frac{1}{408} = \frac{577}{408} = 1.41422
\end{aligned}
$$

5 Numerisches Differenzieren und Integrieren (13 Punkte)

Aufgabe 19: (2 Punkte)

In welchen beiden Fällen benutzt man die Monte-Carlo-Integration anstelle von Integrationsmethoden wie den zusammengesetzten Newton-Cotes-Formeln oder der Gaußquadratur?

Antwort:

Wenn entweder die zu integrierende Funktion nicht genügend glatt oder gar unstetig ist, oder wenn das Integral hochdimensional ist und andere Methoden zu aufwendig werden (ab $d = 5$).

Aufgabe 20: (4 Punkte)

Diskretisiere die Differentialgleichung $\frac{\mathrm{d}f(x)}{\mathrm{d}x} + \alpha f(x) + \beta x = g(x)$ mit Hilfe des folgenden Differenzenquotienten zweiter Ordnung

$$\frac{\mathrm{d}f(x_i)}{\mathrm{d}x} \approx -\frac{1}{12h} f(x_{i+2}) + \frac{2}{3h} f(x_{i+1}) - \frac{2}{3h} f(x_{i-1}) + \frac{1}{12h} f(x_{i-2}).$$

Beschreibe, was getan werden muß, um damit eine Nährung für die Lösung der Differentialgleichung auf dem Interval $[0, L]$ zu erhalten.

Antwort:

1. Erzeuge N äquidistante Stützstellen x_i im Intervall $[0,L]$.
2. Löse das folgende Gleichungssystem (z.B. mittels Gausselimination), um die Näherung für die Lösung der DGL f_i an den Stützstellen x_i zu erhalten:

$$\begin{pmatrix} \alpha & \frac{2}{3h} & -\frac{1}{12h} & 0 & 0 & \cdots & 0 \\ -\frac{2}{3h} & \alpha & \frac{2}{3h} & -\frac{1}{12h} & 0 & \ddots & 0 \\ \frac{1}{12h} & -\frac{2}{3h} & \alpha & \frac{2}{3h} & -\frac{1}{12h} & \ddots & 0 \\ 0 & \frac{1}{12h} & -\frac{2}{3h} & \alpha & \frac{2}{3h} & \ddots & 0 \\ 0 & 0 & \frac{1}{12h} & -\frac{2}{3h} & \alpha & \ddots & -\frac{1}{12h} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \frac{2}{3h} \\ 0 & 0 & 0 & 0 & \frac{1}{12h} & -\frac{2}{3h} & \alpha \end{pmatrix} \cdot \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{pmatrix} = \begin{pmatrix} g(x_1) + \beta x_1 \\ g(x_2) + \beta x_2 \\ \vdots \\ g(x_{n-1}) + \beta x_{n-1} \\ g(x_n) + \beta x_n \end{pmatrix}$$

Aufgabe 21: (5 Punkte)
Schreibe eine Pythonfunktion `compute_pi(N)`, die mit Hilfe von N Schritten der Monte-Carlo-Integration die Kreiszahl π berechnet. Dabei sei `numpy.random.uniform()` eine Funktion, die eine Fließkommazufallszahl zwischen 0 und 1 zurückgibt.

Antwort:

```
def compute_pi(N=10000):
    pi = 0.0
    for i in range(N):
        x = numpy.random.uniform()
        pi += (1-x**2)**0.5

    pi *= 4.0 / N
    return pi
```

Aufgabe 22: (2 Punkte)
Für welche Funktionen liefert die zusammengesetzte Simpsonregel das selbe Ergebnis wie die zusammengesetzte Mittelpunktsregel (bei beliebiger Anzahl von Stützstellen)?

Antwort:

Bei linearen Funktionen, also Polynomen ersten Grades $f(x) = ax+b$.

6 Zufallszahlen (10 Punkte)

Aufgabe 23: (1 Punkt)

Beschreibe die Fouriertransformierte einer guten Zufallsfolge.

Antwort:

Die nullte Mode der Fourierttransformierten sollte eine Amplitude von $\approx \frac{N}{2}$ haben, die anderen Moden sollten kleine Amplituden haben.

Aufgabe 24: (3 Punkte)

Was tut die folgende Pythonfunktion?

```
def f():
    while True:
        x = numpy.random.uniform(-1, 1)
        y = numpy.random.uniform(-1, 1)
        if x**2 + y**2 <= 1:
            return (x, y)
```

Antwort:

Die Funktion erzeugt mittels der *Verwerfungsmethode* einen einzelnen zufälligen Punkt in einem Einheitskreis entsprechend einer Gleichverteilung.

Aufgabe 25: (4 Punkte)

Skizziere (graphisch oder verbal), wie man mit Hilfe von gleichverteilten Zufallszahlen normalverteilte Zufallszahlen erzeugen kann.

Antwort:

Dazu dient die Box-Muller Methode, die eine Erweiterung der Inversionsmethode darstellt. In dieser Methode werden zwei gleichverteilte Zufallszahlen u und u' aus [0,1] gezogen. u' wird als Winkel eines Punktes in Polarkoordinaten interpretiert, der Abstand wird durch die Inversionsmethode gemäß $F(u) = \sqrt{-2\log(u)}$ berechnet. Die karthesischen Koordinaten des Punktes sind dann gemäß einer Normalverteilung verteilt.

Aufgabe 26: (2 Punkte)

Was tut der *seed* eines Zufallszahlengenerators, und welchen Zweck erfüllt er?

Antwort:

Wird der *seed* eines Pseudozufallszahlengenerators mehrfach verwendet, dann wird jedesmal exakt dieselbe Zufallsfolge wieder erzeugt. Dies dient der Reprodzierbarkeit von zufallsbehafteten Ergebnissen.

7 Lineare Algebra II (5 Punkte)

Aufgabe 27: (1 Punkt)

Wofür benötigt man bei der Berechnung von Eigenvektoren die Inverse Iteration? Was muss man zuvor schon berechnet haben?

Antwort:

Wenn man mittels eines Verfahrens die Eigenwerte einer Matrix berechnet hat, dann kann man mit Hilfe der Inversen Iteration die Eigenvektoren berechnen.

Aufgabe 28: (1 Punkt)

Erstelle eine L+D+U-Zerlegung der Matrix $\begin{pmatrix} 2 & 3 & 1 & 7 \\ 1 & 2 & 4 & 1 \\ 4 & 1 & 3 & 0 \\ 3 & 5 & 1 & 1 \end{pmatrix}$.

Antwort:

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 3 & 5 & 1 & 0 \end{pmatrix} D = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} U = \begin{pmatrix} 0 & 3 & 1 & 7 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Aufgabe 29: (3 Punkte)

Was ist die QR-Zerlegung einer Matrix und wozu ist sie nützlich?

Antwort:

Bei der QR-Zerlegung wird eine Matrix A in eine orthonormale Matrix Q und eine rechte obere Dreiecksmatrix R zerlegt, so dass $A = QR$. Da die Matrix Q orthonormal ist, ist es möglich, die Hermitesche Q^H zu berechnen. Dann kann die Gleichung $Ax = b$ wie folgt umgewandelt werden:

$$Ax = QRx = b \Rightarrow Rx = Q^H b$$

Dadurch ist es leicht möglich, die Gleichung $Ax = b$ durch Rücksubtitution zu lösen.

8 Optimierung (7 Punkte)

Aufgabe 30: (4 Punkte)
Gesucht ist das lokale Minimum der Funktion $f(x, y) = \sin((x+1)(y-1))$. Führe vom Ausgangspunkt $x = y = 0$ zwei Schritte des Verfahrens des steilsten Abstiegs mit Schrittweite $\lambda = 0{,}1$ aus.

Antwort:

$$\begin{aligned}
\nabla f(\vec{x}) &= \begin{pmatrix} (y-1)\cos((x+1)(y-1)) \\ (x+1)\cos((x+1)(y-1)) \end{pmatrix} \\
\vec{x}_{i+1} &= \vec{x}_i - \lambda \nabla f(\vec{x}_i) \\
\vec{x}_0 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\vec{x}_1 &= \vec{x}_0 - \lambda \begin{pmatrix} -0.5403 \\ 0.5403 \end{pmatrix} = \begin{pmatrix} 0.05403 \\ -0.05403 \end{pmatrix} \\
\vec{x}_2 &= \begin{pmatrix} 0.05403 \\ -0.05403 \end{pmatrix} - \lambda \begin{pmatrix} -0.5918 \\ 0.5918 \end{pmatrix} = \begin{pmatrix} 0.1008 \\ -0.1008 \end{pmatrix}
\end{aligned}$$

Aufgabe 31: (3 Punkte)
Skizziere (verbal oder graphisch) die Idee eines genetischen Algorithmus.

Antwort:

Ein genetischer Algorithmus ist ein Algorithmus zur globalen Optimierung. Bei einem genetischen Algorithmus wird jede mögliche Lösung des Optimierungsproblems als *Genom* interpretiert, für das man eine eindeutige Vorschrift zur Berechnung der *Fitness* (Qualitätsmaß) vorgibt. Nun beginnt man mit einer *Population* von *N Individuen* (Initiallösungen), deren jeweilige Fitness man berechnet. Die nächste *Generation* von Genomen wird erzeugt, indem die Genome der letzten Generation gemäß der berechneten Fitness übernommen werden. Dann werden die Genome durch *Mutation* (also duch zufällige lokale Änderungen der Lösung) und eventuell durch *Kreuzung* (Rekombination mehrerer Genome) verändert, und wieder ihre Fitness berechnet. Im Verlaufe vieler Generationen sollten die Individuen immer besser werden und sich dem globalen Optimum annähern.

9 Programmieren (7 Punkte)

Aufgabe 32: (5 Punkte)

Schreibe ein Pythonskript, das 1000 Schritte eines eindimensionalen Randomwalks der Schrittweite ± 1 durchführt und danach die Entfernung zum Ursprung ausgibt. Eine Zufallszahl zwischen 0 und 1 erzeugt man mit Hilfe der Pythonfunktion `random.random()`.

Antwort:

```
import random
x = 0
for i in range(1000):
    if random.random() < 0.5:
        x += 1
    else:
        x -= 1

print "Abstand␣=", abs(x)
```

Aufgabe 33: (2 Punkte)

Forme die folgende Pythonschleife in einen entsprechenden NumPy-Ausdruck (ohne Schleife in Python) um. Dabei ist `k` nicht notwendigerweise die Länge von `a` oder `b`.

```
for i in range(k): a[i] += b[i]
```

Antwort:

```
a[:k] += b[:k]
```

10 Weitere Aufgaben (6 Punkte)

Aufgabe 34: (3 Punkte)
Wenn man ein klassisches Fadenpendel als harmonischen Oszillator löst, welche der folgenden Näherungen sind numerischer und welche analytischer Natur?

- Diskretisierung der Zeit
- Reibungsfreiheit
- Rundungsfehler
- $\sin x = x$ für kleine Winkel

Antwort:

- Diskretisierung der Zeit: numerisch
- Reibungsfreiheit: analytisch
- Rundungsfehler: numerisch
- $\sin x = x$ für kleine Winkel: analytisch

Aufgabe 35: (3 Punkte)
Wieso ist es nicht sinnvoll, für das Relaxationsverfahren zum Lösen linearer Gleichungssysteme eine Schrittweitensteuerung zu implementieren, wie sie beim Optimierungsverfahren der steilsten Gradienten verwendet wird (Armijo-Schrittweitensteuerung)?

Antwort:

Bei der Armijo-Schrittweitensteuerung wird durch einen einfachen Algorithmus das Minimum der Qualitätsfunktion entlang der Richtung des Gradienten gesucht. Obwohl das Relaxationsverfahren dem Gradientenabstiegsverfahren formal ähnlich sieht, so gibt es dabei keine solche Qualitätsfunktion für die Lösung des nächsten Iterationsschrittes, so daß es nicht möglich ist, das Miminum davon zu suchen.