Technische Universität Wien
Institut für Analysis und Scientific Computing
Prof. Dr. Ansgar Jüngel

Sommer 2014

Seminar „Numerical Digit Challenge“

Dieses Dokument enthält die Seminarausarbeitungen der Teilnehmenden

1. Philip Lederer, Carl-Martin Pfeiler, Christoph Wintersteiger
2. Stephan Pfannerer, Georg Simbrunner, Bernhard Wipfler
3. Nico Amann, Nathanael Skrepek
4. Maximilian Bernkopf, Axel Böhm, Lisa Mayer

Seminararbeit

The Digit Challenge

Philip Lederer, Carl-Martin Pfeiler, Christoph Wintersteiger
1025754 1027222 1026218

26. Juni 2014

Inhaltsverzeichnis

1 Aufgabe 1

(P. Lederer, C. Pfeiler, C. Wintersteiger)

1.1 Aufgabenstellung

Sei $\mathbb{P}$ die Menge aller Primzahlen. Berechne

$$s = \sum_{p \in \mathbb{P}} \frac{p(p-1)}{p^\pi} \,.$$

1.2 Erste Überlegungen

Diese Reihe konvergiert sehr langsam, wie man an den folgenden Partialsummen sieht.

$$\begin{aligned}
\sum_{\substack{p \in \mathbb{P} \\ p<5000}} \frac{p(p-1)}{p^\pi} &= 1.2605047907783012970 \\
\sum_{\substack{p \in \mathbb{P} \\ p<10000}} \frac{p(p-1)}{p^\pi} &= 1.2827186522330019635 \\
\sum_{\substack{p \in \mathbb{P} \\ p<20000}} \frac{p(p-1)}{p^\pi} &= 1.3013077653321417593
\end{aligned}$$

Das Problem lässt sich jedoch mit Hilfe der folgenden Definition umformulieren.

Definition 1.1. *Für eine komplexe Zahl z mit Realteil $\mathrm{Re}(z) > 1$ ist die* ***Primzetafunktion*** *P wie folgt definiert:*

$$P(z) := \sum_{p \in \mathbb{P}} \frac{1}{p^z}. \tag{1}$$

Wir werden das Problem als Differenz zweier Funktionswerte dieser Funktion darstellen und dieses Problem betrachten. Es gilt

$$s = \sum_{p \in \mathbb{P}} \frac{p(p-1)}{p^\pi} = \sum_{p \in \mathbb{P}} \frac{1}{p^{\pi-2}} - \sum_{p \in \mathbb{P}} \frac{1}{p^{\pi-1}} = P(\pi-2) - P(\pi-1)\,,$$

mit der Primzetafunktion P. Die Auswertung mit WolframAlpha liefert nach wenigen Sekunden Rechenzeit

$$\begin{aligned}
s &= P(\pi-2) - P(\pi-1) \\
&= 1.41702525894502464667989471995404264872028939967775267916650 2...
\end{aligned}$$

Bei der numerischen Berechnung der Primzetafunktion (1) können nur endlich viele Reihenglieder berechnet werden. Dadurch entsteht ein Fehler, der dem Reihenrest entspricht. Um nun eine gewisse Genauigkeit garantieren zu können, werden wir im nächsten Abschnitt versuchen, diesen Fehler abzuschätzen.

1.3 Fehlerabschätzung

Nun wollen wir einige theoretische Überlegungen zur Berechnung der Primzetafunktion anstellen. Dazu benötigen wir die Möbiusfunktion

$$\mu(n) := \begin{cases} (-1)^k & , \text{ falls } n \text{ quadratfrei}, k = \text{Anzahl der Primfaktoren von } n \\ 0 & , \text{ sonst}\,. \end{cases}$$

Mit Hilfe dieser Funktion und der Zetafunktion ζ gilt [7]

$$P(z) = \sum_{n>0} \mu(n) \frac{\log(\zeta(nz))}{n}\,. \tag{2}$$

Für $z \in \mathbb{R}^+$ fällt die Zetafunktion monoton und es gilt

$$\lim_{n\to\infty} \zeta(nz) = 1\,.$$

Dadurch ist die Funktion $\log(\zeta(nz))$ für alle $n \in \mathbb{N}$ wohldefiniert.
Zur Abschätzung des Fehlers wählen wir ein $N \in \mathbb{N}$ und definieren das Restglied

$$R(N+1) := P(z) - \sum_{n=1}^{N} \mu(n) \frac{\log(\zeta(nz))}{n} = \sum_{n=N+1}^{\infty} \mu(n) \frac{\log(\zeta(nz))}{n}\,.$$

Für ein gegebenes $z \in \mathbb{R}$ existiert der Grenzwert

$$c := \lim_{n\to\infty} (\log(\zeta(nz)))^{\frac{1}{n}} \tag{3}$$

und lässt sich numerisch berechnen. Nun gilt mit einem $a \in \mathbb{R}^+$

$$|\log(\zeta(nz))| = \log(\zeta(nz)) \le ac^n \quad \forall n \in \mathbb{N}\,. \tag{4}$$

Damit lässt sich der Betrag des Restglieds durch eine konvergente Reihe abschätzen.

$$|R(N+1)| = \left| \sum_{n=N+1}^{\infty} \mu(n) \frac{\log(\zeta(nz))}{n} \right| \le \sum_{n=N+1}^{\infty} |\log(\zeta(nz))| \le a \sum_{n=N+1}^{\infty} c^n = \frac{ac^{N+1}}{1-c}$$

Will man nun die ersten p Nachkommastellen berechnen, lässt sich aus obiger Restgliedabschätzung bestimmen, bis zu welchen Index $N \in \mathbb{N}$ die Reihe in (2) aufsummiert werden muss. Es gilt

$$|R(N+1)| \le \frac{ac^{N+1}}{1-c} < 10^{-p} \quad \Leftrightarrow \quad N > \frac{\log(1-c) - \log(a10^p)}{\log(c)} - 1\,.$$

1.4 Implementation in Maple

Die näherungsweise Berechnung des in (3) beschriebenen Grenzwertes muss mit einer hohen Rechengenauigkeit durchgeführt werden, da die Funktion $f(n) = \log(\zeta(zn))$ für ein festes $z \in \mathbb{R}^+$ in n stark fällt uns sonst numerisch verschwindet.

```
> Digits:=100;
```

```
> f:=(n,s)->log(Zeta(n*s));
```

$$f := (n, s) \mapsto \ln(\zeta(ns))$$

```
> g:=(n,a,c)->a*c^n;
```

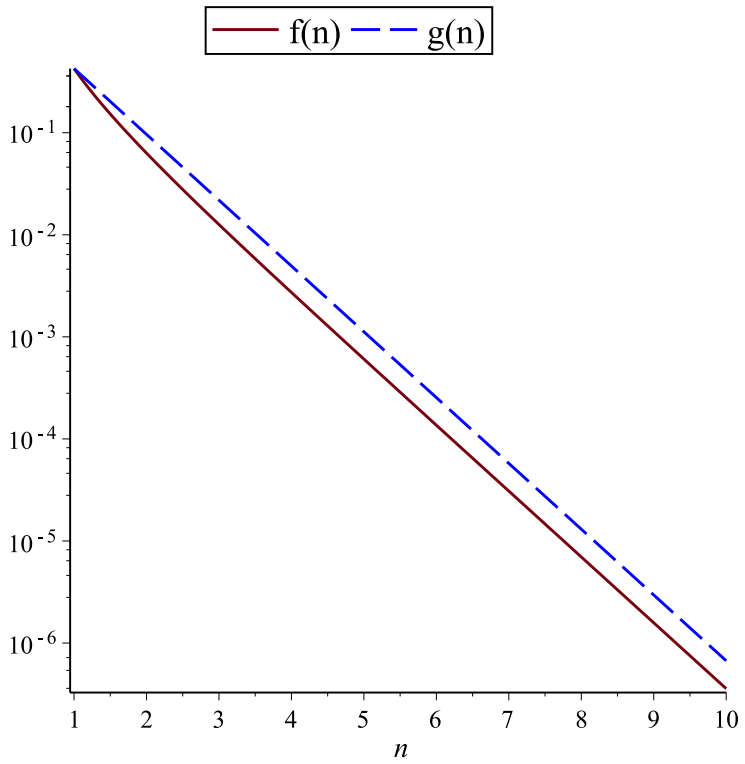
$$g := (n, a, c) \mapsto ac^n$$

```
> c1:=evalf(map(n->f(n,Pi-1)^(1/n),100));
> a1:=evalf(f(1,Pi-1)/c1);
> c2:=evalf(map(n->f(n,Pi-2)^(1/n),100));
> a2:=evalf(f(1,Pi-2)/c2);
```

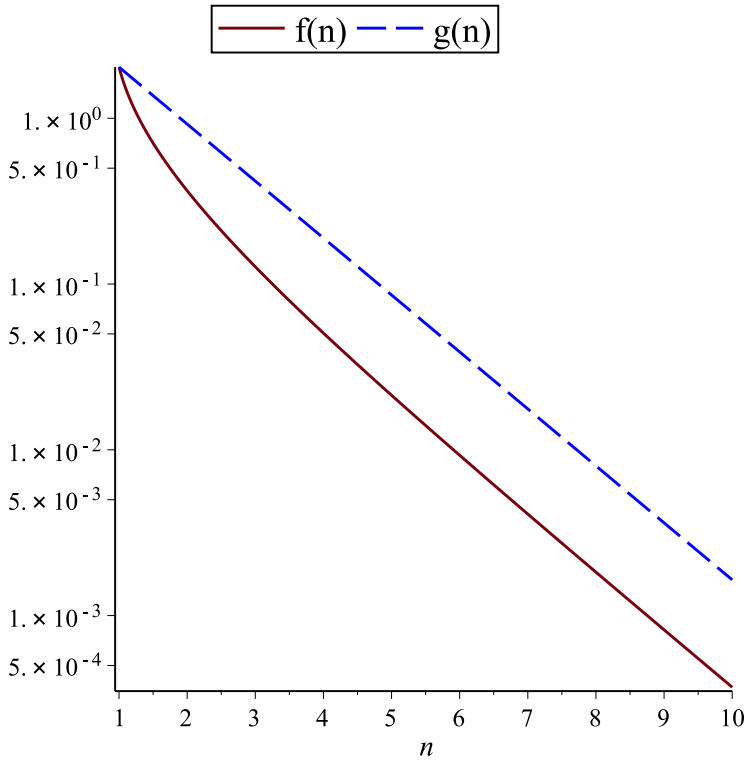
$$c1 := 0.22662946459352174604$$
$$a1 := 1.87544639996363252849$$
$$c2 := 0.45325892918704349208$$
$$a2 := 4.48904106651936766980$$

In den folgenden semilogarithmischen Plots sieht man sehr gut, dass sich $f(n) = \log(\zeta(nz))$ für ein festes $z \in \mathbb{R}$ durch eine Funktion $g(n) = ac^n$ beschränken lässt. Der erste Plot zeigt die Funktionen $f(n)$ und $g(n)$ für $z = \pi - 1$ und der zweite für $z = \pi - 2$.

```
> pf1:=plots[logplot](f(n,Pi-1),n=1..10,legend="f(n)"):
> pc1:=plots[logplot](g(n,a1,c1),n=1..10,color='blue', ...
>                     linestyle=dash,legend="g(n)"):
> display(pf1,pc1);
```



```
> pf2:=plots[logplot](f(n,Pi-2),n=1..10,legend="f(n)"):
> pc2:=plots[logplot](g(n,a2,c2),n=1..10,color='blue', ...
>                     linestyle=dash,legend="g(n)"):
> display(pf2,pc2);
```



Basierend auf der Primfaktorzerlegung `ifactor(m)`, welche die Anzahl der Primfaktoren von m und deren Vielfachheiten berechnet, wurde die Möbiusfunktion implementiert.

```
> moebius := proc(m::integer)
> local pfs:=ifactors(m);
> local n := 1;
> local cnt := 0;
> local pf := [0,0];
> if m = 1 then
>     return 1;
> end if;
> while pfs[1]*n < m do
>     pf := pfs[2][cnt+1];
>     if pf[2] > 1 then
>        return 0;
>     else
>         n := n * pf[1];
>         cnt := cnt + 1;
>     end if;
> end do;
> return (-1)^cnt;
> end proc:
```

Die Implementation der Primzetafunktion $P(z)$ berechnet intern die Konstanten c und a. Damit kann dann die Anzahl N der benötigten Reihenglieder bestimmt werden, die aufsummiert werden müssen, um die gewünschte Genauigkeit `prec` zu erreichen.

```
> primezeta := proc(z,prec::integer)
> local sum := 0;
> local diff := 1;
> local moe := 1;
> local cnt::integer := 1;
> local dig := Digits;
> local c, a, n;
> local i;
> Digits := 100;
> c:=evalf(log(Zeta(100*z))^(1/100));
> a:=evalf(log(Zeta(z))/c);
> n := ceil(log((1-c)/(a*10^prec))/log(c)-1);
> Digits := dig;
> print("z = ",z);
> print("N = ",n);
> for i from 1 to n do
>     moe := moebius(i);
>     if evalb(moe <> 0) then
>         diff := log(Zeta(i*s))/i;
>         sum := sum + moe * diff;
>    end if;
> end do;
> return evalf(sum);
> end proc:
```

Nun wollen wir kontrollieren, ob unsere Implementation der Primzetafunktion dieselben Ergebnisse wie WolframAlpha liefert. Dazu setzen wir die Rechengenauigkeit auf 60 Digits und berechnen die ersten 50 Nachkommastellen von $P(\pi - 1)$ und $P(\pi - 2)$. Dabei wird die Referenzlösung von WolframAlpha jeweils mit `ref` bezeichnet.

```
> Digits := 60:
> interface('displayprecision' = 55):
```

```
> sol1:=primezeta(Pi-1,50);
> # WolframAlpha
> ref1:=0.38916133038450510607662819851630231948978534365916355189122 3;
> err1:=abs(sol1-ref1);
```

$$z = \pi - 1$$

$$N = 78$$

$$sol1 := 0.3891613303845051060766281985163023194897853436591635666$$

$$ref1 := 0.3891613303845051060766281985163023194897853436591635519$$

$$err1 := 1.4740083 \times 10^{-53}$$

```
> sol2:=primezeta(Pi-2,50);
> # WolframAlpha
> ref2:=1.80618658932952975275652291847034496821007474333691623105772 6;
> err2:=abs(sol2-ref2);
```

$$z = \pi - 2$$

$$N = 148$$

$$sol2 := 1.80618658932952975275652291847034496821007474333691623 61$$
$$ref2 := 1.80618658932952975275652291847034496821007474333691623 11$$
$$err2 := 5.08675 \times 10^{-54}$$

Bei der Berechnung von $P(\pi - 1)$ wird die Reihe bis $N = 78$ ausgewertet und es stimmen 52 Nachkommastellen mit der Referenzlösung überein. $P(\pi - 2)$ liefert 53 richtige Nachkommastellen, wobei bis $N = 148$ aufsummiert wird. Um abschätzen zu können, wie sich N bei höheren Genauigkeiten p verhält, vergleichen wir in Tabelle 1 die für die jeweilige Genauigkeit benötigte Anzahl an Reihengliedern. Dabei steht $N_{\pi-1}$ für die Anzahl der Reihenglieder für die Auswertung von $P(\pi - 1)$ und analog $N_{\pi-2}$ für die von $P(\pi - 2)$. Wie man gut erkennen kann, wachsen $N_{\pi-1}$ und $N_{\pi-2}$ linear in p. Somit kann man sehr gut abschätzen, wie sich der Rechenaufwand bei Erhöhung der Genauigkeit ändert und wir können die Primzetafunktion theoretisch beliebig genau auswerten.

p	$N_{\pi-1}$	$N_{\pi-2}$
16	25	49
32	50	95
64	99	188
128	199	375
256	397	747
512	794	1492
1024	1588	2982

Tabelle 1: Auswertungen Primzetafunktion

Nun wenden wir uns wieder unserem eigentlichen Problem zu, bei dem wir $s = P(\pi - 2) - P(\pi - 1)$ berechnen wollen. Für dieses s stimmen 53 Nachkommastellen mit der Referenzlösung überein.

```
> s:=sol2-sol1;
> ref:=ref2-ref1;
> err:=abs(s-ref);
```

$$s := 1.41702525894502464667989471995404264872028939967775266 95$$
$$ref := 1.41702525894502464667989471995404264872028939967775267 92$$
$$err := 9.65334 \times 10^{-54}$$

Somit kann auch s theoretisch beliebig genau berechnet werden. Dazu definieren wir folgende Funktion, die s mit der gewünschten Genauigkeit auswertet.

```
> calcsum:=proc(prec::integer)
> return evalf(primezeta(Pi-2,prec)-primezeta(Pi-1,prec));
> end proc:
```

Abschließend wollen wir die ersten 1000 Nachkommastellen von s berechnen, was in Maple etwa zwei Minuten in Anspruch nimmt. Dabei werden für $P(\pi - 1)$ 1551 und für $P(\pi - 2)$ 2912 Reihenglieder aufsummiert.

```
> Digits:=1010:
> interface('displayprecision' = -1):
> s:=calcsum(1000);
```

$$z = \pi - 2$$
$$N = 2912$$
$$z = \pi - 1$$
$$N = 1551$$

$s = 1.41702525894502464667989471995404264872028939967775267916650262120972230367$
$292792288230740045164225687961089332754335640671640263722796672446549688786446744955$
$502245739594710794123237624622375377087570686762665542109714369346980711002871063777$
$764692948694076190426590074872389061131899391269173751064835086540235341901890382305$
$844104355420514179286167140934434279025220339965953191837871961896188903350469762332$
$949670193657339520125678273079369718146808537061989118198115660867322406156689209011$
$375657016696956826221651054329155650243562157089140998466842511698119933815090947$
$662098348612094755490049752144671960284116574085723059346129421390580500705384534387$
$808773706058439540064924861330187610817807044519409605709645579408545486491229108258$
$531300414905271826257360119933283301687996228526104107513744347423751586544570461$
$846410651378977066623394426505640369897246203083477436866412369548331226372680930180$
$738800923907443756183724770963123615545083026496632264898994237153529384352901811904$
1157634513

2 Aufgabe 2

(C. Wintersteiger)

2.1 Aufgabenstellung

Sei $\mathbb{P}$ die Menge der Primzahlen. Die unendliche Matrix $A = (a_{ij})_{i,j\geq 1}$, definiert durch

$$a_{ij} := \begin{cases} \frac{1}{i+j-1} & , \ i+j \in \mathbb{P} \\ \frac{1}{(i+j-1)^2} & , \ sonst\,, \end{cases} \tag{5}$$

bildet einen unendlich-dimensionalen Operator auf ℓ^2. Welchen Wert hat

$$a = \|A\|_2 = \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}\,, \tag{6}$$

wobei $\|x\|_2 = \sum_{i=1}^{\infty} x_i^2$.

2.2 Theoretische Grundlagen

Der gesuchte Wert a entspricht der Operatornorm. Zur Berechnung dieser Norm benötigt man die Beschränktheit des unendlich-dimensionalen Operators und die Eigenschaft, dass die Normen für endliche Teilmatrizen ein monoton wachsende Folge bilden. Dazu beweisen wir folgendes Lemma.

Lemma 2.1. *Sei $A = (a_{ij})_{i,j\geq 1}$ wie in (5) definiert und sei A_n die n-dimensionale Teilmatrix von A. Für $1 \leq n \leq m$ gilt*

$$\|A_n\|_2 \leq \|A_m\|_2 \leq \lim_{k\to\infty} \|A_k\|_2 = \|A\|_2 \leq \pi \tag{7}$$

Beweis: Sei P_n die Orthogonalprojektion von ℓ^2 auf den n-dimensionalen Teilraum. Dann gilt

$$\|A_n\|_2 = \|P_n A P_n\|_2 = \|P_n P_m A P_m P_n\|_2 \leq \|P_n\|_2 \|P_m A P_m\|_2 \|P_n\|_2 \leq \|A_m\|_2 \ .$$

Dadurch erhält man die Ungleichung

$$\|A_n\|_2 \leq \|A_m\|_2 \leq \lim_{k\to\infty} \|A_k\|_2 \leq \|A\|_2 \ . \tag{8}$$

Da eine vollständige Basis des ℓ^2 existiert, gilt $\lim_{n\to\infty} P_n x = x$ für alle $x \in \ell^2$ und damit

$$\|Ax\|_2 = \lim_{n\to\infty} \|P_n A P_n x\|_2 \leq \lim_{n\to\infty} \|A_n\|_2 \|x\|_2 \ . \tag{9}$$

Bildet man das Supremum von (9) über alle x mit $\|x\|_2 = 1$, folgt gemeinsam mit (8)

$$\|A\|_2 = \lim_{n\to\infty} \|A_n\|_2 \ .$$

Damit bleibt nur noch die Beschränktheit der Operatornorm zu zeigen. Dazu verwenden wir die Hilbertmatrix $H = (h_{ij})_{i,j\geq 1}$, welche durch $h_{ij} := \frac{1}{i+j-1}$ definiert ist. Diese bietet sich für eine Abschätzung an, da für die Operatornorm $\|H\|_2 \leq \pi$ gilt. Für $i + j \in \mathbb{P}$ sind die Einträge

der Matrizen A und H ident und für alle anderen Einträge gilt $a_{ij} = h_{ij}^2$.
Sei nun $x = (x_1, \ldots, x_n)$ mit $\|x\|_2 = 1$. Da $a_{ij} > 0$ für alle $i, j \in \mathbb{N}$, folgt

$$\|A_n x\|_2^2 = \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} a_{ij} x_j \Big)^2 \leq \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} a_{ij} |x_j| \Big)^2 = \|A_n y\|_2^2 \ ,$$

wobei $y := (|x_1|, \ldots, |x_n|)$ und $\|x\|_2 = \|y\|_2$. Durch Bildung der Suprema erhält man

$$\sup_{\|x\|_2=1} \|A_n x\|_2 \leq \sup_{\|y\|_2=1} \|A_n y\|_2 = \sup_{\substack{\|x\|_2=1 \\ x_i \geq 0}} \|A_n x\|_2 \ . \tag{10}$$

Weil auf der linken Seite das Supremum über eine größere Menge gebildet wird, gilt anstelle der Ungleichung in (10) Gleichheit. Somit gilt für die Operatornormen

$$\|A\|_2 = \sup_{\substack{\|x\|_2=1 \\ x_i \geq 0}} \|Ax\|_2 \quad \text{und} \quad \|H\|_2 = \sup_{\substack{\|x\|_2=1 \\ x_i \geq 0}} \|Hx\|_2 \ .$$

Da für unsere Matrixeinträge $a_{ij} \leq h_{ij}$ für alle $i, j \in \mathbb{N}$ gilt, folgt

$$\|A\|_2 = \sup_{\substack{\|x\|_2=1 \\ x_i \geq 0}} \left(\sum_{i=1}^{\infty} \Big(\sum_{j=1}^{\infty} a_{ij} x_j \Big)^2 \right)^{\frac{1}{2}} \leq \sup_{\substack{\|x\|_2=1 \\ x_i \geq 0}} \left(\sum_{i=1}^{\infty} \Big(\sum_{j=1}^{\infty} h_{ij} x_j \Big)^2 \right)^{\frac{1}{2}} = \|H\|_2 \leq \pi \ .$$

■

Durch obiges Lemma können wir uns für die Berechnung der Operatornorm auf endliche Teilmatrizen A_n beschränken. Weiters gilt durch die Symmetrie der Matrix

$$\|A\|_2 = \rho(A) = \max \{ |\lambda| : \lambda \text{ ist Eigenwert von } A \} \ ,$$

wodurch die gesuchte reelle Zahl dem betragsgrößten Eigenwert entspricht. Dieser kann beispielsweise mit den Matlab-Funktionen `norm()` oder `normest()` berechnet werden. Für diese Funktionen muss die ganze Matrix A_n gespeichert werden, was sehr schnell zu Speicherproblemen führt. Durch die spezielle Struktur der Matrix A, muss nur die erste Zeile und die letzte Spalte gespeichert werden, wie man sehr schön an der folgenden Teilmatrix erkennen kann.

$$A_5 = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{9} & \frac{1}{4} & \frac{1}{25} \\ \frac{1}{2} & \frac{1}{9} & \frac{1}{4} & \frac{1}{25} & \frac{1}{6} \\ \frac{1}{9} & \frac{1}{4} & \frac{1}{25} & \frac{1}{6} & \frac{1}{49} \\ \frac{1}{4} & \frac{1}{25} & \frac{1}{6} & \frac{1}{49} & \frac{1}{64} \\ \frac{1}{25} & \frac{1}{6} & \frac{1}{49} & \frac{1}{64} & \frac{1}{81} \end{pmatrix}$$

Allgemein gilt

$$A_n = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ a_2 & a_3 & & \iddots & \vdots \\ a_3 & & \iddots & & \vdots \\ \vdots & \iddots & & & a_{2n-2} \\ a_n & \cdots & \cdots & a_{2n-2} & a_{2n-1} \end{pmatrix} .$$

Somit muss nur der Vektor

$$A_n = (a_1, a_2, \ldots, a_{2n-1}) \tag{11}$$

gespeichert werden, wodurch sich der Speicheraufwand von n^2 auf $2n-1$ reduzieren lässt. Dadurch können aber keine Standardfunktionen zur Berechnung der Norm verwendet werden, da diese die gesamte Matrix benötigen würden. Durch die Größe der Teilmatrizen, für die die Norm berechnet werden soll, liegt es nahe, ein iteratives Verfahren zu verwenden. Viele benötigen allerdings die inverse Matrix, was sich durch die effiziente Speicherung nicht realisieren lässt. Daher fiel unsere Wahl auf die Poweriteration oder auch bekannt unter Power Method.

Poweriteration:
Sei $A \in \mathbb{R}^{n\times n}$ diagonalisierbar mit den Eigenwerten $\lambda_1, \ldots, \lambda_n$ so, dass

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

Mit den linear unabhängige Eigenvektorenn $x_1, \ldots, x_n$ kann ein beliebiger Vektor $x^0 \in \mathbb{R}^n$ dargestellt werden als

$$x^0 = c_1x_1 + c_2x_2 + \cdots + c_nx_n \ .$$

Nun gilt

$$\begin{aligned} x^k &= A^k x^0 \\ &= \sum_{i=1}^{n} c_i A^k x_i = \sum_{i=1}^{n} c_i \lambda_i^k x_i \\ &= \lambda_1^k \left(c_1 x_1 + \sum_{i=2}^{n} c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i \right). \end{aligned}$$

Da $|\lambda_1| > |\lambda_i|$ für $i \in \{2, \ldots, n\}$, konvergiert x^k gegen ein Vielfaches von x_1. Somit lässt sich das Iterationsverfahren formulieren als:
Sei x^0 ein beliebiger Startvektor mit $\|x^0\| = \lambda^0 = 1$, $e^0 = 1$ und τ eine vorgegebene Toleranz.
`while` $e_k > \tau$
  $x^{k+1} = Ax^k$
  $\lambda^{k+1} = \sqrt{(x^{k+1})^T x^{k+1}}$
  $x^{k+1} = \frac{x^{k+1}}{\lambda^{k+1}}$
  $e^{k+1} = \lambda^{k+1} - \lambda^k$
`end`
Wie bereits oben erwähnt, konvergiert die Folge der x^k gegen ein Vielfaches des zum betragsmäßig größten Eigenwert λ_1 zugehörigen Eigenvektor x_1. Des weiteren lässt sich zeigen, dass

$$\lim_{k\to\infty} \|x^{k+1}\|_2 = \lim_{k\to\infty} \lambda^{k+1} = \lambda_1 \,. \tag{12}$$

Dazu sei o.B.d.A. $\|x_1\|_2 = 1$ und $x^k \in \mathbb{R}^n$ der Vektor nach der k-ten Iteration. Da x^k gegen x_1 konvergiert, gilt

$$x^k = x_1 + r^k \quad \text{mit} \quad \lim_{k\to\infty} \|r^k\|_2 = 0 \,,$$

wobei $r^k \in \mathbb{R}^n$ den Fehler im k-ten Iterationsschritt beschreibt. Somit gilt

$$x^{k+1} = Ax^k = Ax_1 + Ar^k = \lambda_1 x_1 + Ar^k \tag{13}$$

und

$$\|x^{k+1}\|_2 \leq \|Ar^k\|_2 + |\lambda_1| \underbrace{\|x_1\|_2}_{=1} .$$

Daraus folgt nun

$$\lim_{k\to\infty} \|x^{k+1}\|_2 \leq \underbrace{\lim_{k\to\infty} \|Ar^k\|_2}_{=0} + |\lambda_1| = |\lambda_1| . \tag{14}$$

Subtrahiert man Ar^k von Gleichung (13) und wendet die Norm an, erhält man

$$|\lambda_1| \underbrace{\|x_1\|_2}_{=1} = \|x^{k+1} - Ar^k\|_2 \leq \|x^{k+1}\|_2 + \|Ar^k\|_2 .$$

Durch bilden des Grenzwertes folgt

$$|\lambda_1| \leq \lim_{k\to\infty} \|x^{k+1}\|_2 + \underbrace{\lim_{k\to\infty} \|Ar^k\|_2}_{=0} = \lim_{k\to\infty} \|x^{k+1}\|_2 . \tag{15}$$

Zusammen ergeben die Gleichungen (14) und (15)

$$\lim_{k\to\infty} \|x^{k+1}\|_2 = |\lambda_1| ,$$

womit die Aussage (12) gezeigt ist.

2.3 Erste Implementation

Da sowohl bei der Matrix-Vektor-Multiplikation, als auch bei der Berechnung der λ^k die Summanden monoton fallend sind - bis auf jene, bei denen die Summe von Zeilen- und Spaltenindex eine Primzahl ist - haben wir uns dazu entschlossen, die Berechnung „verkehrt herum“ zu implementieren. Damit wird bei sämtlichen Summationen bei den kleinen Summanden begonnen, um so Rechenfehler zu minimieren.
Sei $A \in \mathbb{R}^{2n-1}$ die Vektordarstellung unserer Matrix A, wie in (11) beschrieben und seien $x, y \in \mathbb{R}^n$, dann kann die Matrix-Vektor-Multiplikation "$y = Ax$" durch folgende Darstellung einfach realisiert werden.

$$y_i = \sum_{j=1}^{n} A_{i+j-1} x_j \quad \forall i \in \{1, \ldots, n\}$$

Da nach dem ersten Schritt immer nur eine geringe Anzahl an Iterationen nötig ist (siehe Tabelle 2), um ein genaues Ergebnis zu erreichen, wird für jede Teilmatrix so lange iteriert, bis der Eigenwert durch numerische Rundungsfehler wieder kleiner wird. Die Implementation ist so gut wie möglich mit `openmp` parallelisiert, jedoch stößt man bei größeren Matrixdimensionen sehr schnell an die Grenzen der Rechenleistung. Die Berechnung der Ergebnisse in Tabelle 2 dauerte bereits mehrere Stunden, wodurch es nun den Rechenaufwands von $\mathcal{O}(n^2)$ zu reduzieren

gilt.

n	λ_n	Iterationen
5.0e5	1.395771567478847	20
1.0e6	1.395771775535150	8
1.5e6	1.395771843619359	6
2.0e6	1.395771877122765	6
2.5e6	1.395771896997722	6
3.0e6	1.395771910195626	5
3.5e6	1.395771919552097	6
4.0e6	1.395771926530703	7
4.5e6	1.395771931944202	6

Tabelle 2: Ergebnisse der ersten Implementation

n	λ_n	Iterationen
5.0e5	1.395771567478846	20
1.0e6	1.395771775535150	7
1.5e6	1.395771843619358	6
2.0e6	1.395771877122764	7
2.5e6	1.395771896997722	6
3.0e6	1.395771910195626	6
3.5e6	1.395771919552096	6
4.0e6	1.395771926530703	5
4.5e6	1.395771931944202	7
5.0e6	1.395771936259346	5
5.5e6	1.395771939765920	5
6.0e6	1.395771942673666	6
6.5e6	1.395771945130775	6
7.0e6	1.395771947243055	5
7.5e6	1.395771949058432	5
8.0e6	1.395771950646313	5

Tabelle 3: Ergebnisse der Implementation mittels FFT

2.4 Implementation mittels FFT

Sei $A \in \mathbb{R}^{2n-1}$ wieder die Vektordarstellung unserer Matrix A und seien $x, y \in \mathbb{R}^n$. Dann lässt sich wie oben erwähnt, das Matrix-Vektor-Produkt "$y = Ax$" schreiben als

$$y_i = \sum_{j=1}^{n} A_{i+j-1} x_j \quad \forall i \in \{1, \dots, n\}\,. \tag{16}$$

Da sich ein Faltungsprodukt mit Rechenaufwand $\mathcal{O}(n \log(n))$, statt den bisherigen $\mathcal{O}(n^2)$, berechnen lässt, wollen wir (16) mit Hilfe einer Faltung darstellen. Seien $f, g \in \mathbb{R}^n$, dann ist das diskrete Faltungsprodukt $f * g$ definiert als

$$(f * g)_i = \sum_{j=1}^{n} f_{i+1-j}\, g_j \quad \forall i \in \{1, \dots, n\}\,, \tag{17}$$

wobei die Vektoren falls nötig mit Null fortgesetzt werden.
Um diese Eigenschaft ausnützen zu können, formen wir Gleichung (16) wie folgt um

$$y_i = \sum_{j=1}^{n} A_{i+j-1} x_j = \sum_{j=1}^{n} A_{i+n-j} x_{n+1-j}$$

und definieren den Vektor $\tilde{x}_j := x_{n+1-j}$. Somit erhalten wir

$$y_i = \sum_{j=1}^{n} A_{i+n-j} \tilde{x}_j\,,$$

was dem Faltungsprodukt (17) schon sehr ähnlich sieht. Der einzige Unterschied ist, dass die A_{i+n-j} um $n-1$ verschoben sind. Insgesamt erhalten wir also

$$y_i = (A * \tilde{x})_{n-1+i} \quad \forall i \in \{1, \ldots, n\}\,.$$

Das Faltungsprodukt kann mit Hilfe der FFT $\mathcal{F}$ wie folgt berechnet werden.

$$\tilde{y} = (A * \tilde{x}) = \mathcal{F}^{-1}(\mathcal{F}(A) \odot \mathcal{F}(\tilde{x}))$$

Hierbei ist $\odot$ das punktweise Produkt. Da wir von dem Vektor $\tilde{y} \in \mathbb{R}^{2n-1}$ nur Einträge $\tilde{y}_n$ bis $\tilde{y}_{2n-1}$ benötigen, werden etwa doppelt so viele Zahlen berechnet wie nötig wären. Allerdings reduziert sich dabei der Rechenaufwand von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n\log(n))$, was bei größerem n ein überwiegender Vorteil ist.
Durch die Verwendung der FFT aus der `C++` Intel MKL Bibliothek konnten die Ergebnisse in Tabelle 3 in wenigen Minuten berechnet werden. Die Befürchtungen, dass durch Transformation und Rücktransformation die Eigenwerte zu ungenau sind, haben sich nicht bewahrheitet. Wie man sieht, unterscheiden sich die Ergebnisse der Berechnungen ohne FFT in Tabelle 2 und mit FFT in Tabelle 3 nur in der 15ten Nachkommastelle. Dies ist vor allem auf Rundungsfehler durch die verwendete `double`-Precision zurückzuführen. Leider scheint die von Intel zur Verfügung gestellte FFT, Probleme mit Dimensionen größer als $16 \cdot 10^6$ zu haben, wodurch wir gezwungen waren, uns nach alternativen Bibliotheken umzuschauen.
Dabei sind wir auf ein Paket namens FFTW [1] gestoßen, welches problemlos auch Vektoren größerer Dimensionen transformieren kann. Somit konnten wir die Operatornorm für Teilmatrizen, bis zu einer Dimension von 2^{30} berechnen. In Tabelle 4 steht n für eine Matrixdimension 2^n. Des weiteren sind für jede Dimension die Ergebnisse für `double`-Precision und `long double`-Precision (in grau) angeführt, was auf unserem Rechenserver 15 bzw. 18 signifikanten Stellen entspricht. Da sich bei der Erhöhung von $n = 29$ auf $n = 30$ schon die zehnte Nachkommastelle der Operatornorm ändert, ist für unsere weiteren Überlegungen `double`-Precision ausreichend. Für $n = 31$ müsste die FFT für Vektoren der Länge $2 \cdot 2^{31} = 2^{32}$ durchgeführt werden, was mit der Implementation der FFTW Bibliothek nicht so einfach realisiert werden kann, da die Dimension als Integer übergeben wird und 2^{32} nicht mehr als Integer dargestellt werden kann. Daher haben wir uns an dieser Stelle mit der erreichten Matrixdimension zufrieden gegeben und im weiteren versucht, aus den bisher berechneten Ergebnissen noch mehr herauszuholen.

[1] http://www.fftw.org

n	λ_n	Iterationen	n	λ_n	Iterationen
1	1.224533032155127	9	16	1.395768612391341	9
	1.224533032155128	10		1.395768612391342	9
2	1.315118469090224	11	17	1.395770341460167	11
	1.315118469090224	12		1.395770341460167	10
3	1.349253088565394	12	18	1.395771178826327	8
	1.349253088565394	11		1.395771178826328	8
4	1.376416735010431	13	19	1.395771587044598	8
	1.376416735010431	13		1.395771587044598	8
5	1.385851752027037	14	20	1.395771785106662	7
	1.385851752027036	14		1.395771785106662	8
6	1.390875057207507	14	21	1.395771881742114	7
	1.390875057207508	13		1.395771881742115	7
7	1.393447190553941	13	22	1.395771928797438	7
	1.393447190553942	14		1.395771928797439	6
8	1.394672657561888	13	23	1.395771951745957	7
	1.394672657561888	14		1.395771951745958	6
9	1.395237698763728	14	24	1.395771962952247	6
	1.395237698763729	13		1.395771962952248	6
10	1.395511572977691	12	25	1.395771968428743	6
	1.395511572977691	13		1.395771968428745	5
11	1.395645890586425	12	26	1.395771971105506	5
	1.395645890586425	12		1.395771971105508	5
12	1.395711456715845	11	27	1.395771972415637	5
	1.395711456715847	11		1.395771972415640	5
13	1.395742497156653	12	28	1.395771973057140	5
	1.395742497156654	11		1.395771973057143	5
14	1.395757740822297	11	29	1.395771973371442	5
	1.395757740822298	11		1.395771973371444	5
15	1.395765051968595	11	30	1.395771973525542	6
	1.395765051968596	11		1.395771973525544	4

Tabelle 4: Ergebnisse mit FFTW

2.5 Extrapolation

Um Verfahren zur Konvergenzbeschleunigung anwenden zu können, benötigen wir eine monotone Folge. Wie man in Tabelle 4 oder im Plot 1 gut erkennen kann, ist die Folge der Operatornormen monoton wachsend. Dies war nach unseren theoretischen Überlegungen auch so zu erwarten.

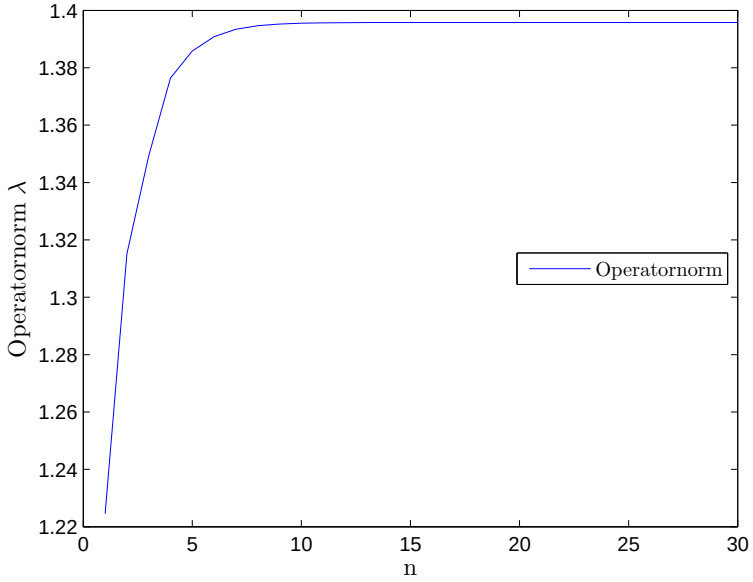


Abbildung 1: Operatornormen

Eine der bekanntesten Möglichkeiten zur Konvergenzbeschleunigung ist das Δ^2-Verfahren von Aitken:
Für eine Folge $(x_m)_{m\in\mathbb{N}}$ mit $\lim_{m\to\infty} x_m = x$ definiert man

$$y_m := x_m - \frac{(x_{m+1} - x_m)^2}{x_{m+2} - 2x_{m+1} + x_m}\,.$$

Dann gilt

$$\lim_{m\to\infty} := \frac{y_m - x}{x_m - x} = 0\,.$$

Alternativ verwenden wir den in [2, Seite 310] erwähnten Wynn'schen Epsilon-Algorithmus. Sei $x \in \mathbb{R}^m$ eine Folge, s eine Matrix mit Zeilenrang m, $s_{k,0} = 0$ und $s_{k,1} = x_k$ für alle $k \in \{1, \ldots, m\}$. Definiert man die weiteren Einträge der Matrix mit folgender Rekursionsformel

$$s_{k,j} = s_{k+1,j-2} + \frac{1}{s_{k+1,j-1} - s_{k,j-1}}\,,$$

dann finden sich die extrapolierten Werte in den Spalten $s_{:,j}$ mit ungeradem j.
Für die Extrapolation mittels Aitken-Verfahren und Wynn'schen Epsilon-Algorithmus wurden jeweils die in Tabelle 4 dargestellten Operatornormen als Startfolge verwendet. Das Aitken-Verfahren wurde einmal angewandt und beim Wynn'schen Epsilon-Algorithmus wurde die Rekursion so weit wie möglich berechnet. Dabei haben wir die in Tabelle 5 aufgelisteten Grenzwerte erhalten, wobei nur die letzten zehn Folgenglieder dargestellt werden. Beide Verfahren liefern 12 übereinstimmende signifikante Stellen. Bei den letzten beiden Werten sind sogar 13 signifikante Stellen ident, jedoch liefert der Wynn'sche Epsilon-Algorithmus eine Folge, bei der eben diese Stelle zwischen 3 und 4 schwankt. Daher wäre es verwegen, die Stelle als richtig anzunehmen.

n	λ_n	$Aitken$	$Wynn$
21	1.39577188174211	1.39577197381273	1.39577197367409
22	1.39577192879744	1.39577197345653	1.39577197367411
23	1.39577195174596	1.39577197359184	1.39577197367408
24	1.39577196295225	1.39577197364706	1.39577197367407
25	1.39577196842874	1.39577197366314	1.39577197367391
26	1.39577197110551	1.39577197366470	1.39577197367395
27	1.39577197241564	1.39577197367160	1.39577197367290
28	1.39577197305714	1.39577197367262	1.39577197367429
29	1.39577197337144	1.39577197367335	1.39577197367392
30	1.39577197352554	1.39577197367377	1.39577197367376

Tabelle 5: Konvergenzbeschleunigung Operatornorm

Eine weitere Möglichkeit Konvergenzbeschleunigung anzuwenden wäre, nicht die Folge der Operatornormen der Teilmatrizen als Startvektor zu verwenden, sondern zu versuchen, das Konvergenzverhalten für jede Matrixdimension zu verbessern. Da wir aber bei der Poweriteration für jede Teilmatrix erst abbrechen, falls der Eigenwert durch Rundungsfehler wieder kleiner wird, konvergieren die beschleunigten Folgen gegen die in Tabelle 4 dargestellten Grenzwerte. Dadurch ergaben sich durch diesen Ansatz keine neuen Erkenntnisse.

n	λ_n	$\lambda_n - \lambda_{n-1}$	$\frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1} - \lambda_{n-2}}$
21	1.39577188174211	0.00000009663545	0.4879049026
22	1.39577192879744	0.00000004705532	0.4869364487
23	1.39577195174596	0.00000002294852	0.4876922992
24	1.39577196295225	0.00000001120629	0.4883230092
25	1.39577196842874	0.00000000547650	0.4886984181
26	1.39577197110551	0.00000000267676	0.4887729238
27	1.39577197241564	0.00000000131013	0.4894459750
28	1.39577197305714	0.00000000064150	0.4896480775
29	1.39577197337144	0.00000000031430	0.4899460831
30	1.39577197352554	0.00000000015410	0.4902931140

Tabelle 6: Abnahmerate Operatornorm

Um unsere bisherigen Ergebnisse zu bestätigen, haben wir im folgenden einen völlig anderen Ansatz verfolgt. Dazu betrachten wir die Abnahmerate

$$\frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1} - \lambda_{n-2}}$$

der Differenzen zweier aufeinander folgender Eigenwerte. Wie man in der letzten Spalte von Tabelle 6 erkennen kann, hat diese Rate immer einen ähnlichen Wert. Dadurch ist es naheliegend, dass sich die λ_n als geometrische Folge

$$\lambda_{n+1} = \lambda_n + cq^n \,.$$

darstellen lassen. Um die Parameter c und q zu bestimmen, wurde die Methode der kleinsten

Quadrate verwendet. Dabei ergeben sich die Gleichungen

$$0 = \sum_{j=k}^{n} q^j(\lambda_{j+1} - \lambda_j - cq^j)\,,$$
$$0 = \sum_{j=k}^{n} j\,q^{j-1}(\lambda_{j+1} - \lambda_j - cq^j)\,.$$

Verwendet man $k = 26$, also versucht man die geometrisch Folge möglichst gut an die letzten fünf Folgenglieder anzupassen, ergibt sich mit Maple genau eine reelle Lösung $c = 0.1521312460852371$ und $q = 0.4895702290295582$. Dabei entspricht q, wie zu erwarten war, etwa der Abnahmerate in der letzten Spalte von Tabelle 6. Somit lässt sich der Grenzwert für unsere Operatornorm wie folgt berechnen.

$$a = \lambda_n + c\sum_{j=n}^{\infty} q^j = \lambda_n + c\frac{q^n}{1-q}\,,$$

was für $n \in \{26, \ldots, 30\}$ folgende Ergebnisse liefert:

n	a
26	1.395771973672781
27	1.395771973672499
28	1.395771973672462
29	1.395771973672685
30	1.395771973673022

Bei diesen Grenzwerten stimmen wieder 12 signifikante Stellen mit unseren extrapolierten Ergebnissen aus Tabelle 5 überein. Wobei hier anzumerken ist, dass die eben berechneten Werte stark schwanken, wenn für die Bestimmung von c und q mehr fünf Werte verwendet werden. Dies liegt daran, dass die Abnahmeraten zu unterschiedlich sind, um eine "globale" geometrische Folge für unsere Eigenwerte zu finden. Da für uns die weitere Entwicklung der Eigenwerte wichtig ist, wurden die Konstanten c und q nur mit den letzten fünf Folgengliedern bestimmt. Ohne der Tatsache, dass wir die so berechneten Grenzwerte schon durch die vorhergehenden Methoden erhalten haben, wäre diese Methode wohl zu ungenau, aber als zusätzliche Absicherung ist sie doch brauchbar.

Somit erhalten wir insgesamt 12 signifikante Stellen

$$a = \|A\|_2 = 1.39577197367\ldots\,.$$

3 Aufgabe 3

(P. Lederer, C. Pfeiler, C. Wintersteiger)

3.1 Aufgabenstellung

Definiert sei die stochastische Folge (x_n) rekursiv durch $x_0 = x_1 = x_2 = 1$ und

$$x_{n+3} = x_{n+2} \pm \frac{1}{3}(x_{n+1} + x_n), \quad n \in \mathbb{N}, n \geq 0, \tag{18}$$

wobei das Vorzeichen mit einer Wahrscheinlicheit $\frac{1}{2}$ positiv oder negativ ist. Berechnen Sie

$$\sigma = \lim_{n\to\infty} |x_n|^{1/n}. \tag{19}$$

3.2 Kurze Analyse

Bei dieser Aufgabe geht es darum, sich Gedanken zu machen, wie der Verlauf der Folge (18) aussieht. Die Startwerte betragen zwar alle 1, aber wie man sieht, hängt das aktuelle Folgenglied von den drei davor liegenden Folgengliedern ab. Deshalb kann es passieren, wenn das Vorzeichen zum Beispiel immer positiv ist, dass der Betrag der Folgenglieder relativ groß wird. Andererseits kann dann durch das Aufreten des jeweils anderen Vorzeichen in (18) die Beträge der Folgenglieder auch schnell wieder sinken. Wir wollen nun im ersten Schritt versuchen den gesuchten Wert (19) einzugrenzen. Klar ist, dass σ nach unten mit 0 beschränkt ist, also

$$0 \leq \sigma,$$

da in (19) nur der Betrag der Folgenglieder x_n verwendet wird. Für eine Abschätzung nach oben betrachten wir jetzt den Fall bei dem immer nur ein positives Vorzeichen gewählt wird. Damit erhält man die Folge

$$y_{n+3} = y_{n+2} + \frac{1}{3}y_{n+1} + \frac{1}{3}y_n$$

bzw. eine explizite Darstellung mittels

$$y_n = \lambda_1^n + \frac{1}{3}\lambda_2^n + \frac{1}{3}\lambda_3^n.$$

Um λ_1, λ_3 und λ_3 zu erhalten löst man zuvor folgende Gleichung

$$\lambda^3 = \lambda^2 + \frac{1}{3}\lambda + \frac{1}{3}.$$

und erhält so

$$\lambda_1 = 1.405786..$$
$$\lambda_2 = -0.202.. + i0.442..$$
$$\lambda_3 = -0.202.. - i0.442..$$

Nun gilt

$$|y_n| = |\lambda_1^n + \frac{1}{3}\lambda_2^n + \frac{1}{3}\lambda_3^n| \leq 3(\max_{i=1,2,3} |\lambda_i|)^n$$

beziehungsweise

$$|y_n|^{1/n} = 3^{1/n}((\max_{i=1,2,3}|\lambda_i|)^n)^{1/n} = 3^{1/n}\max_{i=1,2,3}|\lambda_i|.$$

Da der Betrag der Folgenglieder x_n für eine Mischung der Vorzeichen in (18) immer kleiner ist als der Betrag von y_n (mittels Dreickesungleichung) folgt insgesamt

$$\sigma = \lim_{n\to\infty}|x_n|^{1/n} \leq \lim_{n\to\infty}|y_n|^{1/n} = \lim_{n\to\infty}\underbrace{3^{1/n}}_{\to 1}\max_{i=1,2,3}|\lambda_i| \leq \lambda_1 = 1.405786..$$

3.3 Montecarlo-Methode

Eine der einfachsten Methoden ist die so genannte Montecarlo-Methode. Bei diesem Verfahren werden N zufällige Folgen, wie in (18) gegeben, erzeugt und das $\tilde{n}$-te Folgenglied $x_{\tilde{n}}^{(i)}$ berechnet (wobei $\tilde{n}$ vorgegeben ist). Anschließend werden $\sigma_i = |x_{\tilde{n}}^{(i)}|^{1/\tilde{n}}$ mit $i \in 1,..,N$ berechnet. Die Konvergenz dieser Methode beruht auf dem Gesetz der großen Zahlen. Dazu wurde ein Programm in `C++` implementiert und die Ergebnisse geplottet (siehe Abbildung 2). Es ergaben sich folgende Werte für den Mittelwert und der Varianz (siehe Tabelle 7).

$N\backslash\tilde{n}$	1e6	1e7	1e8
20480	1.009087002140862	1.009087461899701	1.009086077001393
102400	1.009085793939119	1.009085546759626	1.009086206095522
512000	1.009085516295166	1.009086214983217	1.009086277077420

$N\backslash\tilde{n}$	1e6	1e7	1e8
20480	1.601982005425055e-07	1.613561979804300e-08	1.600744840187154e-09
102400	1.629523140371756e-07	1.628541443847066e-08	1.622124246307435e-09
512000	1.617815755375112e-07	1.624282848657560e-08	1.623801488858579e-09

Tabelle 7: Erwartungswerte $\bar{\sigma}$ (oben) und Varianzen $\mathbb{V}$ (unten)

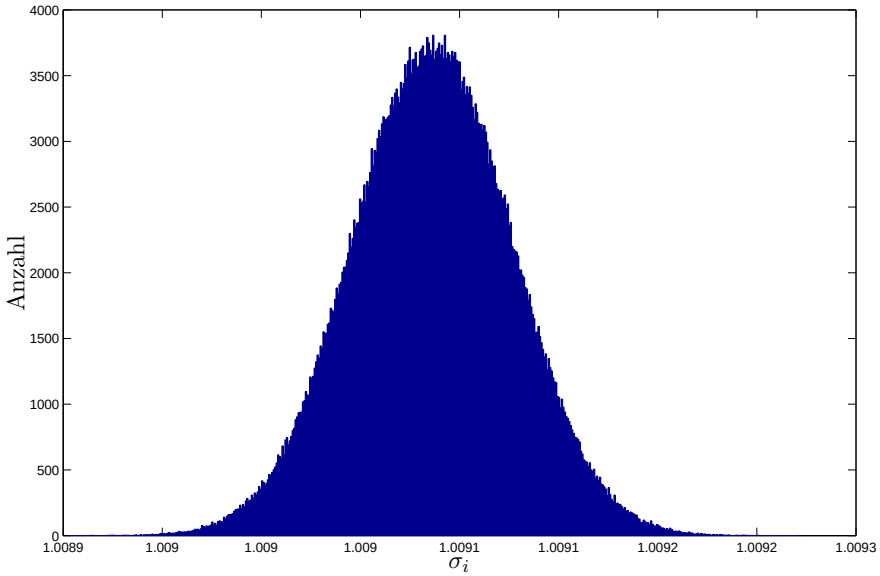


Abbildung 2: Histogramm der σ_i für $N = 512000$ und $\tilde{n} = 1e8$

3.4 Berechnen der Erwartungswerte

Definiert seien die Erwartungswerte

$$\sigma_n := \mathbb{E}(|x_n|^{1/n}).$$

Unter der Voraussetzung, dass die Varianz klein genug ist, konvergiert $\sigma_n \to \sigma$ und es ist daher naheliegend diese Folge zu betrachten. Die σ_n können exakt berechnet werden, jedoch steigt der Rechenaufwand in n exponentiell. Daher wurde für $n = 1..30$ gewählt und berechnet. Dies ergab folgendes Ergebnis:

n	$\mathbb{E}$	$\mathbb{V}$
0	1.000000	0.000000
1	1.000000	0.000000
2	1.000000	0.000000
3	0.939496	0.060582
4	0.998942	0.028334
5	0.960099	0.044305
6	0.970801	0.034007
7	0.967989	0.033061
8	0.976730	0.025736
9	0.975891	0.025195
10	0.980931	0.021218
11	0.980675	0.020442
12	0.983747	0.017706
13	0.986253	0.015941
14	0.987452	0.014706

n	$\mathbb{E}$	$\mathbb{V}$
15	0.988273	0.013869
16	0.989611	0.012736
17	0.990981	0.011733
18	0.991752	0.011061
19	0.992550	0.010432
20	0.993315	0.009828
21	0.994088	0.009256
22	0.994687	0.008800
23	0.995263	0.008373
24	0.995800	0.007979
25	0.996318	0.007610
26	0.996775	0.007285
27	0.997201	0.006985
28	0.997601	0.006707
29	0.997979	0.006447

Tabelle 8: Erwartungswerte und Varianzen

Um eine Aussage über den Verlauf der Erwartungswerte und der Varianz treffen zu können ist $n = 30$ natürlich zu klein, doch es lässt sich eine gewisse Tendenz feststellen. So sieht man zum Beispiel dass die Varianz immer kleiner wird und somit im Einklang mit den Ergebnissen aus Abschnitt 3.3 stehen.

4 Aufgabe 4

(C. Pfeiler)

4.1 Erste Beobachtungen

4.1.1 Die Aufgabenbestellung

Berechnen Sie den Wert des Integrals:

$$\int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(1/x)}\right)}\right) dx\,. \tag{20}$$

Mit dieser Formulierung des Problems wollen wir uns gar nicht lange auseinandersetzen. Wir betrachten anstatt dessen ein äquivalentes Problem, welches wir durch die Variablensubstitution $y = 1/x$ erhalten:

$$\int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(1/x)}\right)}\right) dy \quad = \quad \int_1^\infty \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(y)}\right)}\right) \cdot \frac{1}{y^2}\, dy\,.$$

Gesucht ist nun also

$$\int_1^\infty f(x)g(x)\, dx\,, \qquad \text{mit} \quad f(x) := \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)$$
$$\text{und} \quad g(x) := \frac{1}{x^2}\,.$$

Dieses Lebesgue Integral existiert, da

$$\left|\int_1^\infty f(x)g(x)\, dx\right| \leq \int_1^\infty \frac{1}{x^2}\, dx = 1\,. \tag{21}$$

Desweiteren wollen wir gleich anmerken, dass wir anstatt dem Integrationsbereich $(1,\infty)$ auch nur den Bereich $(1,\infty)\setminus N$ betrachten können, solange N eine Lebesgue Nullmenge ist. Dies ist insbesondere der Fall, wenn N abzählbar ist. Mehr dazu später in den Abschnitten 4.2.2 und 4.2.3.

4.1.2 Besprechung

Zunächst sollten wir uns klarmachen, welche Schwierigkeiten dieses Integral mit sich bringt. In vielen Bereichen oszilliert die Funktion f stark, siehe Abbildungen 3, 4 und 5. Beispielsweise hat f in jedem Intervall der Länge 2π abzählbar unendlich viele wesentliche Singularitäten. Für jede dieser Singularitäten gibt es eine Folge von Nullstellen von f, die gegen die Singularität konvergiert. Man kann sogar eine nicht triviale Folge von Singularitäten finden, die wieder gegen eine Singularität konvergiert. Es ist also keine triviale Aufgabe, eine sinnvolle Quadraturformel für fg zu finden, da gilt:

(i). Der Integrationsbereich ist unbeschränkt.

(ii). Es gibt unendlich viele Unstetigkeitsstellen im Integrationsbereich.

(iii). Um diese Unstetigkeitstellen oszilliert f.

Die Darstellung liefert uns aber auch einiges, das wir eventuell verwenden können:

(i). g ist eine auf unserem Integrationsbereich glatte, rationale Funktion.

(ii). f ist 2π-periodisch.

(iii). Die Nullstellen von fg lassen sich explizit berechnen. Das gibt uns potenziell sinnvolle Intervalle um Quadraturformeln anzuwenden.

(iv). Wir werden in Abschnitt 4.3.1 sehen, dass uns die Oszillationseigenschaft von f Fehlerschranken liefert.

4.2 Ein bisschen Theorie

4.2.1 Umschreiben des Problems auf ein beschränktes Intervall

Da die folgende Rechnung etwas technisch wird, nehmen wir das Ergebnis vorweg. Es gilt

$$A := \int_{\pi/2}^{\infty} f(x)g(x)\,dx \approx \frac{1}{4\pi^2} \sum_{\substack{I=(a,b)\\ \in \mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{j=1}^{n} \omega_j f\left(x_j^{(a,b)}\right) \Psi\left(x_j^{(a,b)}\right). \tag{22}$$

mit Ψ wie in (26). Es gilt

$$\int_1^{\infty} f(x)g(x)\,dx = \int_1^{\pi/2} f(x)g(x)\,dx + \int_{\pi/2}^{\infty} f(x)g(x)\,dx$$

Das zweite, unbeschränkte Integral werden wir nun genauer untersuchen:

$$\begin{aligned} A &:= \int_{\pi/2}^{\infty} f(x)g(x)\,dx \\ &= \sum_{m=0}^{\infty} \int_{\pi/2+2m\pi}^{\pi/2+2(m+1)\pi} f(x)g(x)\,dx \\ &= \sum_{m=0}^{\infty} \sum_{c=1}^{4} \int_{I_{m,c}} f(x)g(x)\,dx \end{aligned}$$

Mit den Intervallen

$$I_{m,c} := \left(c\frac{\pi}{2} + 2m\pi, (c+1)\frac{\pi}{2} + 2m\pi\right), \quad \text{für } m \in \mathbb{N}_0, c \in \{1,2,3,4\}. \tag{23}$$

Sei nun $\mathcal{I}_{0,1}$ eine Zerlegung von $I_{0,1} = \left(\frac{\pi}{2}, \pi\right)$ in offene Intervalle $(a,b) \subset \mathbb{R}$ mit

- Für $I, J \in \mathcal{I}_{0,1}$, $I \neq J$ gelte, dass der Schnitt $I \cap J$ leer ist.
- Die Vereinigung $\bigcup_{I \in \mathcal{I}_{0,1}} I$ sei eine dichte Teilmenge von $I_{0,1} = \left(\frac{\pi}{2}, \pi\right)$.

Dann seien

$$\begin{aligned} \mathcal{I}_{0,2} &:= \{(2\pi - b, 2\pi - a) \mid (a,b) \in \mathcal{I}_{0,1}\} \\ \mathcal{I}_{0,3} &:= \{(a+\pi, b+\pi) \mid (a,b) \in \mathcal{I}_{0,1}\} \\ \mathcal{I}_{0,4} &:= \{(3\pi - b, 3\pi - a) \mid (a,b) \in \mathcal{I}_{0,1}\} \\ \mathcal{I}_{m,c} &:= \{(a+2m\pi, b+2m\pi) \mid (a,b) \in \mathcal{I}_{0,c}\} \quad m \in \mathbb{N}, c \in \{1,2,3,4\} \end{aligned}$$

die entsprechenden Zerlegungen von $I_{m,c}$ in (23). $\mathcal{I}_{m,c}$ erfüllt dann

- für $I, J \in \mathcal{I}_{m,c}$, $I \neq J$ gilt, dass der Schnitt $I \cap J$ leer ist,
- die Vereinigung $\bigcup_{I \in \mathcal{I}_{m,c}} I$ ist eine dichte Teilmenge von $I_{m,c}$,

für alle $m \in \mathbb{N}_0$, $j = 1, 2, 3, 4$. Für $\mathcal{I}_{m,c}$ folgt diese Aussage sofort aus der Forderung, dass die entsprechenden Eigenschaften für $\mathcal{I}_{0,1}$ gelten.

Nun wissen wir, wie wir durch eine Zerlegung $\mathcal{I}_{0,1}$ von $I_{0,1}$, die entsprechende Zerlegung $\mathcal{I}_{m,c}$ von $I_{m,c}$, und damit auch die entsprechende Zerlegung $\bigcup_{m,c} \mathcal{I}_{m,c}$ des unbeschränkten Intervalles $\left(\frac{\pi}{2}, \infty\right)$ darstellen können. Wir rechnen weiter:

$$\begin{aligned} A &= \sum_{c=1}^{4} \sum_{m=0}^{\infty} \int_{I_{m,c}} f(x)g(x)\,dx \\ &= \sum_{\substack{I=(a,b) \\ \in \mathcal{I}_{0,1}}} \sum_{m=0}^{\infty} \left[\int_{a+2m\pi}^{b+2m\pi} f(x)g(x)\,dx + \int_{2\pi-b+2m\pi}^{2\pi-a+2m\pi} f(x)g(x)\,dx \right. \\ &\quad \left. + \int_{a+\pi+2m\pi}^{b+\pi+2m\pi} f(x)g(x)\,dx + \int_{3\pi-b+2m\pi}^{3\pi-a+2m\pi} f(x)g(x)\,dx \right] \end{aligned}$$

Für ein Intervall $I = (a, b) \subset \mathbb{R}$ definiere die affine Transformation

$$\begin{aligned} \Phi_{(a,b)} : (-1, 1) &\to (a, b) \\ t &\mapsto \frac{1}{2}(a + b + t(b - a)). \end{aligned}$$

Es gilt $\Phi'_{(a,b)} \equiv \frac{b-a}{2}$. Nun verwenden wir die Transformationen $\Phi_{(a+2m\pi,b+2m\pi)}$, $\Phi_{(2\pi-b+2m\pi,2\pi-a+2m\pi)}$, $\Phi_{(a+\pi+2m\pi,b+\pi+2m\pi)}$ und $\Phi_{(3\pi-b+2m\pi,3\pi-a+2m\pi)}$:

$$\begin{aligned} A &= \sum_{\substack{I=(a,b) \\ \in \mathcal{I}_{0,1}}} \sum_{m=0}^{\infty} \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) g\left(\frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) \\ &\quad + f\left(2\pi - \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) g\left(2\pi - \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) \\ &\quad + f\left(\pi + \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) g\left(\pi + \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) \\ &\quad + f\left(3\pi - \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) g\left(3\pi - \frac{1}{2}(a+b) + 2m\pi + \frac{t}{2}(b-a)\right) dt \end{aligned}$$

Sei nun Q_n eine Quadraturformel der Länge n auf dem Referenzintervall $(-1, 1)$, deren Stützstellen x_j und Gewichte ω_j symmetrisch um den Nullpunkt gewählt sind, dh. die

$$\begin{aligned} x_j &= -x_{n+1-j}\,, \qquad j = 1, \ldots, n \\ \omega_j &= \omega_{n+1-j}\,, \qquad j = 1, \ldots, n \end{aligned}$$

erfüllen. Ein Beispiel dafür ist die Gauss-Legendre-Quadraturformel der Länge n, diese wurde in der Lehrveranstaltung Einführung in Scientific Computing besprochen.

Nun wollen wir solch eine Quadraturformel anwenden:

$$\begin{aligned}
A \approx \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{m=0}^{\infty}\sum_{j=1}^{n} \Bigg[& \omega_j f\left(\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) g\left(\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) \\
& +\omega_{n+1-j} f\left(2\pi-\frac{1}{2}(a+b)+2m\pi+\frac{x_{n+1-j}}{2}(b-a)\right) g\left(2\pi-\frac{1}{2}(a+b)+2m\pi+\frac{x_{n+1-j}}{2}(b-a)\right) \\
& +\omega_j f\left(\pi+\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) g\left(\pi+\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) \\
& +\omega_{n+1-j} f\left(3\pi-\frac{1}{2}(a+b)+2m\pi+\frac{x_{n+1-j}}{2}(b-a)\right) g\left(3\pi-\frac{1}{2}(a+b)+2m\pi+\frac{x_{n+1-j}}{2}(b-a)\right)\Bigg]
\end{aligned}$$

Aus den Symmetrieeigenschaften der von uns gewählten Quadraturformel folgt:

$$\begin{aligned}
A \approx \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{m=0}^{\infty}\sum_{j=1}^{n} \omega_j \Bigg[& f\left(\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) g\left(\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) \\
& +f\left(2\pi-\frac{1}{2}(a+b)+2m\pi-\frac{x_j}{2}(b-a)\right) g\left(2\pi-\frac{1}{2}(a+b)+2m\pi-\frac{x_j}{2}(b-a)\right) \\
& +f\left(\pi+\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) g\left(\pi+\frac{1}{2}(a+b)+2m\pi+\frac{x_j}{2}(b-a)\right) \\
& +f\left(3\pi-\frac{1}{2}(a+b)+2m\pi-\frac{x_j}{2}(b-a)\right) g\left(3\pi-\frac{1}{2}(a+b)+2m\pi-\frac{x_j}{2}(b-a)\right)\Bigg]
\end{aligned}$$

Wir nutzen die Eigenschaften $f(x) = f(x+2m\pi)$, $f(x) = -f(-x)$ und $f(x+\pi) = -f(x)$, welche f von der Sinusfunktion erbt, aus und definieren zur kompakteren Notation $x_j^{(a,b)} := \frac{1}{2}(a+b)+\frac{x_j}{2}(b-a)$. Damit gilt:

$$\begin{aligned}
A \approx \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{m=0}^{\infty}\sum_{j=1}^{n} \omega_j f\left(x_j^{(a,b)}\right) \Big[& g\left(x_j^{(a,b)}+2m\pi\right) - g\left(2\pi-x_j^{(a,b)}+2m\pi\right) \\
& -g\left(\pi+x_j^{(a,b)}+2m\pi\right) + g\left(3\pi-x_j^{(a,b)}+2m\pi\right)\Big]
\end{aligned}$$

Wir vertauschen die Summen und verwenden $\frac{1}{4\pi^2}\psi_1(\frac{x}{2\pi}) = \sum\limits_{m=0}^{\infty} \frac{1}{(x+2m\pi)^2}$, wobei $\psi_1(\cdot)$ die erste Ableitung der Polygammafunktion, auch Trigammafunktion genannt, bezeichnet:

$$\psi_1(z) := \sum_{m=0}^{\infty} \frac{1}{(z+m)^2} \tag{24}$$

Obiger Zusammenhang folgt sofort durch eine kurze Rechnung. Wir setzen ein:

$$\begin{aligned} A &\approx \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{j=1}^{n} \omega_j f\left(x_j^{(a,b)}\right) \left[\sum_{m=0}^{\infty} \frac{1}{\left(x_j^{(a,b)} + 2m\pi\right)^2} - \sum_{m=0}^{\infty} \frac{1}{\left(2\pi - x_j^{(a,b)} + 2m\pi\right)^2} \right. \\ &\quad \left. - \sum_{m=0}^{\infty} \frac{1}{\left(x_j^{(a,b)} + \pi + 2m\pi\right)^2} + \sum_{m=0}^{\infty} \frac{1}{\left(3\pi - x_j^{(a,b)} + 2m\pi\right)^2} \right] \\ &= \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \frac{1}{4\pi^2} \sum_{j=1}^{n} \omega_j f\left(x_j^{(a,b)}\right) \left[\psi_1\left(\frac{x_j^{(a,b)}}{2\pi}\right) - \psi_1\left(\frac{2\pi - x_j^{(a,b)}}{2\pi}\right) \right. \\ &\quad \left. - \psi_1\left(\frac{\pi + x_j^{(a,b)}}{2\pi}\right) + \psi_1\left(\frac{3\pi - x_j^{(a,b)}}{2\pi}\right) \right] \end{aligned}$$

und erhalten schlussendlich

$$A \approx \frac{1}{4\pi^2} \sum_{\substack{I=(a,b)\\ \in\mathcal{I}_{0,1}}} \frac{b-a}{2} \sum_{j=1}^{n} \omega_j f\left(x_j^{(a,b)}\right) \Psi\left(x_j^{(a,b)}\right). \tag{25}$$

mit

$$\Psi(x) := \left[\psi_1\left(\frac{x}{2\pi}\right) - \psi_1\left(\frac{2\pi - x}{2\pi}\right) - \psi_1\left(\frac{\pi + x}{2\pi}\right) + \psi_1\left(\frac{3\pi - x}{2\pi}\right) \right] \tag{26}$$

für $x \in \left(\frac{\pi}{2}, \pi\right)$. Es gilt in den Umformungen stets Gleichheit, bloß bei der Approximation der Integrale $\int_{-1}^{1} h(x)\,dx \approx Q_n h$ treten Approximationsfehler auf. Wir merken an, dass

$$A = \frac{1}{4\pi^2} \int_{\pi/2}^{\pi} f(x)\Psi(x)\,dx$$

gilt. Beweisskizze: Dies kann man wie eben gezeigt nachrechnen, wenn man die Zerlegung $\mathcal{I}_{0,1}$ so wählt, dass f auf allen $(a,b) \in \mathcal{I}_{0,1}$ stetig ist. Entsprechend gilt dies auch für alle Intervalle in $\mathcal{I}_{m,c}$. Die Konvergenz der Quadraturformeln $Q_n h \to \int h\,dx$ für $n \to \infty$ für stetige Funktionen h schließt den Beweis. Dass man so eine Zerlegung $\mathcal{I}_{0,1}$ tatsächlich wählen kann, sehen wir in Abschnitt 4.2.3.
Technisches Detail zum Beweis: Will man die Behauptung rigoros durchrechnen, könnte man um jedes S_ℓ eine Umgebung U_ℓ mit $|U_\ell| < 2^{-\ell}\varepsilon$, $S_\ell \in U_\ell$ herausnehmen. Für $\varepsilon \to 0$ folgt die Behauptung.

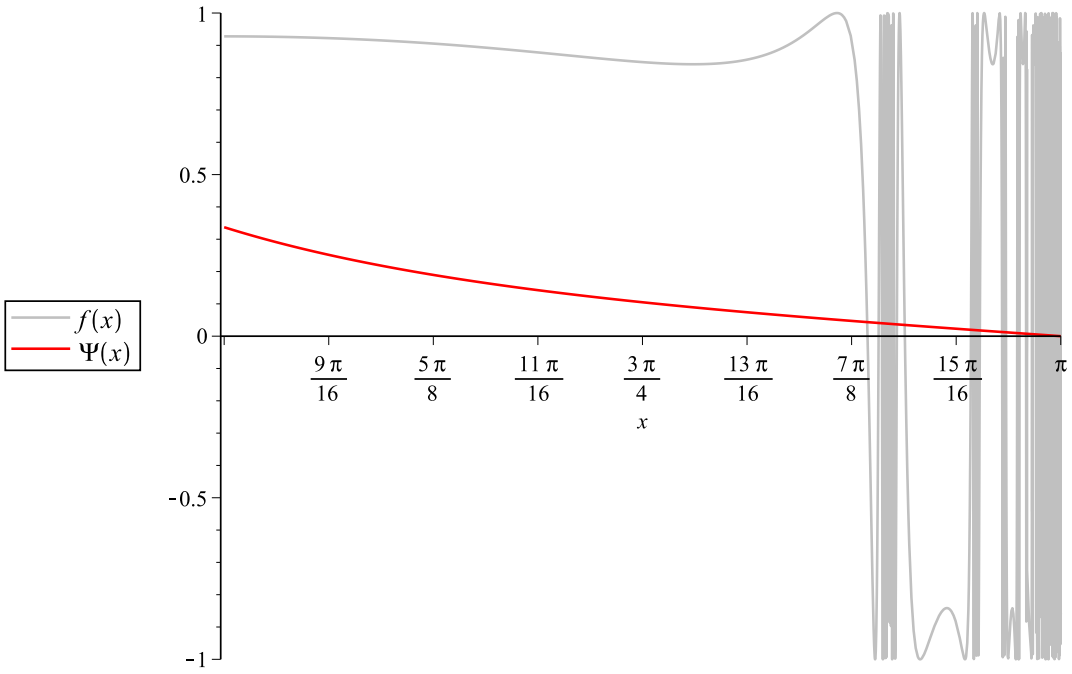


Abbildung 3: Die Funktion f wird mit Ψ gewichtet, siehe Abbildung 4.

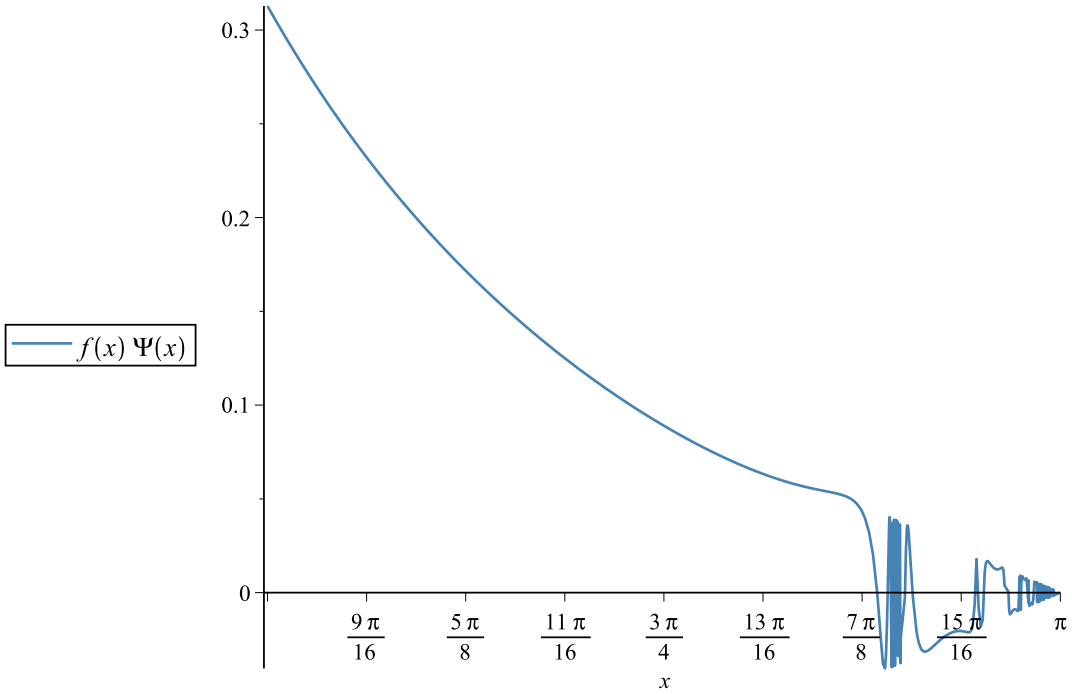


Abbildung 4: Die zu integrierende Funktion $f\Psi$ im Intervall $\left(\frac{\pi}{2}, \pi\right)$.

Wir haben nun das Integral von fg über das unbeschränkte Intervall $\left(\frac{\pi}{2}, \infty\right)$ auf die Integration von $f\Psi$ auf dem beschränkten Intervall $\left(\frac{\pi}{2}, \pi\right)$ umgeschrieben. Der nächste Schritt ist es nun eine für Quadraturformeln Q_n brauchbare Zerlegung $\mathcal{I}_{0,1}$ von $\left(\frac{\pi}{2}, \pi\right)$ zu finden.
Eine kanonische und für Q_n günstige Wahl für die Zerlegung des Intervalls $\left(\frac{\pi}{2}, \pi\right)$ in Teilintervalle

ist die Unterteilung an den Nullstellen des Integranden. Das sind genau die Nullstellen von f in diesem Intervall. Diese Unterteilung werden wir auch wählen.

4.2.2 Die Nullstellen von f

Wenn nicht explizit anders angegeben, betrachten wir von nun f an nur noch auf dem Intervall $\left(\frac{\pi}{2}, \pi\right)$.
Die formale Darstellung der Nullstellen von f erhält man direkt durch einfaches Umformen:

$$\begin{aligned} \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right) &= 0 \\ &\Leftrightarrow \\ x &= \sin^{-1}\left(\frac{1}{\sin^{-1}\left(\frac{1}{\sin^{-1}(0)}\right)}\right) \end{aligned}$$

Wir wollen nun aber eine Darstellung mit der wir arbeiten können. Sei dazu

$$\begin{aligned} \arcsin : [-1,1] &\to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ y &\mapsto \arcsin(y) \end{aligned}$$

die eindeutige, bijektive Umkehrabbildung von $\sin : \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \to [-1,1]$. Dann sind die Nullstellen von f gegeben durch

$$\begin{aligned} N : \mathbb{Z}^\times \times \mathbb{N}^\times &\to \left[\frac{\pi}{2}, \pi\right] \\ (k,\ell) &\mapsto N_{k,\ell} := \pi - \arcsin\left(\frac{1}{(-1)^\ell \arcsin\left(\frac{1}{k\pi}\right) + \ell\pi}\right). \end{aligned}$$

Diese Darstellung liefert sogar noch mehr: Die Singularitäten von f sind gegeben durch

$$\begin{aligned} S_\ell &:= \lim_{|k|\to\infty} N_{k,\ell} = \pi - \arcsin\left(\frac{1}{\ell\pi}\right), \quad \ell \in \mathbb{N}^\times. \\ S_\infty &:= \lim_{\ell\to\infty} S_\ell = \pi\,. \end{aligned}$$

Wir nennen S_ℓ die ℓ-te Singularität von f. Diese Bezeichnung macht Sinn, da $\ell \mapsto S_\ell$ monoton steigend ist. Desweiteren nennen wir $N_{k,\ell}$ die k-te Nullstelle bei S_ℓ, $k \in \mathbb{Z}^\times$, $\ell \in \mathbb{N}^\times$. Auch diese Bezeichnung macht Sinn, da für ungerade ℓ die Folge $(N_{k,\ell})_{k>0}$ monoton steigend in k mit Grenzwert S_ℓ, und die Folge $(N_{-k,\ell})_{k>0}$ monoton fallend in k mit demselben Grenzwert ist. Umgekehrt ist für gerade ℓ die Folge $(N_{k,\ell})_{k>0}$ monoton fallend in k mit Grenzwert S_ℓ, und die Folge $(N_{-k,\ell})_{k>0}$ monoton steigend in k mit Grenzwert S_ℓ. Zuletzt gilt nun auch noch $N_{k_1,\ell_1} > N_{k_2,\ell_2}$ für $\ell_1 > \ell_2$, ganz unabhängig von k_1, k_2. Diese Aussagen kann man leicht nachrechnen, wir wollen sie in einem Plot veranschaulichen:

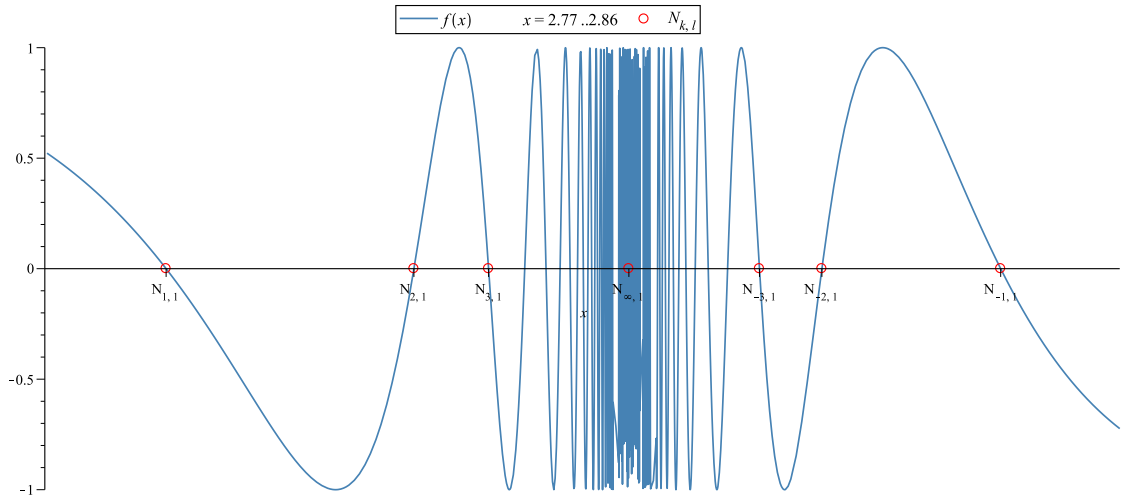


Abbildung 5: Das Verhalten von f im Bereich um S_1 und die ersten drei Nullstellen auf beiden Seiten der Singularität.

4.2.3 Die Zerlegung $\mathcal{I}_{0,1}$

Mit Hilfe dieser Nullstellenfunktion wollen wir nun die Unterteilung $\mathcal{I}_{0,1}$ so wählen, dass f in einem Intervall $I \in \mathcal{I}_{0,1}$ das Vorzeichen nicht wechselt. Definiere

$$\begin{aligned}
\mathcal{I}_{\ell,+} &:= \left\{(N_{k,\ell}, N_{k+1,\ell}) \,\middle|\, k \in \mathbb{N}^\times\right\} \quad \text{für } \ell \in \mathbb{N}^\times \text{ ungerade,} \\
\mathcal{I}_{\ell,-} &:= \left\{(N_{-k-1,\ell}, N_{-k,\ell}) \,\middle|\, k \in \mathbb{N}^\times\right\} \quad \text{für } \ell \in \mathbb{N}^\times \text{ ungerade,} \\
\mathcal{I}_{\ell,+} &:= \left\{(N_{-k,\ell}, N_{-k-1,\ell}) \,\middle|\, k \in \mathbb{N}^\times\right\} \quad \text{für } \ell \in \mathbb{N}^\times \text{ gerade,} \\
\mathcal{I}_{\ell,-} &:= \left\{(N_{k+1,\ell}, N_{k,\ell}) \,\middle|\, k \in \mathbb{N}^\times\right\} \quad \text{für } \ell \in \mathbb{N}^\times \text{ gerade,} \\
\mathcal{I}_0 &:= \bigcup_{\substack{\ell \in \mathbb{N}^\times \\ \sigma \in \{\pm\}}} \mathcal{I}_{\ell,\sigma}, \\
\mathcal{I}_{00} &:= \left\{(N_{(-1)^\ell,\ell}, N_{(-1)^\ell,\ell+1}) \,\middle|\, \ell \in \mathbb{N}^\times\right\}, \\
\mathcal{I}_{0,1} &:= \left\{\left(\frac{\pi}{2}, N_{1,1}\right)\right\} \cup \mathcal{I}_0 \cup \mathcal{I}_{00}\,.
\end{aligned} \tag{27}$$

$\mathcal{I}_{\ell,+}$ sind genau die Intervalle zwischen den zur Singularität S_ℓ gehörenden Nullstellen $N_{k,\ell} < S_\ell$, ebenso sind $\mathcal{I}_{\ell,-}$ genau die Intervalle zwischen den zur Singularität S_ℓ gehörenden Nullstellen $N_{k,\ell} > S_\ell$. $\mathcal{I}_{00}$ sind die Intervalle von der größten zu S_ℓ, bis zur kleinsten zu $S_{\ell+1}$ gehörenden Nullstelle, $\ell \in \mathbb{N}^\times$.
Definiere außerdem zur kompakteren Notation in den nächsten Abschnitten $Q^n_{k,\ell,+}$, $Q^n_{k,\ell,-}$, Q^n_ℓ als die n-Punkt Gauss-Legendre Quadratur Approximation an das Integral über das entsprechende Intervall aus $\mathcal{I}_{\ell,+}$, $\mathcal{I}_{\ell,-}$, $\mathcal{I}_{00}$ mit k, ℓ wie in (27).

4.3 Numerische Berechnung des beschränkten Integrals

Aus der geleisteten Vorarbeit lässt sich schon vermuten, welche Strategie wir zur numerischen Berechnung des Integrals verfolgen werden: Verwende die Approximation (25) mit der in Kapitel 4.2.3 konstruierten Zerlegung $\mathcal{I}_{0,1}$. Der Einfachheit halber, wollen wir annehmen, dass Q_n mit n fest "ausreichend" exakt ist. Wir werden in 4.4 sehen, dass wir stets so ein Q_n finden. Es gibt es in dieser Zerlegung unendlich viele Intervalle, die wir nicht alle betrachten können, deshalb müssen wir uns darüber Gedanken machen, welche endliche Teilmenge $\mathcal{I} \subset \mathcal{I}_{0,1}$ wir

betrachten. In Abbildung 4 sieht man, dass das Intervall $\left(\frac{\pi}{2}, N_{1,1}\right)$ sicher den größten Beitrag zur Lösung hat. Dieses Intervall wird immer ein Element von $\mathcal{I}$ sein, weshalb wir es im Folgenden nicht mehr explizit erwähnen werden. Durch die Visualisierung ist die Entscheidung für dieses Intervall einfach, aber wie könnten wir generell entscheiden, welche Integrale in $\mathcal{I}$ sein sollen? Naheliegende Möglichkeiten dafür sind:

(i). Betrachte nur die N größten Intervalle aus $\mathcal{I}_{0,1}$. Diese Idee macht Sinn, da zu erwarten ist, dass die Integrale über diese Intervalle auch die größten Anteile am Ergebnis haben werden. Diese Strategie wird für $N \to \infty$ auch zum Ziel führen, jedoch ist es keine triviale Aufgabe, die größten N Intervalle auszumachen. Es wird auch schwer eine andere, als die triviale Abschätzung $|\bigcup \mathcal{I}_{0,1} \setminus \bigcup \mathcal{I}|$ an den Fehler zu finden.

(ii). Berechne zu gegebenen L und K die Summe $\sum_{\ell=1}^{L} Q_\ell^n + \sum_{\ell=1}^{L} \sum_{k=1}^{K} \sum_{\sigma\in\{\pm\}} Q_{k,\ell,\sigma}^n$. Auch diese Strategie wird für $K, L \to \infty$ zum Ziel führen. Da diese Methode auch sehr leicht zu implementieren ist, war das unser erster Zugang. Man erhält durch die Oszillation von f ähnlich wie in (iii) eine bessere Abschätzung an den Fehler als in (i). Leider werden bei dieser Strategie die zur Verfügung stehenden Ressourcen nicht gerade optimal genutzt.

(iii). Es wird versucht zu einer vorgegebenen Toleranz $\tau > 0$ eine möglichst kleine Teilmenge $\mathcal{I} \subset \mathcal{I}_{0,1}$ zu finden, sodass

$$\left| \sum_{I\in\mathcal{I}_{0,1}\setminus\mathcal{I}} \int_I f(x)\Psi(x)\,dx \right| \leq \tau \tag{28}$$

gilt. Mit dem Finden dieser Menge $\mathcal{I}$ werden wir uns in 4.3.1 und 4.3.2 beschäftigen. Der Algorithmus wird nach der Idee aus (ii) arbeiten, nur ist L variabel, und zu jedem ℓ wird es ein anderes K geben.

Wir werden Strategie Nummer (iii) verfolgen. Ideen gute Wahlen für L und $K(\ell)$, $\ell = 1..L$ zu treffen, liefern die Überlegungen in Abschnitt 4.3.1.

4.3.1 Abschätzungen des Fehlers

Betrachten wir zunächst $\mathcal{I}_{00}$ mit den Elementen $(N_{(-1)^\ell,\ell}, N_{(-1)^\ell,\ell+1}) \in \mathcal{I}_{00}$, $\ell \in \mathbb{N}^\times$ aus Abschnitt 4.2.3. Die Intervalle werden mit wachsendem ℓ stets kleiner, die Funktion f ist abwechselnd positiv oder negativ, und sieht transformiert auf das Referenzintervall $(-1, 1)$ für verschiedene ℓ fast gleich aus, modulo Vorzeichen und Skalierung, siehe Abbildung 6.

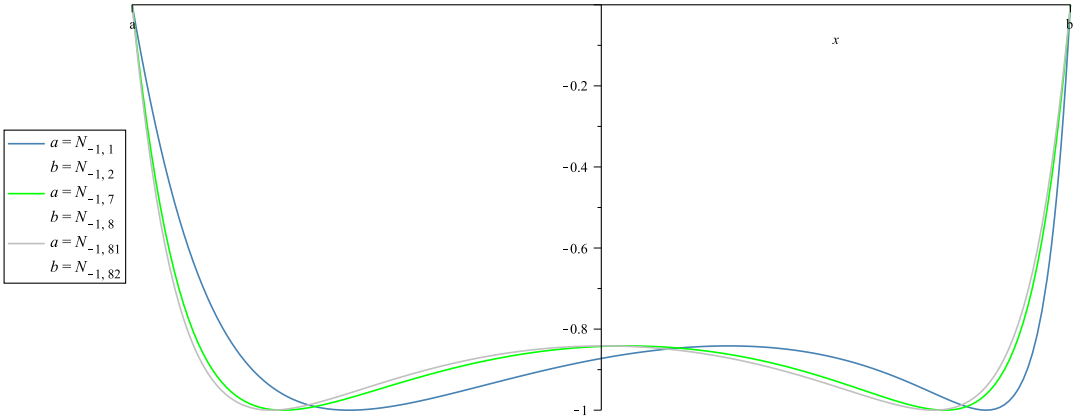


Abbildung 6: Man beachte die unterschiedliche Skalierung. Das skalierte f ändert sich für größere ℓ nicht mehr gravierend. Nur das Vorzeichen ist für ℓ und $\ell + 1$ verschieden.

Die Integrale Q_ℓ^n hängen also hauptsächlich von der Länge des entsprechenden Intervalls ab. Demnach kann man vermuten, dass die Folge $\ell \mapsto Q_\ell^n$ eine alternierende Nullfolge ist. Tabelle 9 bestätigt diese Vermutung. Insgesamt erhält man die Abschätzung

$$\left| \sum_{\ell=1}^{\infty} Q_\ell^n - \sum_{\ell=1}^{L} Q_\ell^n \right| = \left| \sum_{\ell=L+1}^{\infty} Q_\ell^n \right| \leq \left| Q_{L+1}^n \right| . \tag{29}$$

ℓ	Q_ℓ^n
1	-0.002874794991848
2	0.000550385939843
3	-0.000194439912868
4	0.000090344606027
5	-0.000049171737708
6	0.000029682236461
7	-0.000019278457891
8	0.000013223457296
9	-0.000009461777164
10	0.000007002324240

ℓ	Q_ℓ^n
101	-0.000000007725494
102	0.000000007501575
103	-0.000000007286227
104	0.000000007079044
105	-0.000000006879641
106	0.000000006687658
107	-0.000000006502753
108	0.000000006324601
109	-0.000000006152898
110	0.000000005987355

ℓ	Q_ℓ^n
1001	-0.000000000008042
1002	0.000000000008018
1003	-0.000000000007994
1004	0.000000000007970
1005	-0.000000000007946
1006	0.000000000007923
1007	-0.000000000007899
1008	0.000000000007876
1009	-0.000000000007852
1010	0.000000000007829

Tabelle 9: Die Vermutung, dass Q_ℓ^n eine alternierende Nullfolge ist, wird bestätigt.

Ganz ähnlich erhält man eine Abschätzung für den Approximationsfehler im Bereich um eine Singularität S_ℓ. Sei $\ell \in \mathbb{N}^\times$ fest, und betrachte die Folge $k \mapsto Q_{k,\ell,+}^n + Q_{k,\ell,-}^n$. Aufgrund der Oszillation von f liegt die Vermutung nahe, dass auch diese Folge eine alternierende Nullfolge ist. Tabelle 10 bestätigt diese Vermutung. Hier erhält man für festes ℓ die Abschätzung

$$\left| \sum_{k=K+1}^{\infty} Q_{k,\ell,+}^n + Q_{k,\ell,-}^n \right| \leq \left| Q_{K+1,\ell,+}^n + Q_{K+1,\ell,-}^n \right| \tag{30}$$

k	$Q^n_{k,1,+} + Q^n_{k,1,-}$
1	-0.000194650987715
2	0.000036953753822
3	-0.000013036908814
4	0.000006054572745
5	-0.000003294578053
6	0.000001988508652
7	-0.000001291429055
8	0.000000885771837
9	-0.000000633774459
10	0.000000469022336

k	$Q^n_{k,1,+} + Q^n_{k,1,-}$
101	-0.000000000517405
102	0.000000000502408
103	-0.000000000487986
104	0.000000000474110
105	-0.000000000460755
106	0.000000000447897
107	-0.000000000435514
108	0.000000000423582
109	-0.000000000412082
110	0.000000000400995

k	$Q^n_{k,100,+} + Q^n_{k,100,-}$
1	0.000000000001625
2	-0.000000000000313
3	0.000000000000111
4	-0.000000000000052
5	0.000000000000028
6	-0.000000000000017
7	0.000000000000011
8	-0.000000000000008
9	0.000000000000005
10	-0.000000000000004

Tabelle 10: Die Vermutung, dass $Q^n_{k,\ell,+} + Q^n_{k,\ell,-}$ für festes ℓ eine alternierende Nullfolge in k ist, wird bestätigt. Hier zu sehen an den Beispielen $\ell = 1, 100$.

Nun ist

$$(k, \ell) \mapsto Q^n_{k,\ell,+} + Q^n_{k,\ell,-}$$

aber nicht nur eine alternierende Nullfolge in k, sondern auch in ℓ. Das liefert uns die Abschätzung

$$\left| \sum_{\ell=L+1}^{\infty} \sum_{k=1}^{\infty} Q^n_{k,\ell,+} + Q^n_{k,\ell,-} \right| \leq \left| Q^n_{1,L+1,+} + Q^n_{1,L+1,-} \right| . \tag{31}$$

Da uns nun brauchbare Abschätzungen an die Approximationsfehler zur Verfügung stehen, können wir einen adaptiven Algorithmus formulieren.

4.3.2 Adaptiver Algorithmus

Die Beobachtungen aus dem vorangegangenen Abschnitt 4.3.1 werden wir nun verwenden um einen adaptiven Algorithmus zu formulieren. Es soll zu vorgegebener Toleranz $\tau > 0$ eine Approximation berechnet werden, welche sich von dem tatsächlichen Ergebnis um weniger als τ unterscheidet.

```
I = a4_adaptive_parallel(τ)
   τ1 = τ/3; τ2 = τ/3;

// Preprocessing
   for(L1 = L1_start; |Q^{n1}_{1,L1,+} + Q^{n1}_{1,L1,-}| > τ1; L1 += L1_step); // see (31)

   for (L2 = L2_start; |Q^{n2}_{L2}| > τ2; L2 += L2_step); // see (29)

   τ3 = τ/L1/3;
   I = 0;

// Evaluate integrals of fΨ over I ∩ I0
   #pragma omp parallel for schedule(dynamic)
   for(ℓ = 1; ℓ < L1+1; ++ℓ)
      for(k = 1; |Q^{n1}_{k,ℓ,+} + Q^{n1}_{k,ℓ,-}| > τ3; ++k) // see (30)
         I += Q^{n1}_{k,ℓ,+} + Q^{n1}_{k,ℓ,-};
```

```
      end for;
   end for;

// Evaluate integrals of fΨ over I ∩ I_00
   #pragma omp parallel for schedule(static)
   for(ℓ = 1; ℓ < L2+1; ++ℓ)
      I += Q_ℓ^{n_2};
   end for;

   I += ∫_{π/2}^{N_{1,1}} fΨ dx // evaluate using Q_{n_2}
   I += ∫_1^{π/2} fg dx // evaluate using Q_{n_2}

   return I;
end.
```

Wenn in der Iteration eine Toleranz unterschritten wird, wird noch die Hälfte des zuletzt berechneten Wertes zu `I` addiert. Das ist genau der Mittelpunkt des Intervalls in dem die exakte Lösung dieser Iteration liegen muss. Wir können dadurch die Abschätzung des Approximationsfehlers halbieren. Für den absoluten Fehler `err` der Approximation gilt demnach mit (29), (30) und (31)

$$\texttt{err} < \frac{\tau_1 + \tau_2 + \texttt{L1}\tau_3}{2} \leq \frac{\tau}{2}\,.$$

4.4 Numerische Details

4.4.1 Software

Zunächst wurde MATLAB verwendet. Das bietet sich an, da einerseits die Quadraturformeln durch ausnutzen der vorhandenen optimierten Vektorarithmetik sehr effizient implementiert werden können. Anderseits kann die in der MAPLE-Toolbox vorhandene Trigammafunktion verwendet werden. Leider ist in der Studentenversion von MATLAB die Variable-Precision-Toolbox nicht enthalten, wodurch MATLAB für das Finden möglichst vieler Nachkommastellen, allein schon durch die verwendete `double`-Precision, unbrauchbar wird.
In MAPLE kann man die Rechengenauigkeit beliebig einstellen und die Trigammafunktion ist auch in beliebiger Genauigkeit vorhanden. Also war eine Implementierung in MAPLE der nächste Schritt. So weit so gut, aber leider muss in dem Algorithmus aus Abschnitt 4.3.2 viel gerechnet werden. An diesem Punkt wird MAPLE unbrauchbar, es rechnet einfach zu langsam. Also was tun? Soll man sich einen neuen Algorithmus überlegen, in dem weniger gerechnet werden muss, und ist das überhaupt möglich? Außerdem wäre ja dann die bisherige mathematische Analyse umsonst gewesen. Man will die Trigammafunktion und beliebige Rechengenauigkeit wie in MAPLE, aber Geschwindigkeit wie in MATLAB.
Die Antwort lautet `C++` in Verbindung mit der Boost Multiprecision Library. Das ist eine `C++` Bibliothek die das Rechnen mit Gleitkommazahlen mit beliebig vielen Nachkommastellen unterstützt. Die einzige Beschränkung an die verfügbare Genauigkeit stellt die verwendete Hardware dar. Für höhere Genauigkeit als `double` sind die Rechnungen natürlich langsamer als die entsprechenden `double` Precision Berechnungen in MATLAB, aber sie sind gravierend schneller als in MAPLE. Die von uns benötigten Standardfunktionen sin() und arcsin() sind vorhanden. Bloß die Trigammafunktion steht leider nicht zur Verfügung. Der Kompromiss lautet also: Wir haben Geschwindigkeit und Multiple Precision Arithmetic, jedoch keine Trigammafunktion. Ab-

hilfe verschaffen wir uns, indem wir unsere eigene Trigammafunktion $\tilde{\psi_1}$ schreiben, siehe dazu Abschnitt 4.4.3.

4.4.2 Die Quatraturformel Q_n

Wie schon in Abschnitt 4.2.1 angemerkt, wollen wir die symmetrische n-Punkt Gauss-Legendre Quadraturformel auf dem Referenzintervall $(-1, 1)$ verwenden. Die Gewichte ω_j und die Stützstellen x_j erhält man durch Lösung eines leicht zu implementierenden Eigenwertproblems. Dieses Eigenwertproblem wurde mit der Hilfe von MAPLE gelöst.
Nun stellt sich noch die Frage, welches n wir verwenden wollen. In den Abbildungen 3, 5 und 6 haben wir gesehen, dass f zwischen benachbarten Nullstellen nur zwei verschiedene Formen annimmt. Wenn die Nullstellen zu verschiedenen Singularitäten gehören, sieht f aus wie in Abbildung 6, sonst wie die Peaks in 5. Tests in MAPLE haben zeigten, dass für Intervalle wie in Abbildung 6 128 Quadraturpunkte ausreichend sind, wenn wir das Integral auf 50 Stellen exakt auswerten wollen. Für Intervalle wie in Abbildung [Ref: To be made] reichen bereits 64 Quadraturpunkte für die selbe Genauigkeit. Wir wählen also Q_{n_1} und Q_{n_2} mit $n_1 = 64$ und $n_2 = 128$. Die Stützstellen und Gewichte wurden innerhalb weniger Minuten in MAPLE auf 50 Stellen exakt berechnet.

4.4.3 Die Trigammafunktion ψ_1

In der Boost Multiprecision Library findet sich bis zum heutigen Tage (noch) keine Implementierung der Trigammafunktion. Deshalb wurde eigenständig eine für unsere Zwecke brauchbare, und ausreichende Variante $\tilde{\psi_1}$ implementiert. In der Definition (26) von Ψ sehen wir, dass wir ψ_1 nur für x aus dem Intervall $(0.25, 1.25)$ zu approximieren brauchen. Wir werden $\tilde{\psi_1}$ in jedem Intervall $I \in \mathcal{I} \cap \mathcal{I}_0$ genau $4 \cdot 64 = 256$ mal, und in jedem Intervall $I \in \mathcal{I} \cap \mathcal{I}_{00}$ sogar 512 mal auswerten. Unsere Anforderungen an eine Approximation der Trigammafunktion $\tilde{\psi_1}$ lauten deshalb wie folgt:

(i). Für $x \in (0.25, 1.25) \subset \mathbb{R}$ stimme $\tilde{\psi_1}(x)$ auf 50 Nachkommastellen mit $\psi_1(x)$ überein.

(ii). Die Berechnung von $\tilde{\psi_1}(x)$ habe eine konstante Rechenzeit, unabhängig von $x \in (0.25, 1.25)$.

Die theoretische Grundlage zur Implementierung von $\tilde{\psi_1}$ bilden die Rekursionsformel [6]

$$\psi_1(z+1) = \psi_1(z) - \frac{1}{z^2} \tag{32}$$

und die asymptotische Berechnung von $\psi_1(z)$ durch die Bernoullizahlen B_{2k} [4]

$$\psi_1(z) \sim \frac{1}{z} + \frac{1}{2z^2} + \sum_{k=1}^{N} \frac{B_{2k}}{z^{2k+1}} \,. \tag{33}$$

Wir müssen aufpassen, (33) ist mit Vorsicht zu genießen: Die Reihe ist für kein z konvergent, wenn man $N \to \infty$ laufen lässt. Jedoch stellt sie für nicht zu groß gewählte N eine sehr gute Näherung dar. Je größer $|z|$ ist, umso größer kann N gewählt werden.
Mit (32) und (33) können wir $\tilde{\psi_1}$ sinnvoll definieren.

$$\tilde{\psi_1}(x) := \sum_{m=0}^{N_1} \frac{1}{(x+m)^2} + \frac{1}{x+N_1+1} + \frac{1}{2(x+N_1+1)^2} + \sum_{k=1}^{N_2} \frac{B_{2k}}{(x+N_1+1)^{(2k+1)}} \tag{34}$$

Tests zeigten, dass bereits die Wahl $N_1 = N_2 = 30$ die gewünschte Approximation liefert: Es gilt für beliebiges $x \in (0.25, 1.25)$ dann $|\psi_1(x) - \tilde{\psi_1}(x)| <$`1e-50`. Die Tests wurden mit

MAPLE durchgeführt, dort ist die Trigammafunktion standardmäßig vorhanden. Die Zahlen B_{2k}, $k = 1, ..., 30$ wurden innerhalb kürzester Zeit mit MAPLE auf 50 Nachkommastellen genau berechnet.

4.5 Resultate

Alle Resultate wurden mit dem Algorithmus aus Abschnitt 4.3.2 in `C++` unter Verwendung der Boost Multiprecision Library berechnet. Es wurden auf 50 Stellen exakte Gleitpunktzahlen verwendet. Quadraturpunkte, Gewichte und Bernoullizahlen wurden einmal im Vorhinein mit MAPLE berechnet, abgespeichert und zu Beginn der Laufzeit des Algorithmus importiert. Die Berechnung der besten Approximation mit vorgegebener Toleranz 10^{-16} dauerte auf dem Notebook ca. 6 Stunden. Die Berechnung jeder Zeile der Tabelle dauerte ungefähr 2.5 mal so lange wie die Zeile darüber.

τ	I	μ	$\tilde{\mu}$
1e-8	0.49621397161429681876	1.80e-09	6.03e-12
1e-9	0.49621397160236697160	2.91e-10	5.90e-12
1e-10	0.49621397160757040120	2.81e-11	6.99e-13
1e-11	0.49621397160810632996	4.15e-12	1.63e-13
1e-12	0.49621397160824946675	4.59e-13	2.02e-14
1e-13	0.49621397160826758700	4.60e-14	2.08e-15
1e-14	0.49621397160826944703	4.69e-15	2.16e-16
1e-15	0.49621397160826964210	4.87e-16	2.11e-17
1e-16	0.49621397160826966315	4.87e-17	-

Tabelle 11: Das Ergebnis des Algorithmus aus Abschnitt 4.3.2 zur vorgegebenen Toleranz τ. I bezeichnet das Ergebnis $I \approx \int_1^\infty f(x)g(x)\,dx$, μ ist eine zuverlässige Schätzung an den absoluten Fehler, die wir durch (29), (30) und (31) erhalten. $\tilde{\mu}$ ist der absolute Fehler, den wir erhalten, wenn wir unsere beste Approximation als Lösung des Problems betrachten.

Wir kommen zu dem Ergebnis

$$\int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(1/x)}\right)}\right) dx \approx \mathbf{0.4962139716082669}66315\ldots \tag{35}$$

Die fett gedruckten Stellen sind garantiert richtig. Das Verhalten von $\tilde{\mu}$ lässt jedoch vermuten, dass der tatsächliche Fehler kleiner $\mu/20$ ist, wodurch die Vermutung berechtigt ist, dass auch die Ziffern $\ldots 66$ richtig sind. Was ist der Grund dafür, dass die tatsächliche Approximation besser ist als es die Fehlerschranke vermuten lässt? Es liegt an dem Verhalten des Algorithmus, wenn eine Toleranz unterschritten wird: Anscheinend liefert uns die Strategie, immer den Intervallmittelpunkt des Intervalles in dem die Lösung liegen muss als weitere Approximation zu verwenden, wenn eine Toleranz unterschritten wird, ein bis zwei weitere Nachkommastellen.

5 Aufgabe 5

(P. Lederer)

Die letzte Aufgabe beschäftigte sich mit einer partiellen Differentialgleichung von der ein bestimmter Wert gesucht war. Bei der Numerik von solchen Gleichungen spielt vor allem die Wahl einer geeigneten Diskretisierung von Ort und Zeit eine zentrale Rolle.

5.1 Aufgabenstellung

Sei $u(x,t)$ die klassische Lösung von

$$\begin{aligned} u_t - \Delta u &= e^u \quad \text{in } \Omega = (0,1)^2,\ t > 0, \\ u(x,t) &= 0 \quad \text{auf } \partial\Omega,\ t > 0, \\ u(x,0) &= 1 \quad \text{in } \Omega. \end{aligned} \tag{36}$$

Man bestimme

$$t^* = \min\{t > 0 : \|u(t)\|_{L^\infty(\Omega)} = +\infty\}. \tag{37}$$

Gesucht ist also jener Zeitpunkt, bei dem die Lösung einen "blow-up" erzeugt.

5.2 Kurze Analyse und eine erste Simulation

Gleichung (36) beschreibt eine nichtlineare partielle Differentialgleichung zweiter Ordnung. Betrachtet man im ersten Schritt nur das stationäre Problem

$$-\Delta u = e^u,$$

so kann man Δu als Diffusionsterm und e^u als Reaktionterm interpretieren. Die zeitliche Entwicklung der Lösung wird also davon abhängen, ob der Laplace-Operator stärker als die Exponentialfunktion wirkt und sich somit eine stationäre Lösung einstellt. Um einen ersten Eindruck der zeitlichen Entwicklung zu bekommen, wurde eine einfache Finite Elemente Methode für den Ort und ein explizites Euler-Verfahren für die Zeit in NGSolve implementiert. Mit entsprechend kleinem Zeitschritt erhielten wir, selbst mit einem expliziten Euler-Verfahren, welches verstärkende Eigenschaften hat, eine stationäre Lösung (siehe Abschnitt 5.5). Damit existiert für (37) kein endlicher Wert. Eine genauere Analyse des Problems war also der nächste Schritt.

5.3 Analysis

5.3.1 Verallgemeinerung

Wir werden im ersten Schritt unsere Aufgabenstellung etwas verallgemeinern, und uns erst später mit der expliziten Aufgabe (36) beschäftigen. Betrachten wir dazu die Reaktions-Diffusionsgleichung

$$\begin{aligned} u_t - \Delta u &= f(u) \quad \text{in } \Omega,\ t > 0 \\ u(x,0) &= u_0(x) \quad \text{in } \Omega, \end{aligned} \tag{38}$$

mit Ω beschränkt, $u_0 \in C^1(\Omega)$ und eine nichtlineare Funktion $f(u)$. Außerdem werden noch Dirichlet-Randdaten

$$u(x,t) = h \quad \text{auf } \partial\Omega,\ t > 0$$

vorausgesetzt. Für solch allgemeine Aufgabenstellungen gibt es bereits einige Resultate bezüglich der Existenz und der Eindeutigkeit und daher auch ein gutes Verständnis der Lösung, zumindest lokal in der Zeit. Für eine genauere Betrachtung kann man (38) auch einfach als nichtlineare Version der wohlbekannten Gleichung

$$u_t = \Delta u$$

betrachten. Die Lösung von diesem Problem existiert für alle $t > 0$ und konvergiert gegen Null für $t \to \infty$. Dass es nun einen Zeitpunkt t geben könnte, bei dem die Lösung einen "blow-up" erzeugt, hängt also im Grunde nur von den charakteristischen Eigenschaften der Nichtlinearität ab.
Um ein besseres Verständnis für $f(u)$ zu bekommen betrachten wir dazu zunächst die Differentialgleichung

$$u_t = f(u) \tag{39}$$

wobei f positiv und stetig ist und

$$\int_0^\infty \frac{1}{f(u)} du < \infty$$

erfüllt. Diese Differentialgleichung hat eine eindeutige Lösung $u(t)$, die, je nachdem wie sich f verhält, schneller oder langsamer für eine endliche Zeit T_{max} gegen Unendlich strebt.

Ob nun die Lösung von (38) für alle Zeiten $t > 0$ existiert oder bei einer endlichen Zeit $t = T_{max} < \infty$ einen "blow-up" erzeugt, sprich

$$\lim_{t \to T_{max}} \|u(\cdot, t)\|_\infty = \infty,$$

wird, wie bereits in Kapitel (5.2) kurz angesprochen, davon abhängen, welcher der beiden Teile der Gleichung dominiert.

Speziell für den Fall $f(u) = \sigma u + |u|^{p-1}u$, mit $p > 1$, gibt es sehr viele Resultate, welche uns aber schlussendlich auch Aussagen über das Verhalten von (36) liefern wird. Siehe [5, Seiten 15-16].

5.3.2 Analytische Aussagen

Wir betrachten nun das Problem

$$\begin{aligned} u_t - \Delta u &= \sigma u + |u|^{p-1}u \quad \text{in } \Omega = (0,1)^2,\ t > 0, \\ u(x,t) &= 0 \quad \text{auf } \partial\Omega,\ t > 0, \\ u(x,0) &= u_0 \quad \text{in } \Omega, \end{aligned} \tag{40}$$

mit $p > 1$ und $\sigma \in \mathbb{R}$.

Satz 5.1. *Man betrachte das Problem* (40) *mit* $p > 1$, $\sigma \in \mathbb{R}$ *und* Ω *beschränkt. Sei* ϕ_1 *die kleinste Eigenfunktion des Laplace-Operators mit zugehörigem Eigenwert* λ_1, $u_0 \in L^\infty(\Omega)$ *nicht negativ und*

$$\int_\Omega u_0 \phi_1 dx > c := (max(0, \lambda_1 - \sigma))^{1/(p-1)}. \tag{41}$$

Dann gilt, dass $T_{max}(u_0) < \infty$ *und*

$$\limsup_{t \to T_{max}(u_0)} \|u(\cdot,t)\|_\infty = \infty.$$

Beweis: Mit Hilfe des Maximumxprinzips und da $u_0 \geq 0$ wissen wir bereits, dass $u \geq 0$ sein muss. Sei $y(t) := \int_\Omega u(t)\phi_1 dx$. Multipliziert man Gleichung (40) mit ϕ_1 und integriert über Ω, erhält man

$$y'(t) = \int_\Omega u_t(x,t)\phi_1(x)\ dx = \int_\Omega \Delta u(x,t)\phi_1(x)\ dx + \int_\Omega \sigma u(x,t)\phi_1(x)\ dx + \int_\Omega u(x,t)^p\phi_1(x)\ dx.$$

Integriert man den ersten Term auf der rechten Seite zweimal partiell und nützt aus, dass $\phi_1(x) = 0$ für $x \in \partial\Omega$, erhält man

$$\int_\Omega \Delta u(x,t)\phi_1(x)\ dx = \int_\Omega u(x,t)\Delta\phi_1(x)\ dx.$$

Da $-\Delta\phi_1(x) = \lambda_1\phi_1(x)$ folgt damit insgesamt

$$y'(t) = (-\lambda_1 + \sigma)\int_\Omega u(x,t)\phi_1(x)\ dx + \int_\Omega u(x,t)^p\phi_1(x)\ dx.$$

Sei o.B.d.A $c > 0$. Dann gilt $(-\lambda_1 + \sigma) = -c^{p-1}$ und somit, mit Hilfe der Jensen Ungleichung bezüglich der konvexen Funktion $x \mapsto x^p$,

$$\begin{aligned}&(-\lambda_1 + \sigma)\int_\Omega u(x,t)\phi_1(x)\ dx + \int_\Omega u(x,t)^p\phi_1(x)\ dx \\ &\geq -c^{p-1}\int_\Omega u(x,t)\phi_1(x)\ dx + \left(\int_\Omega u(x,t)\phi_1(x)\ dx\right)^p,\end{aligned}$$

bzw.

$$y'(t) \geq y(t)^p - c^{p-1}y(t) = y(t)^p\left(1 - \left(\frac{c}{y(t)}\right)^{p-1}\right).$$

Da $y(0) > c$ ist, gilt mit $\varepsilon := 1 - (\frac{c}{y(0)})^{p-1} > 0$

$$y'(t) \geq y(t)^p - c^{p-1}y(t) \geq \varepsilon y(t)^p. \tag{42}$$

Die Differentialgleichung $y'(t) = \varepsilon y(t)^p$ hat die Lösung $y(t) = (\varepsilon x - p\varepsilon x + C_1)^{1-p}$ welche einen "blow-up" bei endlicher Zeit erzeugt. Damit ist dies auch für die Differentialungleichung (42) sicher gestellt, womit letztendlich folgt, dass $u(x,t)$ nicht global existieren kann. ■

Damit wissen wir nun, unter welchen Vorraussetzungen ein "blow-up" für die Aufgabenstellung (40) erzeugt werden kann. Um nun auch Aussagen für unser Problem zu bekommen, muss die rechte Seite verallgemeinert werden.

Satz 5.2. *Man betrachte das Problem* (38)*, wobei* $f : \mathbb{R} \to \mathbb{R}$ *eine konvexe* $C^1(\Omega)$ *Funktion ist und* Ω *beschränkt ist. Man nehme an, es existiere ein* $a > 0$*, so dass* $f(s) > 0$ *für alle* $a \geq s$ *und*

$$\int_0^\infty \frac{1}{f(s)}\ ds < \infty. \tag{43}$$

Dann gilt die Aussage von Satz (5.1) *wobei c in (41) durch* $C = C(\Omega, f) > 0$ *groß genug ersetzt werden muss.*

Beweis: Sei wieder $y(t) := \int_\Omega u(t)\phi_1 dx$. Mit dem selben Argument wie im Beweis zuvor, erhält man

$$y'(t) = \int_\Omega u_t(x,t)\phi_1(x)\ dx = \int_\Omega u(x,t)\Delta\phi_1(x)\ dx + \int_\Omega f(u)\phi_1(x)\ dx \geq -\lambda_1 y(t) + f(y). \tag{44}$$

Da f konvex ist, folgt, dass $g(s) := \frac{f(s)-f(a)}{s-a}$ nicht fallend für $s > a$ ist und mit (43) außerdem, dass $g(s) \to \infty$ für $s \to \infty$. Also existiert eine Konstante $C \geq a$ so, dass $f(s) \geq 2\lambda_1 s$ für alle $s \geq C$. Falls $y(0) \geq C$, folgt mit (44), sofern $u(x,t)$ existiert, dass $y(t) \geq C$ und

$$y'(t) \geq f(y) - \lambda_1 y(t) \geq \frac{1}{2} f(y).$$

Damit kann $u(x,t)$ nicht global existieren. ■

Siehe [3, Seiten 92-93].

5.3.3 Anpassung der Aufgabenstellung

Da $f(u) = e^u$ eine konvexe Funktion ist, liefert uns Satz 5.2 nun die Bedingungen die erfüllt sein müssen, so dass ein "blow-up" erzeugt wird. Da wir dafür den kleinsten Eigenwert und die dazugehörige Eigenfunktion des Laplace-Operators benötigen machen wir uns zuerst darüber Gedanken.

Für das eindimensionale Problem auf $\Omega = (0,1)$ kann man mit Hilfe des charachteristischen Polynoms schnell eine allgemeine Lösung für das Eigenwertproblem

$$-\phi'' = \lambda\phi,$$
$$\phi(0) = \phi(1) = 0$$

finden, welche durch

$$\phi(x) = C_1 \sin(\sqrt{\lambda}x) + C_2 \cos(\sqrt{\lambda}x)$$

gegeben ist. Mit Hilfe der Randbedingungen erhält man dann $\phi(x) = \sin(\sqrt{\lambda}x)$ mit

$$\lambda = k^2\pi^2 \quad k \in \mathbb{N}.$$

Für das zweidimensionale Problem auf $\Omega = (0,1) \times (0,1)$

$$-\Delta\phi = \lambda\phi$$
$$\phi(x) = 0 \quad x \in \partial\Omega$$

geht man wie folgt vor: Man nehme an, dass die Lösung seperabel sei, also $\phi(x,y) = f(x)g(y)$. Aus $-\Delta\phi = \lambda\phi$ folgt dann

$$-\Delta\phi(x,y) = -\left(\frac{d^2 f(x)g(y)}{dx^2} + \frac{d^2 f(x)g(y)}{dy^2}\right) = \lambda\phi(x,y)$$

bzw.

$$-\frac{d^2 f(x)g(y)}{dx^2} - \frac{d^2 f(x)g(y)}{dy^2} = -g(y)f''(x) - f(x)g''(y) = \lambda f(x)g(y)$$
$$\Rightarrow -\frac{f''(x)}{f(x)} - \frac{g''(y)}{g(y)} = \lambda \Rightarrow -\frac{f''(x)}{f(x)} = \lambda + \frac{g''(y)}{g(y)} =: \mu$$
$$-f''(x) = \mu f(x)$$

Analog erhält man $g''(y) = (\mu - \lambda)g(y)$. Da die eindimensionale Lösung schon bekannt ist, folgt für die Eigenwerte

$$\mu = n^2\pi^2 \quad \text{und} \quad \lambda = \mu + m^2\pi^2 \quad m, n \in \mathbb{N},$$

mit den zugehörigen Eigenfunktion $\phi(x, y) = \sin(n\pi x)\sin(m\pi y)$. Der kleinste Eigenwert beträgt also $\lambda_1 = 2\pi^2$ und die kleinste Eigenfunktion ist definiert durch $\phi_1(x, y) = \sin(\pi x)\sin(\pi y)$.

Wir wollen nun unsere Aufgabenstellung (36) so abändern, dass wir mit Sicherheit einen "blow-up" erzeugen. Dazu müssen wir nur u_0 ändern. Wie bereits erwähnt, ist $f(u) = e^u$ eine konvexe Funktion und es gilt

$$\int_0^\infty \frac{1}{e^s}\, ds = 1 < \infty.$$

Um das Problem so einfach wie möglich zu halten, werden wir $u_0(x) = u_0$ konstant auf Ω annehmen. Es muss nun also der Startwert so gewählt werden, dass

$$\int_\Omega u_0(x)\phi_1(x)\, dx = u_0 \overbrace{\int_\Omega \phi_1(x)\, dx}^{=\int_\Omega \sin(\pi x_1)\sin(\pi x_2)\, dx} = u_0 \frac{4}{\pi^2} > C(f, \Omega) \tag{45}$$

erfüllt ist. Um die Konstante $C(f, \Omega)$ zu finden, werden wir nochmal den Beweis von Satz 5.2 betrachten. Dort muss $C \geq a$ so gewählt werden, dass $f(s) \geq 2\lambda_1 s$ für alle $s \geq C$. Bei unserem Problem entspricht dies $e^s \geq 2\lambda_1 s$. Die Gleichung $e^x = 4\pi^2 x$ hat eine positive Lösung bei $x = 5.35350...$, mit (45) muss also

$$u_0 \frac{4}{\pi^2} > 5.35350...$$

und damit $u_0 > 13.209...$ gelten. Wir wählen $u_0 = 14$. Die Aufgabe lautet nun

$$\begin{aligned} u_t - \Delta u &= e^u \quad \text{in } \Omega = (0,1)^2,\ t > 0, \\ u(x,t) &= 0 \quad \text{auf } \partial\Omega,\ t > 0, \\ u(x,0) &= 14 \quad \text{in } \Omega. \end{aligned} \tag{46}$$

5.3.4 Schranke für die Zeit

Für den "blow-up"-Zeitpunkt $T_{max}(u_0)$ für Aufgabe (40) gibt es mit $\sigma = 0$ und $\int_\Omega u_0\phi_1\, dx \geq (2\lambda_1)^{1/(p-1)}$ die Abschätzung

$$T_{max}(u_0) \leq \frac{2}{p-1}\left(\int_\Omega u_0\phi_1\, dx\right)^{1-p}.$$

Da die Funktion $p \mapsto \frac{2}{p-1}\left(\int_\Omega u_0\phi_1\, dx\right)^{1-p}$ streng monoton fallend ist und $e^x > x^2$, folgt mit $p = 2$ für unser neues Problem (46)

$$T_{max}(14) \leq \frac{\pi^2}{28} = 0.352485...$$

Siehe [3, Seite 94].

5.4 Diskretisierung

Wir beschäftigen uns jetzt mit der Diskretisierung unserer Aufgabenstellung. Wir werden dafür eine Finite Elemente Methode im Ort und eine Runge-Kutta-Methode für die Zeit verwenden.

5.4.1 Ortsdiskretisierung

Gegeben sei die partielle Differentialgleichung

$$\begin{aligned} -\Delta u - e^u &= 0 \quad \text{in } \Omega = (0,1)^2, \\ u(x) &= 0 \quad \text{auf } \partial\Omega, \\ u(x) &= 14 \quad \text{in } \Omega. \end{aligned} \tag{47}$$

Als erstes machen wir uns Gedanken über die schwache Formulierung. Dazu multiplizieren wir (47) mit Testfunktionen $v \in V := H_0^1(\Omega)$ und integrieren über Ω. Nach anschließender partieller Integration erhalten wir

$$\int_\Omega \nabla u \nabla v - \int_\Omega e^u v = 0. \tag{48}$$

Gesucht ist nun $u \in V$, so dass (48) für alle $v \in V$ erfüllt ist. Für die Diskretisierung benötigen wir eine (reguläre) Triangulierung $\mathcal{T}$ und einen Ansatzraum für die Testfunktionen. Wir wählen $V_h := \{v \in H_0^1(\Omega) \mid v \in \mathcal{P}^p(T)\ \forall T \in \mathcal{T}\} \subset V$, wobei $\mathcal{P}^p$ Polynome der Ordnung p sind. Wir unterteilen unser Problem nun in zwei Teile, und zwar in

$$\begin{aligned} a(u,v) &:= \int_\Omega \nabla u \nabla v \\ b(u,v) &:= -\int_\Omega e^u v. \end{aligned}$$

Für die Finite Elemente Methode wählt man nun eine Basis $(\varphi)_i^N$ für V_h und bekommt so die Lösung $u_h(x) = \sum_i^N u_i \varphi_i(x)$. Gesucht sind also die Koeffizienten u_i. Dafür erstellt man sich folgende Matrix (auch genannt Steifigkeitsmatrix)

$$A := \begin{pmatrix} \int_\Omega \nabla\varphi_1 \nabla\varphi_1 & \cdots & \int_\Omega \nabla\varphi_1 \nabla\varphi_N \\ \vdots & \ddots & \vdots \\ \int_\Omega \nabla\varphi_N \nabla\varphi_1 & \cdots & \int_\Omega \nabla\varphi_N \nabla\varphi_N \end{pmatrix}.$$

Für $b(u,v)$ funktioniert dies leider nicht ganz so einfach, da (im Gegensatz) zur Ableitung die Funktion e^u nicht linear ist. Wir werden jedoch sehen, dass dies später für die Diskretisierung der Zeit kein Problem darstellt. Mit Hilfe der Basis $(\varphi)_i^N$ und einem gegebenen $u_h(x) = \sum_i^N u_i \varphi_i(x)$, wobei $\boldsymbol{u} = (u_1, ..., u_N)$ den Koeffizientenvektor entspricht, erzeugen wir uns den Vektor

$$B(\boldsymbol{u}) := \begin{pmatrix} -\int_\Omega e^{u_1\varphi_1}\varphi_1 \\ \vdots \\ -\int_\Omega e^{u_N\varphi_N}\varphi_N \end{pmatrix}.$$

γ	γ	0
1	$1-\gamma$	γ
	$1-\gamma$	γ

0	0	0	0
γ	γ	0	0
1	δ	$1-\delta$	0
	δ	$1-\delta$	0

Tabelle 12: Butcher tableaus: links implizit, rechts explizit

Damit können wir folgende Gleichung formulieren

$$A\boldsymbol{u} + B(\boldsymbol{u}) = 0$$

welche nach dem Koeffizientenvektor $\boldsymbol{u}$ aufzulösen ist. Dazu könnte man zum Beispiel das Newton-Verfahren anwenden, was wir für unsere Zeitdiskretisierung aber nicht benötigen werden.

5.4.2 Zeitdiskretisierung

Für die Zeitdiskretisierung verwenden wir ein so genanntes IMEX Runge-Kutta Verfahren, das einen Teil der Gleichung implizit und den anderen explizit löst. Wir fordern an unser Verfahren außerdem, dass es sowohl A-stabil, als auch L-Stabil ist, weil das Verfahren damit für schnell fallende Lösungen auch numerisch schnell fallende Lösungen erzeugt. Dies ist bei unserer Aufgabenstellung wichtig, denn sonst könnte es passieren, dass die korrekte Lösung von (36) vielleicht schon stabil und damit stationär werden würde, die numerische Lösung aber aufschwingt und einen "blow-up" erzeugt. Des weiteren wollen wir keine numerische Oszillationen in unserer Lösung $u(x,t)$ erzeugen (daher A-Stabilität), da sich dies bei e^u schnell auswirken kann.

Wir werden eine Methode mit Ordnung 2 verwenden, die mit zwei Butcher-Tableaus angeschrieben werden kann (siehe Tabelle 12). Dabei ist $\gamma = \frac{2-\sqrt{2}}{2}$ und $\delta = 1 - \frac{1}{2\gamma}$. Der lineare Teil unserer Gleichung soll implizit und der nichtlineare Teil explizit gelöst werden. Die Aufgabenstellung (46) schreiben wir daher nun mit Hilfe von A und B aus Abschnitt 5.4.1 und der Massenmatrix M, mit $M_{ij} := \int_\Omega \varphi_i \varphi_j$, als

$$M\frac{u^{n+1}-u^n}{\tau} + Au^{n+1} + B(u^n) = 0,$$

wobei $\tau := t_{n+1} - t_n$ der Zeitschritt ist. Nun sehen wir auch warum es keine Probleme macht, dass wir für die Berechnung des Vektors $B(u)$ eine Lösung u benötigen, da wir für die Berechnung des nächsten Zeitschritts u^{n+1} immer nur die Lösung u^n des alten Zeitschritts benötigen. Damit ergibt sich folgendes Schema

$$\begin{aligned} u_1 &= u^n + (M + \tau\gamma A)^{-1}\left(-\tau\gamma B(u^n) - \tau\gamma A u_n\right) \\ u^{n+1} &= u^n + \tau\,(M + \tau\gamma A)^{-1}\left(-\delta B(u_1) - \gamma A u_1 - (1-\delta)B(u_n) - (1-\gamma)Au_n\right) \end{aligned}$$

Siehe [1]

5.4.3 Auswertung von $u(x,t)$

Für den gesuchten Wert $t^* = \min\{t > 0 : \|u(t)\|_{L^\infty(\Omega)} = +\infty\}$ muss man sich drüber Gedanken machen, wie man die Lösung aus der Diskretisierung auswertet. Da es sich bei (46) um ein symmetrisches Problem handelt, liegt die Vermutung nahe, dass der "blow-up" im Zentrum von

Ω, also bei $(0.5, 0.5)$, auftreten wird. Da der Wert der Lösung kurz vor dem "blow-up" sehr groß sein wird, hat man hier das Problem, dass numerische Effekte auftreten und die Lösung daher nur in der "Nähe"vom Zentrum ihr Maximum haben wird. Die Idee ist nun folgende: Im ersten Schritt wird ermittelt, auf welchem Element $T \in \mathcal{T}$ sich der Punkt $(0.5, 0.5)$ befindet. Da bei der Finiten Elemente Methode immer nur auf einem Referenzdreieck $\hat{T}$ gerechnet wird, gibt des zu jedem Element T auch eine entsprechende Tranformation $J : \hat{T} \to T$. Man kann sich schnell überlegen, dass $h := \sqrt{|\det J'|}$ in etwa dem "Durchmesser" des Dreiecks T entspricht. Wir wählen nun im Abstand h weitere 8 Punkte um $(0.5, 0.5)$ (siehe Abbildung 7) und überprüfen, auf welchen Dreiecken diese Punkte liegen. Das Maximum wird nun auf all jenen Dreiecken gesucht die so gefunden wurden.

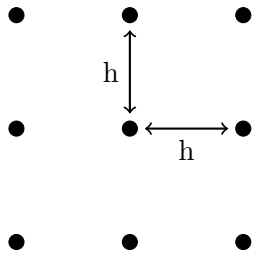


Abbildung 7: Anordnung der Punkte

5.5 Numerische Ergebnisse

5.5.1 Zeitschleife

Wir haben nun eine vollständige Diskretisierung unseres Problems. Dieses Verfahren wurde inklusive der Randbedingungen in NGSolve implementiert. Das Verfahren rechnet so lange, bis die Lösung entweder einen gewissen Wert überschritten hat, oder der Wert der Lösung auf 0 gefallen ist. Dies passiert wenn die Lösung nicht mehr verarbeitet werden kann, sprich ein "blow-up" entstanden ist (NGSolve setzt dann die Lösung automatisch auf 0). Die Zeitschleife sieht dann wie folgt aus:

```
1  double gamma = (2 - sqrt(2)) / 2;
2  double delta = 1 - 1. / (2 * gamma);
3
4  BaseMatrix & mata = bilinearformA->GetMatrix();
5  BaseMatrix & matm = bilinearformM->GetMatrix();
6  BilinearFormApplication applyb(bilinearformB);
7  BaseMatrix summat = matm + dt * gamma* mata;
8  BaseMatrix & invmat = *summat.InverseMatrix();
9
10 BaseVector & vecu = gridfunction->GetVector();
11
12 int run=1;
13 double max;
14 double t=0.0;
15
16 while(run==1)
17 {
18         t+=dt;
19          {
20                  bun = applyb*vecu;
21                  d = 0.0;
```

```
22                  d -= bun;
23                  d -= mata*vecu;
24                  u1 = vecu + dt*invmat*d*gamma;
25
26                  bu1 = applyb*u1;
27                  d = vecf;
28                  d -= (1 - delta)*bu1 + delta* bun;
29                  d -= gamma*mata*vecu + (1 - gamma)*mata*u1;
30                  vecu += dt * invmat * d;
31          }
32
33          max = gfumax(vecu, lh);
34
35          if ((max >= umax) || (max == 0 ))
36            run = 0;
37  }
```

5.5.2 stationäre Lösung

Wie bereits in Abschnitt (5.2) erwähnt, bekommt man für die originale Angabe, also mit $u_0 = 1$, eine stationäre Lösung. Dies wurde mit verschiedenen Methoden gestestet. Die Werte der Lösung werden zwar immer kleiner, aber sie konvergiert nicht gegen die Nulllösung $u(x,t) = 0$, da diese unsere Differentialgleichung nicht erfüllt. In Abbildung (8) und (9) sieht man die stationäre Lösung und den Verlauf des Maximums.

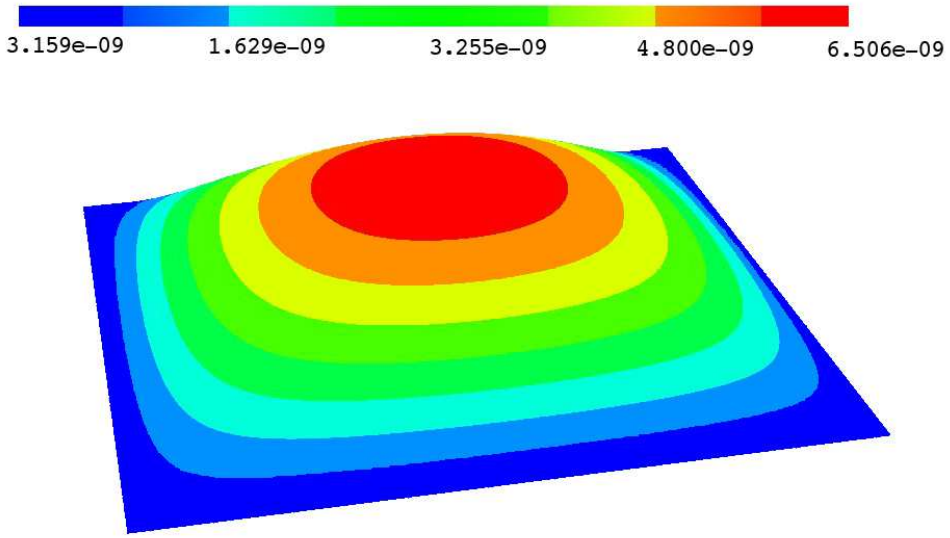


Abbildung 8: stationäre Lösung $u(x,t)$

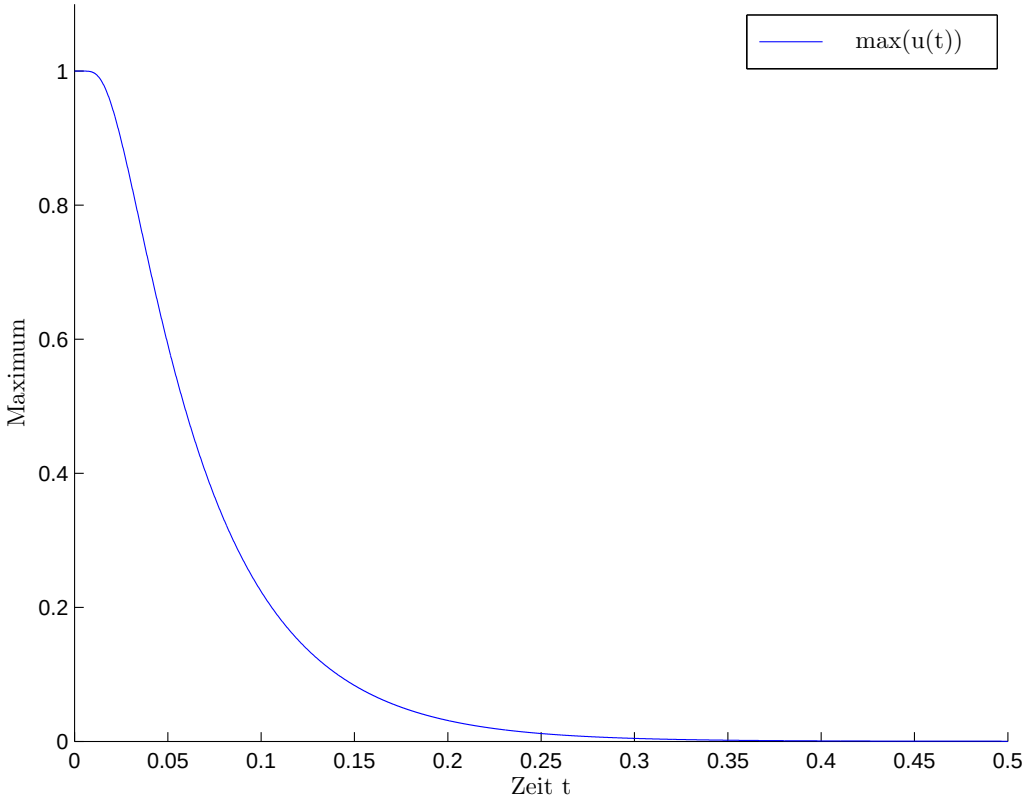


Abbildung 9: Konvergenzplot des Maximums

5.5.3 "blow-up" Berechnung

Für die Berechnung müssen wir nun folgende 3 Punkte entscheiden:

- Die Wahl des Netzes, sprich die Anzahl der Elemente (adaptive FEM wird hier nicht in Betracht gezogen)
- Die Wahl des Polynomgrades p
- Die Wahl des Zeitschrittes τ

Wir haben uns für drei Netzte entschieden. Die erste Triangulierung $\mathcal{T}_1$ hat 878 Elemente, die zweite $\mathcal{T}_2$ 2740 Elemente und die dritte $\mathcal{T}_3$ 5836 Elemente. Die Frage welchen Polynomgrad man wählt, ist nicht so einfach zu beantworten. Da unser Runge-Kutta-Verfahren die Ordnung 2 hat, werden wir auch nur zwischen $p = 1$ und $p = 2$ unterscheiden, da die Genauigkeit der Lösung mit einer größeren Ordnung für die Ortsdiskretisierung durch die niedrigere Ordnung der Zeitdiskretisierung wieder verschlechtert werden würde. Für die Wahl der Schrittweite könnte man entweder eine Schrittweitensteuerung implementieren oder aber schon von Beginn an mit einem entsprechend kleinen Zeitschritt arbeiten. Da die Schrittweitensteuerung auch mit einer Toleranz arbeitet und dies einen weiteren variablen Parameter hinzufügen würde, haben wir uns für eine feste Wahl von τ entschieden. Dies ergab dann folgende Ergebnisse t^*:

	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$
$p = 1$	0.000265253	0.000956505	0.002156273
$p = 2$	0.000251419	0.000945702	0.002145048

Tabelle 13: Numerische Ergebnisse

Hier kann man zwei Effekte erkennen. Als erstes sieht man, dass je feiner die Triangulierung wird der gesuchte Zeitpunkt immer später auftritt. Dies liegt daran, dass für ein zu grobes Netz die Ortsdiskretisierung die exakte Lösung zu schlecht approximiert, bzw. genauer gesagt, dass der Laplace-Operator weniger dämpft, als er das eigentlich tun sollte und sich dementsprechend das Maximum der Lösung zu schnell erhöht.
Dass Der "blow-up" für eine höhere Polynomordnung früher auftritt, ist schwierig zu interpretieren. Einerseits könnte man meinen, dass eine genauere Ortsdiskretisierung eine bessere Lösung erzeugt und man damit einen genaueren Zeitpunkt erhält. Andererseits erzeugt man mit einem höheren Polynomgrad auch größere Oszillationen, welche sich auf der rechten Seite unserer Aufgabenstellung (46) natürlich stärker auswirken und somit der 'blow-up" früher erzeugt werden könnte.

Wie man sehen kann, ist eine Berechnung des Zeitpunkts relativ schwierig und benötigt eine noch genauere Analyse des Problems. Außerdem sollte man noch andere Methoden für die Orts- und Zeitdiskretisierung zum Vergleich heranziehen, wie zum Beispiel Finite Volumen. In Abbildung (10) sieht man noch den Verlauf des Maximums von $u(t)$, wobei hier $\mathcal{T}_3$ verwendet wurde. Man kann erkennen, wie der Wert immer mehr ansteigt, bis die Lösung nicht mehr verarbeitet werden kann und auf 0 abfällt.

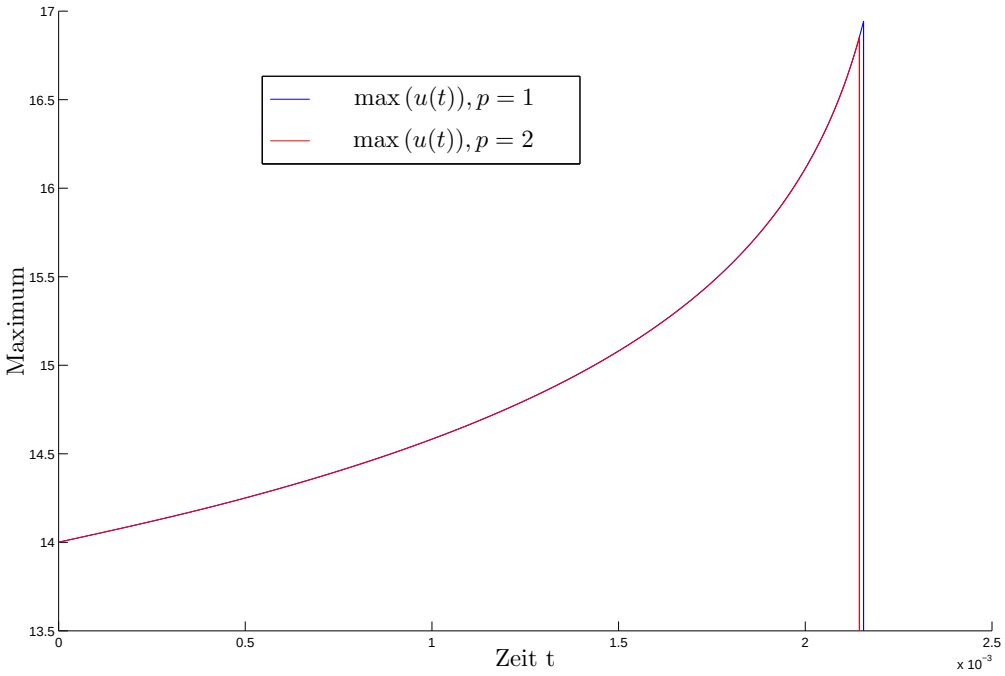


Abbildung 10: Plot bist zum "blow-up"

Literatur

[1] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Appl. Numer. Math*, 25:151–167, 1997.

[2] Folkmar Bornemann, Dirk Laurie, Stan Wagon, and Jörg Waldvogel. *Vom Lösen numerischer Probleme - Ein Streifzug entlang der SIAM 10x10-Digit Challenge.* Springer, 2006.

[3] Philippe Souplet Pavol Quittner. *Superlinear Parabloic Problems, Blow-up, Global Existence and Steady States.* Birkhäuser Advanced Texts, Basel Boston Berlin, 2007.

[4] Polygamma funcitons: Series representations (subsection 06/02). (abgerufen am 26.06.2014).

[5] Aappo Pulkkinen. Blow-up in reaction-diffusion equations with exponential and power-type nonliniarities, 2011.

[6] Weisstein, Eric W. "Polygamma Function."From MathWorld–A Wolfram Web Resource. (abgerufen am 26.06.2014).

[7] WolframAlpha. http://mathworld.wolfram.com/PrimeZetaFunction.html (abgerufen am 26.06.2014).

The Digit Challenge - Resultate Team 2

Stephan Pfannerer (1026392)
Georg Simbrunner (0900487)
Bernhard Wipfler (1025879)

26. Juni 2014

Für Franciscus (13.3.2013-6.6.2014), Stephans Laptop

Inhaltsverzeichnis

Zusammenfassung

Ziel der Digit Challenge war es, fünf unterschiedliche Aufgaben numerisch zu behandeln und ein unbekanntes Ergebnis auf möglichst viele Stellen genau zu bestimmen. Teilweise war dies mit eleganten Umformulierungen des Problems sehr einfach möglich. So gelang uns bei Aufgabe 1 mit passender Software 10000 Stellen exakt zu bestimmen. Bei Aufgabe 3 aber scheitert man bereits zu beweisen, dass überhaupt eine Stelle wirklich richtig ist.

Erschwert wurde der Fortschritt kurzzeitig auch durch einen Defekt eines unserer Arbeitsgeräte, welches außerdem die Vernichtung von experimentellem Quellcode zur Folge hatte. So sind Versuche einige Algorithmen in C++ mit Hilfe der Graphikkarte zu beschleunigen verloren gegangen.

Wir haben danach hauptsächlich mit den Tools Matlab/Octave, R und PARI/GP gearbeitet.

Jedes Teammitglied hat sich besonders in eine Aufgabe vertieft und die theoretischen Aspekte behandelt. So ergibt sich folgende Arbeitsaufteilung:

- Aufgabe 1: Bernhard Wipfler, Georg Simbrunner (Kapitel 1.4)
- Aufgabe 2:
  - Theorie: Georg Simbrunner
  - Implementierung in Matlab, PARI/GP und Optimierungen: alle
- Aufgabe 3: Stephan Pfannerer
- Aufgabe 4:
  - Theorie: Georg Simbrunner, Bernhard Wipfler
  - Implementierungen: Georg Simbrunner, Stephan Pfannerer
- Aufgabe 5:
  - Implementierungen: Georg Simbrunner, Stephan Pfannerer
  - Analytische Aussagen: Bernhard Wipfler

1 Aufgabe 1

1.1 Problembeschreibung

Im folgenden Abschnitt sei $\mathbb{P}$ die Menge aller Primzahlen. Zu berechnen ist $s = \sum_{p\in\mathbb{P}} \frac{p\cdot(p-1)}{p^\pi}$.

1.2 Umformulierung mittels analytischer Zahlentheorie

Die Riemannsche Zetafunktion hat die bekannte Darstellung als Euler-Produkt:

$$\zeta(x) = \prod_{p\in\mathbb{P}} \frac{1}{1-p^{-x}} \left(= \sum_{n=1}^{\infty} \frac{1}{n^x}\right)$$

Logarithmieren ergibt:

$$\log(\zeta(x)) = \sum_{p\in\mathbb{P}} \log\left(\frac{1}{1-p^{-x}}\right) = -\sum_{p\in\mathbb{P}} \log\left(1-\frac{1}{p^x}\right)$$

Sei die Primzetafunktion definiert durch

$$P(x) := \sum_{p\in\mathbb{P}} \frac{1}{p^x}$$

Man entwickelt den natürlichen Logarithmus im Punkt 1 und erhält nach [17, S.5]:

$$\log(\zeta(x)) = -\sum_{p\in\mathbb{P}} \log\left(1-\frac{1}{p^x}\right) = \sum_{n=1}^{\infty}\sum_{p\in\mathbb{P}} \frac{1}{np^{nx}} = \sum_{n=1}^{\infty} \frac{P(nx)}{n}$$

Mittels Möbiusscher Umkehrformel ergibt sich:

$$P(x) = \sum_{n=1}^{\infty} \frac{\mu(n)\log\zeta(nx)}{n} \tag{1.1}$$

Wobei $\mu(\cdot)$ die Möbiusfunktion ist.
Aufgrund der schnelleren Konvergenz dieser Summe wird diese implementiert. Damit erhält man

$$s = \sum_{p\in\mathbb{P}} \frac{p(p-1)}{p^\pi} = \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-2}} - \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-1}} = P(\pi-2) - P(\pi-1)$$

1.3 Implementierung mittels Pari/GP

In PARI/GP sind einige mathematische Funktionen, wie die Riemannsche Zetafunktion und die Möbiusfunktion, bereits effizient vorimplementiert. Damit lässt sich die Primzahlzetafunktion durch folgendes einfaches Programm in PARI/GP-Notation berechnen:

Listing 1: Pari/GP-Algorithmus für die Primzahlzetafunktion

```
1 pz(x,N)={
2 S=0;
3 for(n=1,N, a=moebius(n); if(a!=0,S=S+a*log(zeta(n*x))/n));
4 return(S);
5 }
```

Wobei N die Anzahl der Summanden in (1.1) ist.
Man erhält also mit `pz(π-2,N)-pz(π-1,N)` das gewünschte Ergebnis.

1.3.1 Auslöschung

Ein Problem bei dieser Art von Berechnung sind mögliche Auslöschungen. In diesem Abschnitt wird gezeigt, dass keine Auslöschungen auftreten.
Man sieht, dass Auslöschungen an zwei Stellen auftreten könnten:

- innerhalb der jeweiligen Summen
- bei $P(\pi-2)-P(\pi-1)$

Sei $f_s(x)=\frac{1}{x^s}$ und $s>1$, also $f_s(x)$ streng monoton fallend.
Aus dem Integralkriterium folgt:

$$0 \leq \zeta(s)-1=\sum_{n=2}^{\infty} f_s(n) \leq \int_1^{\infty} f_s(t)dt \leq \sum_{n=1}^{\infty} f_s(n)=\zeta(s)$$

Berechne $\int_1^{\infty} f_s(t)$.

$$\begin{aligned}\int_1^{\infty} f_s(t)dt &= \lim_{N\to\infty}\int_1^N \frac{1}{t^s}dt = \lim_{N\to\infty}\frac{1}{N^{s-1}(1-s)}-\frac{1}{1-s}=\frac{1}{s-1}\\ &\Rightarrow 1 \leq \zeta(s) \leq \frac{1}{s-1}+1=\frac{s}{s-1}\\ &\Rightarrow \log\zeta(s) \leq \log\frac{s}{s-1}\end{aligned}$$

Daraus ergibt sich eine Abschätzung für $s>1$ da $\frac{1}{x}$ monoton fallend:

$$\log\zeta(s) \leq \log\frac{s}{s-1}=\log(s)-\log(s-1)=\int_{s-1}^{s}\frac{1}{x}dx \leq \int_{s-1}^{s}\frac{1}{s-1}dx=\frac{1}{s-1}$$

Also fällt $\log\zeta(s)$ zumindest linear und damit die Glieder $\left|\frac{\mu(n)\log\zeta(nx)}{n}\right|$ in der Summe (1.1) zumindest quadratisch. Daher kann keine Auslöschung auftreten.
Da $P(\pi-2)\approx 1.8$ und $P(\pi-1)\approx 0.4$ kann auch bei $P(\pi-2)-P(\pi-1)$ keine Auslöschung auftreten.

1.3.2 Ergebnis

PARI/GP bietet die Möglichkeit mit beliebig vielen Stellen (eingeschränkt vor allem durch den Arbeitsspeicher) zu rechnen.
Dadurch ergeben sich 2 Variablen mit denen man die Genauigkeit vom Ergebnis erhöhen kann: Anzahl der Stellen (\p in PARI/GP) und Anzahl der Summanden N in (1.1).
Es hat sich gezeigt, dass die Summe (1.1) sehr schnell konvergiert, also man sehr viele Stellen bekommt ohne viele Summanden zu berechnen.
Das folgende Ergebnis wurde mit 10000 Stellen Genauigkeit berechnet mit $N=2^{30}$ Summanden. Das Ergebnis wurde mit 11000 Stellen Genauigkeit und $N=2^{30}+1000$ überprüft. Diese beiden Ergebnisse stimmten auf den ersten 10000 Stellen überein, also auf allen im ersten Durchgang berechneten.
Das garantiert die Korrektheit dieser Stellen, die der Vollständigkeit halber hier angeführt sind:

```
1.41702525894502464667989471995404264872028939967775267916650262120972230367292792288230740045164225687961089332754335640671640263722796672446549688786446744955022457395947107941232376246223753770875706867626655421097143693469807110026
```

87106377764692948694076190426590074872389061131899391269173751064835086540235341901890382305844104355420514179286167140934434279025220339965953191837871961896188903350469762332949670193657339520125678273079369718146808537061989118198115660867322406156689209011375657016696956826221651054329155650243562157089140998466842511698119933815090947662098348612094755490049752144671960284116574085723059346129421390580500705384534387808773706058439540064924861330187610817807044519409605709645579408545486491229108258531300414905271826257360119933283301687996228526104107513744347423751586544570461846410651378977066623394426505640369897246203083477436866412369548331226372680930180738800923907443756183724770963123615545083026496632264898994237153529384352901811904115763451351519770911924151217870526146109704496565676347289489367918055946539248108823727024833345045984931839650154377134834461070017595090989366000366981574741189716594674981919724276355131936182478969806026830501448616108409904066175078371779384062300885704654581943695846505768725995319987351986244771326412123876071570251392812726166469772848351124769047997031908666734681458372986201473337812749350596171920820142639504414065468496711006109128947579907121785344365357293716771142568298848847502039975475773405887848084785404340972852910792886806553003935940729500054366634920393599666851634879636331535925034776550336180409424742318576748678609939897523156266534304229684784335217872178702193600687493983533829183560285757713549479981292858250259161860919493257822437048021340610660570153377652507582357017651802304988364834590507392643609080296555693223235824113490743396404392674376380550623997231114139191227005274709180570355228793173967542351141625360625918279332657006184318075633562364759132575359734385918598271426968294652130044448574122853486592703177889708469330317595064938027606417214867325305622470415788415545236404809009638045233611948169903717681898428755465483130508105288074341748477345700576899098545688654667150231439063439170649932955220483649824893815499445432219395142528331791299859063858170415685773096717249010342548099604327025224578526662347705120276561555995793510645179557890745283834436766675486029832335608019090433357873613998029314259876933995132963660989608497923757574646059804089019393028231343354907807062990083615434311992720969870011513831133328857501029291951212955203404415796097201304799641745725056438518868349892979183846661120523465609202140091649461623991215032395670367179340853031609354330761763804574369772700605127957960738314809380499188717315678896800305181446255159926176897998633119684033741501622688081893696712316544935510566154430874108658981644111495440245338649528637627231628545292057778167911939933680035679191124677803156644474423022723813125928901065986176700653170784285643265717659027381120583586172247684602285640559686562217769371341843300019195506186944174839542622241892311659623135870417341239615241384129034675075211581721545441374117311893262235386928274975386122117671524617058253047510132336822174103812729753431486964107741992716638174867755809644294272949250602505010578100688018740456743384677639877386398377184249863010816110462509835628626708311049340761702278395592394121119015407459687121767171265837920990475879393243933700739357552788714323060044790673089544213720151756682842645453394761707706805559319392194554280037699759504867147223048551271923525007381528437204660941809560405521456163518259918146565899602637388043410825995042603391090347858617866220433800351660989793591487765744292709436335586829870221011041754437177300990949114015432880982222401273551089609587158982856809698285966725403279853719916102309182000614911433478834432043114449560566676476496730398712859282403625857887532108993963189554340576384648130173126799766921641425408795214118481308615988981 4

65665353767726356179670665611633628710750481724520460642476592983585761976423758852109228428945454634038360078757329161053169848709312723917449644281882999780190751984571037444947093015379246203451690475146209230367487609789404148652363603920095495202239010587948582747681656564780812693413798837347819990816179071267371717792284581917071256742770503930917511893050988537512284938340291811038208648508592667276064014519959032594723902322667346457942412626388388642336842556340847414948112033082106851283455675364179907846603988928831407720361262275966353620137610386044414791487864898784254684670923650670905082482902782807905240200430785186093367832349492900892332814205418518541384567315084594875741766360009787995654299681574713961750606173768388516095070424198709633399246495647141577159823933220218716359094804181662958435689655791992343868715622869933099852249737853908018659408094081584185765576853334453901632102368187635592520936374907663056726289825191470467352710142542131035145470831523552336304982457878438014755406072230350571361407960185200167905829983030699310149870329164094624966091191064977972271884310065824265688257471556390601917424677167684776220267896344872047010481180834952102791288781689006879250612923157019848448408377313372395715701142781386589634380839726555658085689764216807999110721408462153249027782881467186775977838502123699114685913342896698088520275056312367850684231627513893824141301650813278646573823913089597642184883422383957295515751927787361749985200226815649901833876542393171334969597812622576761644040432849008879761380122321985825174112499191957573102333555623709089333754030716050979694373349751694618959757234215972562725426846109693706624259313575376393309025361754768230757115521760725044032857491355754632451492045798718636258045845497846653849335763679469818964990766621039098359040818152733773211401117308865705236794853677676214420910386868576171354125197673141938428396964803120940698938183990720704504882212168398629742338728578240003945234557543804943274672489635166184331741570475087912631460873631161271898159193815157436295773320795076734583475189383522488177176005540419177319291214994779540958228344500782089433302505669014444913688140799207285666141542380439847776898752325212805807651888491726012237299710910593620213988505089655126998241236275750101549610349066167497921693758806222927069337928124797618624424432075981255634872969775425300365262840678055381718092142692973042954673768153105302509681929841209956567909382718239276075556296117369849833518922211461625190633229052604517930540021077760910771367121285800024965014661854493383892300112910629662458323165887767293176630172675076094030505311065271748600149495919130213356404287568494589872318399199094426712060388061423912392500525951621432771610316499926529311012420577762688974768512169369862433260686679441010614057313449868164702505470412490701318954861286206334812046157386564493105849585781699190038904988225819638779683241487540190345187266638636889078019648052112109692266373319887643143745164620717602323434326089208536477308054922584562883228234724560717836856396438632793373181161375978747814544717353682158284462059576851130835308219918612914032907412313235171136973056423581839913125178430321682820188089234901246584567356339699634815247095955682219854090019837562301490731436704409686924142329814336166934457207313538063458160752981635198056334914988302777053463269639323690376346942396402358727251770925479521209072925920717483368959984201172827330643996600656479638557255544138353979999530418494340287559269576294957505344134399625131077826783369095775502771090234947258043505042361072191018251182170665167790723889685726611342536350661296839726772722232894482915310067567792062495105507426890283220502346371503841536427521794805911500616406748246541108707780309578380986468 77

650943499117347560813905761215671646868050454107108736891727233894237896257653
378950423379189552305087215185195297725254271788513349706505492662223450335014
466022637555388609057417948732825050656617757687129196623698861288477661297081
147375105191181585577970376588511901375460087907691062248566095794540928913884
302864997800608399432011337880610643153105995662761594065002840679448555015822
932347719508495299551010270420396735054793891206816186670503688952181033102479
298063783531879322908427434714328068115009746559756479218825124515941990320609
771886747912671367768482332634739853964110210734895179180153918576763037562961
845511085230161844394485393907099236364946871573311846674906670740660661147 12
321627988191970583853510122347571312737714505140334004476251417358101755299230
388643433995821087197757246034140668495282362586617295787374541035446521432346
133623438840932304489104101190226739091722947595224686072628745116790424511423
712772510424026262970688272712110862813578928826586748980547408434715328457501
173770128622614414987847405365902297231751227223458363304127043042682340754666
070890954767818499878558172996685112034471826985359726778235497182441761638089
591803331547644014283845069986318471947069299321495483122411589120541288220766
230144146614360358692785255346427797632847500064893011056255083096272106538323
671277304850816073270804234385479925855906904648807586564388484197605624954052
924274610082270679272980439010596502678890741079935921360983280618725518256648
702592065416862517905194040537195313161344635666801705144433394147085361356373
716813219766707511729246589337463922647037455424796848365841824482569669237823
039774633194628343591323477590140223798210858427926894540874560203526764469760
523971595429883934721201693029986253108438713374955676541287604707960497591834
708586067645200268584627601454518539181576647528330626540806945874450980772109
654524247805719795844490856412042672182518762490366675596084631736385844083540
286824809045954560744584138046430739707458320304841816894369373975804658934585
983573002029610006076699665074219623764417774646560978283115768368855633902669
18628514864686779

140934434279025220339965953191837871961896188903350469762332949670193657339520125678273079369718146808537061989118198115660867322406156689209011375657016696956826221651054329155650243562157089140998466842511698119933815090947662098348612094755490049752144671960284116574085723059346129421390580500705384534387808773706058439540064924861330187610817807044519409605709645579408545486491229108258531300414905271826257360119933283301687996228526104107513744347423751586544570461846410651378977066623394426505640369897246203083477436866412369548331226372680930180738800923907443756183724770963123615545083026496632264898994237153529384352901811904115763451

Eine genauere Auswertung ist nur eine Frage von mehr Rechenzeit. Obwohl Mathematica langsamer als Pari/GP ist, war der Aufwand für 1000 Stellen im Bereich von wenigen Minuten auf einem modernen Laptop, der auch nicht gerade eine Ausgeburt an Rechenleistung ist. Das Ergebnis stimmt mit der Pari/GP-Arithmetik überein. Durch die effiziente Umformulierung des Problems und die zahlentheoretischen Algorithmen am Markt ist somit eine effiziente Berechnung auf nahezu beliebig viele Stellen kein Problem.

2 Aufgabe 2

2.1 Problembeschreibung

In dieser Aufgabe betrachteten wir eine unendlichdimensionale Matrix $A = (a_{ij})_{i,j\in\mathbb{N}^\times}$, die einen unendlichdimensionalen Operator $A : \mathbb{N}^\times \times \mathbb{N}^\times \to \ell^2$ auf den Raum der quadratisch summierbaren Folgen ℓ^2 definiert. Die Einträge waren dabei wie folgt gegeben (wobei $\mathbb{P}$ die Menge der Primzahlen bezeichnet):

$$a_{ij} = \begin{cases} \frac{1}{i+j-1} & i+j \in \mathbb{P} \\ \frac{1}{(i+j-1)^2} & i+j \notin \mathbb{P} \end{cases} \tag{2.1}$$

Zu berechnen war die reelle Zahl a - die Spektralnorm des Operators A - die gegeben ist durch

$$a = \|A\| := \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} \tag{2.2a}$$

$$\|x\|_2^2 := \sum_{j=1}^{\infty} x_j^2 \tag{2.2b}$$

2.2 Konvergenzaussage mittels Approximation durch endlichdimensionale Teilräume

Als nächstes wollen wir zur numerischen Berechnung der Spektralnorm von A ein Verfahren motivieren, welches darauf setzt, dass die Spektralnorm auf ℓ^2, welche einen unendlichdimensionalen Operator definiert, durch die Spektralnormen endlichdimensionaler Teilmatrizen $A_k \in \mathbb{R}^{k\times k}$ approximiert werden kann.

$$A = \left(\begin{array}{c|c} A_k & \dots \\ \hline \vdots & \ddots \end{array}\right)$$

Lemma 2.1. *Für $1 \leq n \leq m$ gilt*

$$\|A_n\| \leq \|A_m\| \leq \lim_{k\to\infty} \|A_k\| = \|A\| \leq \pi \tag{2.3}$$

Beweis. Wir führen den Beweis analog zu [2, S. 57]. $P_n : \ell^2 \to \operatorname{span}(e_1, \dots, e_n) \subset \ell^2$ bezeichne ferner die Orthogonalprojektion von ℓ^2 auf den n-dimensionalen Teilraum, der durch die ersten n Basisfolgen $(e_j)_k = \delta_{ij}$ aufgespannt wird. Diesen Teilraum können wir mit $\mathbb{R}^n$ identifzieren und erhalten $A_n = P_n A P_n$. Für $n \leq m$ gilt $P_n = P_n P_m = P_m P_n$ und daher gilt $A_n = P_n A_m P_n$. Mittels der Submultiplikativität der Operatornorm und der Tatsache, dass für Orthogonalprojektionen $\|P_n\| \leq 1$ gilt, erhält man

$$\|A_n\| \leq \|P_n\|^2 \|A_m\| \leq \|A_m\| = \|P_m A P_m\| \leq \|P_m\|^2 \|A\| \leq \|A\| \tag{2.4}$$

Aufgrund von obiger Abschätzung ist $\|A_n\|$ monoton wachsend in n. Wir benötigen jedoch ein Argument, dass $\|A\| < \infty$. Leider divergiert die Frobenius-Norm von A_k für $k \to \infty$ im Gegensatz zum Beweis von Bornemann, daher kann diese auch nicht zum Abschätzen der Spektralnorm herangezogen werden (dies ist sehr schade, da dies auch einige Aussagen über den betragsmäßig zweitgrößten Eigenwert und somit die Konvergenzrate zugelassen hätte). Jedoch gibt es für die Spektralnorm der Hilbertmatrix $H_{ij} = \frac{1}{i+j-1}$ eine direkte Abschätzung $\|H\| \leq \pi$[9, S.212f] für $(x_1, x_2, \dots), (y_1, y_2, \dots) \subset \ell^2$ und $\|x\|_2 = \|y\|_2 = 1$:

$$\|A\| = \sup_{\|x\|_2=1} \|Ax\|_2 = \sum_{i,j=1}^{\infty} x_i a_{ij} x_j \leq \sum_{i,j=1}^{\infty} \frac{x_i y_j}{i+j-1} = (x, Hx) \leq \pi \tag{2.5}$$

Folglich existiert der Grenzwert $\lim_{n\to\infty} \|A_n\| = \sup_{n\in N} \|A_n\| \leq \pi$ der monoton wachsenden Folge. Wegen der Vollständigkeit der Basisfolgen gilt $\lim_{n\to\infty} P_n x = x$ für alle $x \in \ell^2$ und daher gilt

$$\|Ax\| = \lim_{n\to\infty} \|P_n A P_n x\| \leq \lim_{n\to\infty} \|A_n\| \|x\| \tag{2.6}$$

□

und somit $\|A\| \leq \lim_{n\to\infty} \|A_n\|$.

Insbesondere bekommen wir durch obiges Lemma die Konvergenzaussage, dass die Matrixnorm, die für immer größere Teilmatrizen berechnet wird, letztendlich monoton wachsend gegen den Grenzwert a steigt. Insofern machte für uns das numerische Verfahren Sinn, mittels der Spektralnorm der endlichdimensionalen Teilmatrizen $A_k \in \mathbb{R}^{k\times k}$ möglichst viele Stellen zu berechnen.

2.3 Berechnung mittels Power-Iteration der Submatrix

Der einfachste und naheliegenste Ansatz zur Berechnung der Spektralnorm der Teilmatrizen A_k, welche unsere Näherung an die Norm des unendlichdimensionalen Operators liefert, ist die Potenzmethode nach von Mises (engl. Power-Iteration):

Data: Matrix A, Startvektor $z^{(0)}$ mit $\|z^{(0)}\|_2 = 1$ als Näherung für den Eigenvektor zum betragsgrößten Eigenwert λ_1
for *1,2,...* **do**
  $\widetilde{z}^{(k)} = Az^{(k-1)}$;
  $z^{(k)} = \widetilde{z}^{(k)} / \|\widetilde{z}^{(k)}\|_2$;
  $\mu^{(k)} = (z^{(k)})^T A (z^{(k)})$;
end
Result: $\mu^{(j)}$, eine Näherung für λ_1 der Matrix A und $z^{(j)}$, eine Näherung für den zugehörigen Eigenvektor nach j Iterationen

Algorithm 1: Pseudocode für die Potenzmethode nach von Mises.

Die Konvergenzaussage für die Methode liefert folgender Satz:

Satz 2.1. *Die (nicht notwendigerweise diagonalisierbare) Matrix $A \in K^{n\times n}$ habe die Eigenwerte*

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|,$$

ferner sei y ein linker Eigenvektor zum Eigenwert λ_1 mit $\|y\| = 1$. Ist $z^{(0)} \in \mathbb{K}^n$ ein Startvektor mit $\|z^{(0)}\| = 1$ und $y^* z^{(0)} \neq 0$, dann gilt

$$\limsup_{k\to\infty} \left| \|\tilde{z}^{(k)}\| - |\lambda_1| \right|^{\frac{1}{k}} \leq \left| \frac{\lambda_2}{\lambda_1} \right| \tag{2.7}$$

und es existiert ein Eigenvektor v von A zum Eigenwert λ_1 mit $\|v\| = 1$ und

$$\limsup_{k\to\infty} \|z^{(k)} - sign((\lambda_1)^k)v\|^{\frac{1}{k}} \leq \left| \frac{\lambda_2}{\lambda_1} \right| \tag{2.8}$$

wobei $sign(\lambda) := \frac{\lambda}{|\lambda|}$

Beweis. Siehe [8, S. 220f] □

Da die Berechnung der Spektralnorm der Berechnung des betragsgrößten Eigenwertes entspricht, können wir mittels der Approximation $\mu^{(j)}$ (Approximation von $|\lambda_1|$ nach j Iterationen) des betragsgrößten Eigenwerts λ_1 von A_k eine endlichdimensionale Näherung an die Spektralnorm des unendlichdimensionalen Problems finden. Wir benötigen jedoch noch als nicht unmittelbar ersichtliche Voraussetzung des Satzes, dass λ_1 ein Eigenwert der Vielfachheit 1 ist. Der Satz von Perron-Frobenius für positive Matrizen (d.h. $a_{ij} > 0$ für $1 \leq i, j \leq n$) liefert jedoch genau die Aussage, dass der betragsgrößte Eigenwert einer positiven Matrix einfach und positiv ist[14, S.664f]. Da diese Voraussetzungen nach (2.1) erfüllt sind, sind alle Voraussetzungen für Satz 2.1 erfüllt. Da A_k zudem eine symmetrische Matrix ist, folgt aus dem Spektralsatz Diagonalisierbarkeit. Wir erhalten die Konvergenzabschätzung (siehe [1, S. 273]) mit $C_\lambda > 0$

$$\left| \mu^{(j)} - \lambda_1 \right| \leq C_\lambda \left| \frac{\lambda_2}{\lambda_1} \right|^j = C_\lambda \rho^j \tag{2.9}$$

Somit liefert uns die Power-Iteration ein brauchbares Verfahren, das nach Lemma 2.1 für $k \to \infty$ gegen $a = \|A\|$ ansteigt und konvergiert. Wir wissen aufgrund des Satzes von Perron-Frobenius, dass $\rho < 1$ und somit das Verfahren konvergent ist. Leider waren mir keine analytischen Abschätzungen für den betragsmäßig zweitgrößten Eigenwert λ_2 zugänglich, womit wir eine analytische Abschätzung für ρ und somit die Anzahl der neuen richtigen Stellen pro Iteration bekommen hätten. Bei Bornemann funktioniert eine Abschätzung für λ_2 über eine Abschätzung mittels der Frobenius-Norm des unendlichdimensionalen Operators, dies erwies sich jedoch als Irrweg, da die Frobenius-Norm $\|A\|_F$ in unserem Beispiel divergiert. Wie aus Tabelle 1 ersichtlich wird, scheint $|\lambda_2| \approx 0.55$ für wachsende k zu sein. Weiters ergibt sich auch für ρ eine Tendenz, wie aus Tabelle 2 ersichtlich wird. Mittels $-\log_{10}(\rho)$ erhält man die Anzahl der gültigen Stellen, die pro Iteration bei Vernachlässigung von Rundungsfehlern stimmen sollten. Sollte tatsächlich für große k gelten, dass $\rho \approx 10^{-0.4}$ gilt, so würde durchschnittlich alle 3 Iterationen eine gültige Stelle hinzukommen. In unseren Simulationen haben wir meist um die 20 Power-Iterationen berechnet, was auf eine Gültigkeit von zumindest 8 Stellen hinweisen sollte. Leider ist das jedoch kein gültiges analytisches Argument, nachdem jedoch die Fehleranalysis versagt hat, scheint die Simulation doch zu zeigen, dass es eine Abschätzung für ρ bei großen k geben sollte.

2.3.1 Beschleunigung der Matrix-Multiplikation mit FFT

Nach der Definition von A in (2.1) erhalten die endlichdimensionalen Teilmatrizen A_k eine spezielle Struktur, welche wir zur effizienteren Berechnung der Matrix-Vektor-Multiplikationen

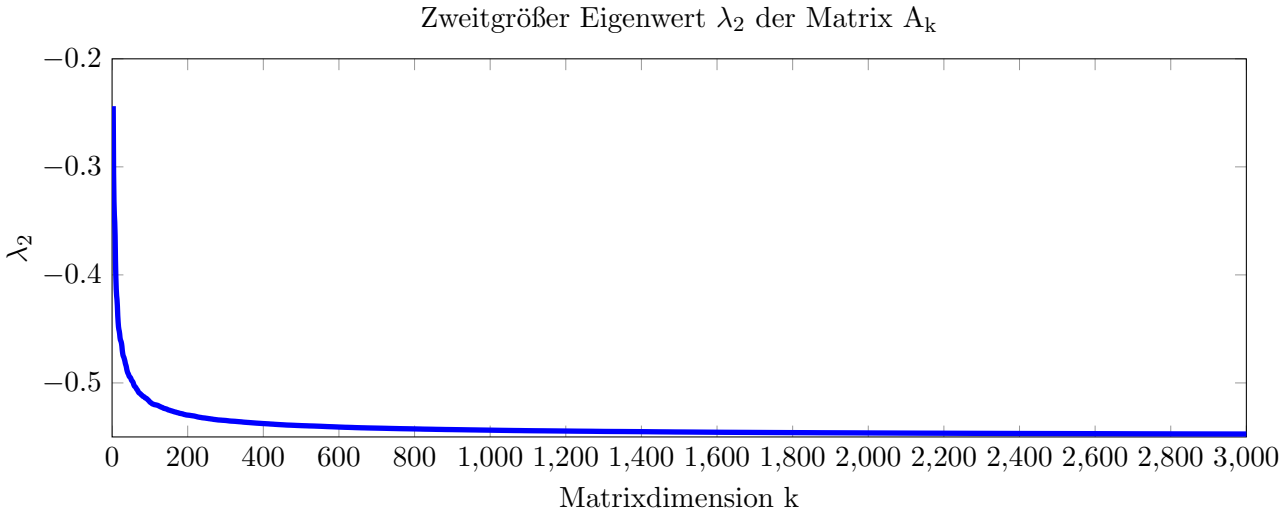


Abbildung 1: Bestimmung des betragsmäßig zweitgrößten Eigenwerts λ_2 für diverse A_k-Matrizen

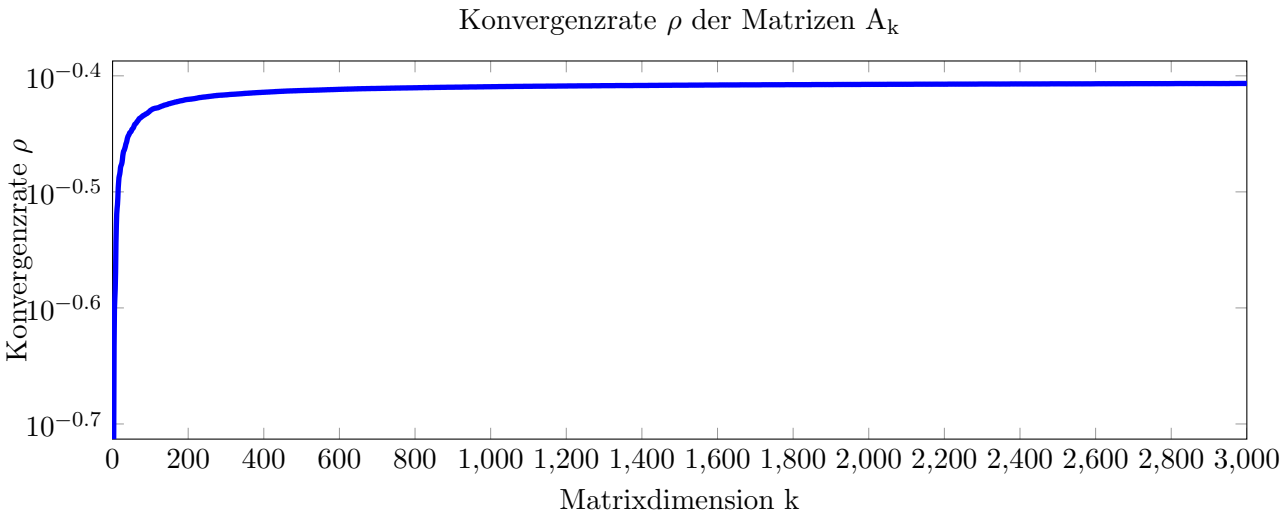


Abbildung 2: Bestimmung der Konvergenzrate $\rho = \left|\frac{\lambda_2}{\lambda_1}\right|$ für diverse A_k-Matrizen

verwenden werden. Dazu benötigen wir folgende Definitionen:

Definition 2.1 (Hankel-Matrix). *Eine (quadratische) Hankel-Matrix ist eine Matrix $A \in \mathbb{K}^{N\times N}$, für deren Einträge*

$$A_{i,j} = A_{i-1,j+1} \tag{2.10}$$

gilt, sofern $1 < i, j < N$. Insbesondere sind die Werte der von rechts oben nach links unten verlaufenden Gegendiagonalen konstant. Sie sind somit durch die oberste Zeile und die äußerste rechte Spalte vollständig beschrieben und die $N\times N$-Hankelmatrizen haben höchstens $2N-1$ verschiedene Einträge. Nummeriert man die Elemente mittels $a_0, a_1, \ldots, a_{2N-2}$ durch, so gilt

$$A_{i,j} = a_{i+j-1} \tag{2.11}$$

für $1 < i, j < N$, d.h. es ergibt sich die Struktur

$$A = \begin{pmatrix} a_0 & a_1 & \dots & a_{N-2} & a_{N-1} \\ a_1 & a_2 & a_3 & \dots & a_N \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{N-2} & a_{N-1} & & a_{2N-4} & a_{2N-3} \\ a_{N-1} & a_N & \dots & a_{2N-3} & a_{2N-2} \end{pmatrix} \tag{2.12}$$

Im unendlichdimensionalen Fall nennt man einen Operator, der (2.10) erfüllt, einen Hankel-Operator.

Anmerkung 2.1. *Nach der Definition von A mittels (2.1) erfüllt diese die Relation (2.10). A ist somit ein Hankel-Operator mit positiven Einträgen, die Submatrizen A_k sind demzufolge Hankel-Matrizen.*

Definition 2.2 (Zirkulante Matrix). *Eine zirkulante Matrix $C \in \mathbb{K}^{N\times N}$ ist eine Matrix mit N Freiheitsgraden $c_0, \dots, c_{N-1}$ $C = [c_{j-i}]_{ij}$ mit $c_k = c_{N+k}$ und $1 - N \leq k < 0$, d.h.*

$$C = \begin{pmatrix} c_0 & c_1 & \dots & c_{N-2} & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \dots & c_{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & & c_{N-1} & c_0 & c_1 \\ c_1 & c_2 & \dots & c_{N-1} & c_0 \end{pmatrix}$$

Anmerkung 2.2. *Insbesondere ist eine zirkulante Matrix eine Toeplitz-Matrix, welche die Relation*

$$A_{i,j} = A_{i+1,j+1} \tag{2.13}$$

erfüllt, d.h. in der alle Haupt- und Nebendiagonalen konstante Einträge haben. Sie wird zudem zu einer Hankel-Matrix, falls die Zeilen der Toeplitz-Matrix in umgekehrter Reihenfolge angeschrieben werden.

Unser Ziel ist nun die Berechnung der Matrix-Multiplikation mittels FFT, um den Aufwand der Berechnung der Matrixmultiplikation für die Power-Iteration beträchtlich zu verringern. Wir benötigen dazu folgenden Hilfssatz:

Satz 2.2. *Ist $C \in \mathbb{C}^{N\times N}$ eine zirkulante Matrix und F die N-dimensionale Fouriermatrix*

$$F_{j,k} = e^{\frac{-2\pi i(j-1)(k-1)}{N}} \tag{2.14}$$

mit $1 \leq j, k \leq N$, dann gilt

$$CF^* = F^*D \tag{2.15}$$

wobei D die Eigenwerte λ_p, $p = 0, \ldots, N-1$ von C enthält.

Beweis. Siehe [8, S. 413]. □

Insbesondere kann die Matrix-Vektor-Multiplikation für zirkulante Matrizen nun in $\mathcal{O}(n \log(n))$ statt in $\mathcal{O}(n^2)$ Schritten durchgeführt werden[8, S. 413]. Dieses Ergebnis ist für uns relevant, da wir aufgrund der Struktur unserer Hankel-Matrix $A \in \mathbb{R}^{n \times n}$ diese in eine zirkulante Matrix $C \in \mathbb{R}^{2n \times 2n}$ einbetten können, für welche wir dieses Ergebnis benützen können (siehe [8, S. 413ff]). Aufgrund der verbesserten Asymptotik des Aufwands wird dies für große n immer noch einen niedrigeren Aufwand zur Folge haben. Wie ersichtlich wird, hat der Algorithmus abgesehen von der verbesserten asymptotischen Laufzeit den zusätzlichen Vorteil, dass nur noch Vektoren der Länge $2n-1$ gespeichert werden müssen (entspricht der Anzahl der Freiheitsgrade einer Hankel-Matrix), aber es nicht mehr nötig ist, die ganze Matrix im Speicher zu haben. Da eine voll besetzte $n \times n$-Matrix n^2 `double`-Variablen speichern muss, ist dies ein gewaltiger Vorteil, da insbesondere sich die Verfügbarkeit von genügend Arbeitsspeicher als entscheidender Flaschenhals herausgestellt hat, weshalb insbesondere der in MATLAB vorimplementierte Schätzer für die Spektralnorm `normest()` nicht besonders zielführend war, da diesem die gesamte (vollbesetzte) Matrix übergeben werden musste. Somit nützt der FFT-Algorithmus vollends die Matrixstruktur aus.
In der konkreten Implementierung haben wir zudem noch einige Eigenschaften der Faltung mittels der Fouriertransformation benützt. Sei $\vec{d} = [0, a_0, a_1, \ldots, a_{2N-2}] \in \mathbb{R}^{2N}$ (der erste Eintrag von $\vec{d}$ wird mit 0 initialisiert, siehe dazu auch Anmerkung 2.3) der Vektor mit den Freiheitsgraden der Hankel-Matrix. Weiters seien $x = [x_0, \ldots, x_{N-1}] \in \mathbb{R}^N$ und $[y_0, \ldots, y_{N-1}] = A_N \cdot x =: y \in \mathbb{R}^N$.

$$y[i] = (A_N \cdot x)[i] = \sum_{j=0}^{N-1} d[i+j]x[j] \tag{2.16}$$

Wir definieren nun einen Hilfsvektor $\widetilde{x} = [\mathbf{0}_N, x_{N-1}, \ldots, x_0] \in \mathbb{R}^{2N}$ ($\mathbf{0}_N \in \mathbb{R}^N$ bezeichne den N-dimensionalen Nullvektor). Weiters definieren wir den Hilfsvektor $\widetilde{y} = [\ldots, y_0, \ldots, y_{N-1}] \in \mathbb{R}^{2N}$, von dem uns nur die letzten N Stellen interessieren, weil in diesen das Matrix-Vektorprodukt $A_N \cdot x = y$ stehen wird. Es gilt für $0 \leq i \leq N-1$

$$\begin{aligned} y[i] = (A_N \cdot x)[i] &= \sum_{j=0}^{N-1} d[i+j]x[j] = \sum_{j=0}^{N-1} d[i+N-1-j]\widetilde{x}[j] \\ &= \sum_{j=0}^{N-1} d[N-1-j+i]\widetilde{x}[j] = (A_N * \widetilde{x})[i] \end{aligned}$$

wobei $*$ die (diskrete) Faltung bezeichnet. Dann gilt unter Ausnützung des Faltungssatzes für die Fouriertransformation $\mathcal{F}$ (siehe [4, S. 196]) gilt

$$\widetilde{y} = \mathcal{F}^{-1}(\mathcal{F}(d) \bullet \mathcal{F}(\widetilde{x})) \tag{2.17}$$

wobei $\bullet$ die punktweise Multiplikation bezeichnet. Der Algorithmus findet sich in Listing 3 in MATLAB implementiert.

2.3.2 Implementierung in MATLAB

Numerische Berechnungen mit großen Vektoren lassen sich in MATLAB effizient implementieren. Leider ist man auf 16-stellige Genauigkeit beschränkt, daher könnten sich bei großen Dimensionen signifikante Fehler einschleichen. Deshalb wurden die Ergebnisse im nächsten Kapitel durch PARI/GP überprüft.

Listing 2: Matrixkonstruktion in MATLAB

```
function d=matrixdiag(n)
% n...Matrixdimension
% d...Eintraege der Matrix in erster Zeile und erster Spalte
    aneinandergereiht
P=isprime(2:2*n);
d=P./[1:2*n-1]+(1-P)./[1:2*n-1].^2;
end
```

MATLAB bietet in Vergleich zu PARI/GP eine vorimplementierte FFT. Daher lässt sich die Poweriteration in wenigen Zeilen programmieren.
Man beachte, dass die numerischen Eigenwerte im Laufe der Poweriteration streng monoton ansteigen. Eine gute Abbruchbedingung ergibt sich daher bei gleichbleibenden (oder durch Rundungsfehler kleiner werdenden) Eigenwerten.

Listing 3: Implementierung der Power-Iteration in MATLAB mittels FFT

```
function  y  = mypowerit(N)
%d...Spaltenvektor mit den Eintraegen von 1. Spalte
% und n-ter Zeile der Hankel-Matrix,verwende Funktion
% matrixdiag(n)
%x...Startvektor fuer Power-Iteration, sollte Naeherung fuer EV
    sein
count=0;
d=[0;matrixdiag(N)'];

fftd = fft(d);

x=[zeros(N,1);ones(N,1)];
y=x;

ew=y'*y;
ew=abs(ew/norm(y));
ewa=ew-1;
vn = y/norm(y);
while ew>ewa || count<=3
    v=ifft(fftd.*fft(y(end:-1:1))).*x;
    vn = v/norm(v);
```

```
21      ewa=ew;
22      ew=y'*vn;
23      y=vn;
24      count=count+1;
25  end
26  v=ifft(fftd.*fft(y(end:-1:1))).*x;
27  y=y'*v;
28  end
```

Anmerkung 2.3. *Die FFT funktioniert optimal für einen Input der Länge 2^k (für irgendeine natürliche Zahl k), daher haben wir bei dieser Implementierung bewusst d um eine Null verlängert. Später werden wir nämlich für $N = 2^n$ setzen, weshalb* `matrixdiag(N)` *eine Länge von $2^{2n} - 1$ aufweist.*

2.3.3 Implementierung in Pari/GP

Es stellte sich für uns die Frage, ob bei so hochdimensionalen Matrixoperationen irgendwann Rundungsfehler ins Spiel kommen bzw. ob wir mehr als Double-Genauigkeit in diesem Beispiel erreichen können. Leider stellte sich heraus, dass höhere Rechengenauigkeit im Gegensatz zum Erhöhen der Matrix-Dimension nur eine untergeordnete Rolle spielt. Wir haben dennoch die Power-Iteration mit FFT in Pari-GP implementiert, indem wir mittels Odd/Even-Rekursion FFT in Pari implementiert haben[8, vgl. S. 406ff]. Da der Programmaufruf der FFT in unser naiven Implementierung rekursiv erfolgt ist, wird das Programm bei hohen Matrixdimensionen leider eher langsam und speicherintensiv, obwohl es noch um Vielfaches schneller ist als mit einer DFT-Implementierung. Unsere Ergebnisse in Kapitel 2.3.4 zeigen jedoch, dass für alle Matrixdimensionen, für welche wir für unseren Pari/GP-Code genug Arbeitsspeicher hatten, die Ergebnisse in 50-Digit-Precision mit den Double Stellen aus MATLAB weitestgehend bis auf Double-Genauigkeit übereinstimmten.

Listing 4: Implementierung der Power-Iteration in Pari mittels Odd/Even-Rekursion-FFT

```
1  \p 50

3  divvec(A,B)=vector(#A,i,A[i]/B[i]);
4  multvec(A,B)=vector(#A,i,A[i]*B[i]);
5  skpvec(A,B)=sum(i=1,#A,A[i]*B[i]);
6  gluevec(A,B)={local(b);b=vector(#A+#B);for(i=1,#A,b[i]=A[i];);for(
       i=#A+1,#A+#B,b[i]=B[i-#A];);return(b);};

8  DFT(tau,Omega,M)={local(m,a,b,Gamma,A,B);m=M/2;Gamma=vector(M); a=
       vector(m,i,tau[i])+vector(m,i,tau[m+i]);b=multvec(vector(m,i,
       tau[i])-vector(m,i,tau[m+i]),vector(m,i,Omega^(i-1)));if(M==2,
       return(gluevec(a,b));,A=DFT(a,Omega^2,m);B=DFT(b,Omega^2,m);for
       (i=1,m,Gamma[2*i-1]=A[i]; Gamma[2*i]=B[i];);return(Gamma);)};

10 FFT(v,N)={local(Omega,c);Omega=exp(-I*2*Pi/N);c=DFT(v,Omega,N);
       return(c);}
11 IFFT(v)={return((1/#v)*conj(FFT(conj(v),#v)));};
```

```
13 matrixdiag(n)={local(P,d);P=isprime(vector(2*n-1,i,i++)); d=divvec
     (P,vector(2*n-1,i,i))+divvec((vector(2*n-1,i,1)-P),multvec(
     vector(2*n-1,i,i),vector(2*n-1,i,i))); return(d);};

15 fftmult(d,x)={local(n,t,y,v);n=(length(d)+1)/2;v=vector(2*n); for(
     i=2,n,v[i]=d[n+i-1]; v[n+i]=d[i-1];); v[1]=d[n]; v[n+1]=0; y=
     vector(2*n); for(i=1,n,y[i]=x[n+1-i];); t=IFFT(multvec(FFT(v,#v
     ),FFT(y,#y))); z=vector(n,i,t[i]); return(z);};

17 powerit(N)={d=matrixdiag(N); x=vector(N,i,1); y=x; for(i=1,15,v=
     real(fftmult(d,y)); y=1/sqrt(norml2(v))*v;p=y;); v=real(fftmult
     (d,y)); z=skpvec(p,v); return(z);};
```

Jedoch können wir diese Ergebnisse heranziehen, um somit unabhängig unsere Ergebnisse aus MATLAB überprüfen zu können. Leider war es aufgrund der Implementierung mit 8GB Arbeitsspeicher bei einer Matrixdimension von $N = 2^{20}$ nicht mehr möglich ein Ergebnis zu berechnen. Wieder erwies sich der Speicher als wesentlichste Limitierung unser Berechnungsmöglichkeiten.

2.3.4 Ergebnisse MATLAB/Pari

n	MATLAB	Pari/GP in 50 Digit Precision
2	1.22453303215513	1.22453303215512752667379116835434769358722537358 86
2^2	1.31511846909022	1.31511846909022363329048796451894872225201591436 98
2^3	1.34925308856539	1.34925308856539373857061369888875470107306742511 27
2^4	1.37641673501043	1.37641673501043131079316524763499731331939935014 48
2^5	1.38585175202704	1.38585175202703658121988762261164961376412290884 62
2^6	1.39087505720751	1.39087505720750831096731171259897763197526259953 65
2^7	1.39344719055394	1.39344719055394166949537850245317423592860396863 80
2^8	1.39467265756189	1.39467265756188714270056470825475524447733875680 65
2^9	1.39523769876373	1.39523769876372674279636064175358522685079599722 55
2^{10}	1.39551157297769	1.39551157297768745951531604583447844826468636773 65
2^{11}	1.39564589058642	1.39564589058641788663022719613342690226175691888 75
2^{12}	1.39571145671584	1.39551157297768745951531604583447844826468636773 65
2^{13}	1.39574249715665	1.39574249715663720795209726745384442823753692806 01
2^{14}	1.3957577408223	1.39575774082227385595888465231070486351948577613 71
2^{15}	1.3957650519686	1.39576505196856396922334148260960379016314312385 21
2^{16}	1.3957686123913	1.39576861239130078508410040168913292841493618820 15
2^{17}	1.39577034146014	1.39577034146011552609250916810743844723200041184 65
2^{18}	1.39577117882631	1.39577117882626398132382688396342222135018574079 19
2^{19}	1.39577158704458	1.39577158704452121985111976645250118857207745894 96
2^{20}	1.39577178510669	
2^{21}	1.39577188174183	
2^{22}	1.39577192879746	
2^{23}	1.3957719517462	
2^{24}	1.39577196295994	
2^{25}	1.39577196845025	
2^{26}	1.395771972126048	

Diese Ergebnisse legen nahe, dass die Stellen 1.39577 signifikant sind.

Mit dem Aitken'schen Δ^2 - Verfahren (siehe [1, S.177]) soll die Folge extrapoliert werden um eine weitere Tendenz der Stellen auszumachen.
Wie aus der Tabelle ersichtlich sichert diese Extrapolation keine weiteren Stellen, da immer noch eine 9 nach der dadurch potenziell falschen 1 befindet.

n	Ohne Extrapolation	Aitken Extrapolation
2^1	1.224533032155130	
2^2	1.315118469090220	
2^3	1.349253088565390	1.369893573489136
2^4	1.376416735010430	1.482264754361809
2^5	1.385851752027040	1.390872983164116
2^6	1.390875057207510	1.396594740603290
2^7	1.393447190553940	1.396146254675003
2^8	1.394672657561890	1.395787833121464
2^9	1.395237698763730	1.395721131659331
2^{10}	1.395511572977690	1.395769181457340
2^{11}	1.395645890586420	1.395775165872138
2^{12}	1.395711456715840	1.395773985076979
2^{13}	1.395742497156650	1.395770404173266
2^{14}	1.395757740822300	1.395772450745059
2^{15}	1.395765051968600	1.395771790415474
2^{16}	1.395768612391300	1.395771992168419
2^{17}	1.395770341460140	1.395771973956796
2^{18}	1.395771178826310	1.395771965167012
2^{19}	1.395771587044580	1.395771975353962
2^{20}	1.395771785106690	1.395771971770738
2^{21}	1.395771881741830	1.395771973811526
2^{22}	1.395771928797460	1.395771973457689
2^{23}	1.395771951746200	1.395771973592430
2^{24}	1.395771962959940	1.395771973675575
2^{25}	1.395771968450250	1.395771973716935
2^{26}	1.395771972126048	1.395771979572397

3 Aufgabe 3

3.1 Problembeschreibung

Bei dieser Aufgabe ist das exponentielle Wachstum einer stochastischen Folge zu betrachten. Genauer sei (x_n) definiert durch $x_0 = x_1 = x_2 = 1$ und

$$x_{n+3} = x_{n+2} \pm \frac{1}{3}(x_{n+1} + x_n) \quad n \in \mathbb{N}^+ \tag{3.1}$$

, wobei das Vorzeichen mit Wahrscheinlichkeit $\frac{1}{2}$ positiv oder negativ gewählt wird. Zu untersuchen ist das Konvergenzverhalten von $\lim_{n\to\infty} \sqrt[n]{|x_n|} =: \sigma$. Sei zusätzlich $\sigma_n := \sqrt[n]{|x_n|}$.

3.2 Erste Schritte

Zunächst haben wir direkt begonnen derartige stochastische Folgen in R zu implementieren (siehe Listing 5). Zu diesem Zeitpunkt war uns noch nicht klar, ob es sich bei σ um eine Zahl, oder tatsächlich um eine Verteilung von Zahlen - wie die σ_n - handelt. Erst eine theoretische Abhandlung zeigt, dass jede Folge (σ_n) fast sicher konvergiert.

Listing 5: Naive Monte Carlo Simulation in R

```
n <- 30000
it <- 1500
y <-array(0,dim=4)
x <- array(0,dim=it)
y[1] <- y[2] <- y[3] <- 1
for(i in 1:it)
{
  d <- sample(c(1,-1),n, replace=T)
  d <- d*2-1
  y <- rep(1,4)
  for(j in 4:n)
  {
  y[4] <- y[3] + d[j]/3*(y[2]+y[1])
  y[1]<-y[2]
  y[2]<-y[3]
  y[3]<-y[4]
  }
  x[i]=y[4]
}

mean(abs(x)^(1/n))
var(abs(x)^(1/n))
max(abs(x)^(1/n)) - min(abs(x)^(1/n))
```

Ziel dieser frühen Simulationen war es, die Verteilung von σ_n für große n zu analysieren. Bereits eine relative kleine Stichproben aus σ_{10000} im Umfang von 100 Simulationen ergibt überraschend kleine Streuungsparameter. So hatten haben wir Stichprobenvarianzen im Bereich 10^{-5} und sogar Spannweiten der Größenordnung 10^{-2}. Das lässt vermuten, dass (σ_n) tatsächlich stets gegen den selben Grenzwert konvergiert und bereits wenige Monte-Carlo Simulationen gute Näherungswerte liefern können, solange n groß genug gewählt wird. Hier jedoch scheitert der

naive Ansatz. Für $n = 10^5$ werden bereits die Werte x_n so groß, dass sie einen Überlauf erzeugen. Wir haben später zwei verbesserte Varianten entwickelt (siehe Abschnitt 3.9), die dieses Problem geschickt umgehen. Tatsächlich werden derartige Monte-Carlo Simulationen letztendlich unsere Ergebnisse liefern. Sehr ernüchternd, aber es geht scheinbar nicht besser.
Eine weitere wichtige Frage, ist die Abhängigkeit der Startwerte auf den Grenzwert. Offensichtlich beeinflusst eine Streckung diesen nicht. Setzten wir nämlich $\tilde{x}_0 = \tilde{x}_1 = \tilde{x}_2 = t$ und definieren die Folge $(\tilde{x}_n)$ über die Rekursion (3.1), so ist die Zufallsvariable $\tilde{x}_n$ genau $t \cdot x_n$. Für entsprechende $\tilde{\sigma}_n$ gilt der Zusammenhang $\tilde{\sigma}_n = t^{\frac{1}{n}} \cdot \sigma_n$. Im Grenzfall gilt nun aber $t^{\frac{1}{n}} \to 1$.
Auch folgende Überlegung weißt auf eine Unabhängigkeit der Startwerte hin. Dröselt man eine gewöhnliche lineare Rekursion der Form

$$a_{n+l} = \sum_{i=0}^{l-1} c_i \cdot a_{n+i}$$

auf, so erhält man im allgemeinen Fall

$$a_n = \sum_{i=0}^{l-1} A_i \cdot \lambda_i^n$$

, wobei $|\lambda_0| < |\lambda_1| < \cdots < |\lambda_{l-1}|$ die der Betragsgröße nach geordneten Nullstellen[2] des charakteristischen Polynoms $\lambda^l - \sum_{i=0}^{l-1} c_i \lambda^i$ seien. Die Koeffizienten A_i hängen nur von den gegebenen Startwerten $a_0, \ldots, a_{l-1}$ ab. Im Grenzfall gilt daher $\sqrt[n]{|a_n|} \to |\lambda_{l-1}|$, solange zumindest einer der Startwerte $\neq 0$ ist. Das alles suggeriert bereits den Satz 3.4.
Wir befassen uns nun wieder mit der ursprünglichen Aufgabe. Über derartige Folgen gibt es relativ wenige Resultate. Eine wichtige Arbeit dazu ist jedoch von Viswanath [19], welche sich mit der „Random-Fibonacci-Sequence“ befasst. Ein wesentlicher Schritt ist hier die Umformulierung auf ein Problem mit dem Produkt zufällig gewählter Matrizen.

3.3 Umformulieren auf ein Matrizenproblem

Wir betrachten nun den Vektor $\begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix}$. Es gilt nach Einsetzen der Rekursionsformel (3.1)

$$\begin{pmatrix} x_{n+1} \\ x_{n+2} \\ x_{n+3} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \pm\frac{1}{3} & \pm\frac{1}{3} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_n \\ x_{n+1} \\ x_{n+2} \end{pmatrix}$$

Sei (T_i) eine Folge zufälliger Matrizen, welche gleichverteilt aus $\{A, B\}$, mit

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{3} & -\frac{1}{3} & 1 \end{pmatrix} \quad t_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad G_n := \prod_{i=1}^{n} T_i$$

gewählt wird. Die ursprüngliche Rekursion kann also durch $x_n = \pi_1(G_n \cdot t_0)$ gelöst (wenn die Verteilung der G_n bekannt ist) werden.
Wir betrachten aber nun ein verändertes Problem: Bestimme

$$\gamma := \lim_{n \to \infty} \frac{\log(\|G_n\|)}{n} \tag{3.2}$$

[2] Wir beschränken uns auf den Fall, dass diese Verschieden sind

γ nennt man den Lyapunov Exponenten. Dabei ist $||M||$ eine Matrixnorm, also eine Norm in $\mathbb{R}^{3\times 3}$, die zusätzlich

$$\|M \cdot N\| \leq \|M\| \cdot \|N\| \tag{3.3}$$

erfüllt.
Zusätzlich definieren wir analog zu oben $\gamma_n := \frac{\log(\|G_n\|)}{n}$.

> **Anmerkung 3.1.** *Da in $\mathbb{R}^n$ alle Normen äquivalent sind, ist dieser Limes unabhängig von der tatsächlich gewählten Norm. Sehr wohl spielt die Wahl der Norm für γ_n eine wesentliche Rolle. Man vergleiche dazu mit unseren anfänglichen Überlegungen zur skalierten Folge $(\tilde{x}_n)$ In diesem Abschnitt sei aber üblicherweise $\|x\|$ die euklidische Norm und $\|M\| := \sup_{x\in\mathbb{R}^3\setminus\{0\}} \frac{\|Mx\|}{\|x\|}$.*

Da offensichtlich $\|A\| = \|B\| =: \chi \approx 1.473446150061121$, erhalten wir mit (3.3) bereits eine erste Abschätzung für γ

$$\gamma \leq \lim_{n\to\infty} \frac{\log(\prod_{i=1}^n \|T_i\|)}{n} = \lim_{n\to\infty} \frac{\log(\chi^n)}{n} = \log(\chi) \approx 0.387603976929652 \tag{3.4}$$

Die Sinnhaftigkeit dieser Umformulierung steht jedoch noch etwas im Raum. Erst die zentrale Aussage $e^\gamma = \sigma$ des nächsten Abschnitts rechtfertigt das neue Problem.

3.4 Konvergenzaussagen und Lyapunov Exponenten

Eine der ersten Publikationen über das Konvergenzverhalten von derartigen Matrizenprodukten geht auf Frustenberg und Kesten [6] zurück. Weitere relevante Sätze, welche wir ohne Beweis zitieren möchten, stammen aus [3, 18].
Wir betrachten nun ganz allgemein einen Stochastischen Prozess $G_n := \prod_{i=1}^n T_i$, wobei T_i unabhängig identisch verteilte Zufallsmatrizen aus $\mathbb{R}^{d\times d}$ mit Wahrscheinlichkeitsverteilung μ seien. Zusätzlich setzten wir Voraus, dass nur invertierbare Matrizen angenommen werden dürfen.

> **Anmerkung 3.2.** *Die letzte Voraussetzung ist nicht für alle folgenden Sätze notwendig, vereinfacht aber die Notation.*

Der erste wichtige Satz zeigt uns, dass γ aus (3.2) existiert. Sei für eine reellwertige Funktion f der positive Teil definiert als: $f^+(x) := \max\{f(x), 0\}$

Satz 3.1. *Sei $\log(\mathbb{E}(T_1)) < \infty$, so existiert eine Konstante $\gamma < \infty$ mit*

$$\lim_{n\to\infty} \frac{\log(\|G_n\|)}{n} = \gamma$$

fast sicher. γ wird als Lyapunov Exponent von μ bezeichnet.

Unser Grenzwert im veränderten Problem existiert also mit Wahrscheinlichkeit 1.
Wir benötigen nun einige Begriffe:

Definition 3.1. *Sei S eine Menge invertierbarer Matrizen aus $\mathbb{R}^{d\times d}$.*

(i) S heißt irreduzibel, wenn es keinen linearen Unterraum $V \neq \{0\}$ von $\mathbb{R}^d$ gibt, sodass $M(V) = V$ für alle $M \in S$ gilt.

(ii) S heißt stark irreduzibel, wenn es keine endliche Familie $V_1, \dots, V_n$ aus linearen Unterräumen von $\mathbb{R}^d$ gibt, sodass für alle $M \in S$ gilt:

$$M(V_1 \cup V_2 \cup \cdots \cup V_n) = V_1 \cup V_2 \cup \cdots \cup V_n$$

(iii) Für die Wahrscheinlichkeitsverteilung μ im Raum der Matrizen, bezeichne $\mathcal{G}_\mu$ die kleinste Untergruppe von $\mathbb{R}^{d\times d}$, die den Träger von μ enthält.

Anmerkung 3.3. *$\mathcal{G}_\mu$ ist offenischtlich genau dann stark irreduzibel, wenn der Träger von μ es ist.*

Satz 3.2. *Sie μ eine Wahrscheinlichkeitsverteilung auf $\left\{M \in \mathbb{R}^{d\times d}, |\det M| = 1\right\}$, mit $\mathcal{G}_\mu$ stark irreduzibel. Ist $\mathcal{G}_\mu$ nicht kompakt, so folgt $\gamma > 0$*

Satz 3.3. *Sei $\mathcal{G}_\mu$ irreduzibel. Gelte weiters $\mathbb{E}(\log^+(\|T_1\|)) < \infty$ und $\mathbb{E}(\log^+(\|T_1^{-1}\|)) < \infty$, so gilt*

$$\inf_{x\in\mathbb{R}^d} \frac{1}{n}\mathbb{E}\log\frac{\|G_n x\|}{\|x\|} \leq \gamma \leq \sup_{x\in\mathbb{R}^d} \frac{1}{n}\mathbb{E}\log\frac{\|G_n x\|}{\|x\|}$$

Wobei die beiden Schranken gegen γ konvergieren, für $n \to \infty$

Satz 3.4. *Sei $\mathcal{G}_\mu$ stark irreduzibel und $\mathcal{G}_\mu$ enthalte eine Folge von Matrizen, die gegen eine mit Rang 1 konvergiert. Gelte zusätzlich $\mathbb{E}(e^{\tau\log^+(\|T_1\|)+\log^+(\|T_1^{-1}\|)}) < \infty$ für ein $\tau > 0$. So gilt für alle $x, y \neq 0$*

$$\lim_{n\to\infty} \frac{1}{n}|y^T G_n x| = \gamma$$

fast sicher.

Anmerkung 3.4. *Da wir als μ eine Gleichverteilung auf $\{A, B\}$ gewählt haben, $\det(A) = \det(B) = \frac{1}{3}$, A und B verschiedene Eigenvektoren haben und die Matrizen $A, A^2, A^3, \dots$ und $B, B^2, B^3, \dots$ alle verschieden[3], aber invertierbar sind, ist es leicht, alle relevanten Voraussetzungen zu überprüfen. Außerdem gilt $\lim_{n\to\infty}(AB)^n \to \begin{pmatrix} -0.5 & 0 & 1.5 \\ -0.25 & 0 & 0.75 \\ -0.5 & 0 & 1.5 \end{pmatrix}$. Insbesondere*

ist somit Satz 3.4 anwendbar. Wir können also alle zitierten Sätze ohne große Überlegung anwenden.

Also folgt mit Satz 3.4 mit $x = t_0$ und $y = (1, 0, 0)^T$ für unser Problem:

$$e^{\gamma} = \sigma$$

Insbesondere ist damit auch die fast sichere Konvergenz von σ_n bewiesen.
Ein weiterer wichtiger Satz von Frustenberg, erlaubt es das Problem auf ein analytisches Problem zu reduzieren:
Wir haben bereits gesehen, dass eine Skalierung das Problem nicht verändert, weshalb es Sinn macht, den Projektiven Raum $P(\mathbb{R}^d)$ zu betrachten. Für $x \in \mathbb{R}^d$ bezeichne $\bar{x} = \lambda \cdot x, \lambda \in \mathbb{R}$, jenes Element aus $P(\mathbb{R}^d)$, welches in Richtung x geht.

Definition 3.2. *Sei ν nun eine Wahrscheinlichkeitsverteilung auf $P(\mathbb{R}^d)$, so wird durch*

$$\mu * \nu(A) := \int \int \mathbb{1}_A(\bar{x}) d\mu(M) d\nu(\bar{x})$$

Ein Maß auf den Borellmengen auf $P(\mathbb{R}^d)$ definiert.
ν heißt μ-invariant, wenn $\mu * \nu$

Satz 3.5 (Frustenberg). *Sei $\mathcal{G}_\mu$ stark irreduzibel und $\mathbb{E}(\log^+(\|T_1\|)) < \infty$, dann gilt für alle μ-invarianten ν*

$$\gamma = \int \int \log \frac{\|Mx\|}{\|x\|} d\mu(M) d\nu(\bar{x}) \tag{3.5}$$

Man kann also die Aufgabe so umformulieren:
Finde ein μ-invariantes Maß auf $P(\mathbb{R}^3)$ und berechne das Integral (3.5).

3.5 Die ersten Schranken

Wir haben bereits in (3.4) eine obere Schranke für γ erhalten. Wie schon angemerkt ist, gilt $\det(A) = \det(B) = \frac{1}{3}$ Skalieren wir die Matrizen A und B um den Faktor $\sqrt[3]{3}$, so ist Satz 3.2 anwendbar. Diese Skalierung ergibt uns:

$$\gamma = \lim_{n\to\infty} \frac{\log(\|\sqrt[3]{3}^n G_n\| \cdot \sqrt[3]{3}^{-n})}{n} = \lim_{n\to\infty} \frac{\log(\|\sqrt[3]{3}^n G_n\|)}{n} - \log(\sqrt[3]{3}) \geq -\log(\sqrt[3]{3})$$

Damit erhlaten wir also eine erste Abschätzung für den eigentlich gesuchten Wert σ

$$0.69 < \frac{1}{\sqrt[3]{3}} \leq \sigma \leq \chi < 1.48$$

3.6 Frust mit Frustenberg

Wie wir eben auch gesehen (Satz 3.5) haben, reicht es ein passendes μ-invariantes Maß ν zu konstruieren. Viswanath verfolgt tatsächlich diese Strategie bei der Untersuchung der zufälligen Fibonacci Zahlen, die wir gerne Adaptieren möchten. An einer Stelle geht jedoch wesentlich ein, dass bei den Fibonacci Zahlen, das Problem eine Dimension weniger hat. So findet er folgende äquivalente Formulierungen

$$f_{n+2} := \pm f_{n+1} + f_n$$
$$\frac{f_{n+2}}{f_{n+1}} = \pm 1 + \frac{f_n}{f_{n+1}}$$

und interpretiert das als einen Random Walk über Steigungen $m \to \pm 1 + \frac{1}{m}$ Diese vielen verschiedenen Sichtweisen auf das selbe Problem erlauben ihm eine sehr elegante Konstruktion des Maßes ν.
Andere Paper verwenden zusätzliche Annahmen, die wir nicht erfüllen können. Zum Beispiel fordert [16, Assumption 1], dass alle Matrixeinträge positiv sein müssen. Also wieder kein Erfolg. Lange Gedankenketten, ob man nicht doch anders ein entsprechendes Maß konstruieren kann enden ziellos. Wir ärgern uns also schlichtweg über die zusätzliche Dimension und stehen nun vor drei Möglichkeiten: Entweder wir versuchen durch geschickte Berechnungen $\mathbb{E}(\gamma_n)$ für kleine n exakt zu bestimmen, oder für große n näherungsweise durch Monte-Carlo-Simmlationen. Oder aber, wir verändern die Angabe.
Wir entscheiden uns zunächst für letzteren Ansatz.

3.7 Wir schneiden eine Dimension ab

Frustriert über Frustenberg und über die zusätzliche Dimension entscheiden wir uns also, diese einfach wegzulassen und die Rekursion

$$x_{n+2} = x_{n+1} \pm \frac{1}{3} x_n \quad n \in \mathbb{N}^+$$

zu betrachten. Auch wenn es dafür eine Anleitung gibt! - oder gerade weil es dafür eine gibt?
Wie ganz am Anfang werfen wir eine kurze Simulation an und sind doch überrascht. Bereits die Folge x_n konvergiert und zwar nach 0. Anstatt also exponentiell zu wachsen, fällt diese Folge. Das entscheidende ist wohl der Faktor $\frac{1}{3}$.
Wir fragen uns also für welche β die Folge

$$x_{n+2} = x_{n+1} \pm \beta x_n \quad n \in \mathbb{N}^+$$

exponentiell wächst, also $\sigma > 1$ ist. Eine kurze Recherche beantwortet aber sofort diese Frage. Die Grenze liegt bei $\beta^* \approx 0.70258$. Für $\beta < \beta^*$ nimmt die Folge exponentiell ab, für β über dieser Schranke wächst sie exponentiell. Embree und Trefethen [5] sprechen im ersten Fall auch davon, dass die Folge stabil ist. Diese betrachten allgemein Folgen der Bauart $x_{n+2} = \alpha x_{n+1} \pm \beta x_n$ und bestimmen die stabilen bzw. instabilen Bereiche (α, β). Abbildung 3 visualisiert deren Ergebnisse.
Nach diesem kurzen Ausflug kehren wir doch wieder zur ursprünglichen Angabe zurück und gehen die anderen beiden Wege.

3.8 Zyklendarstellung von $\mathbb{E}(\gamma_n)$

Sei $\{A, B\}^n$ die Menge aller endlichen Folgen der Länge n und $M_n := \{G : G = \prod_{i=1}^n X_i, X \in \{A, B\}^n\}$. Das ist also die Menge aller möglichen G_n.

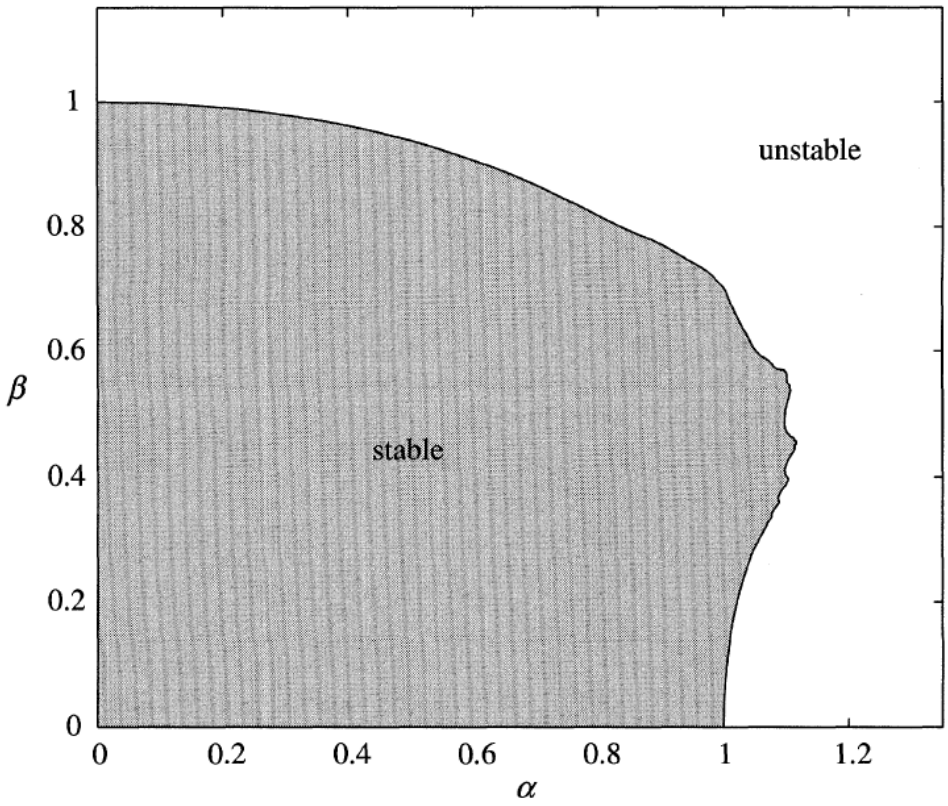


Abbildung 3: Stabile und instabile Bereiche in Abhängigkeit von α und β (aus [5, Fig. 6])

Wir versuchen nun

$$\mathbb{E}(\gamma_n) = \frac{1}{n \cdot 2^n} \sum_{G \in M_n} \log(\|G\|) \tag{3.6}$$

für kleine n exakt zu berechnen. Offenbar ist das Problem, die Mächtigkeit von M_n und die aufwändige Normbestimmung. Daher ist eine Matrixnorm, die vielen G_n den selben Wert zuweist vorteilhaft.
Die Idealvorstellung ist eine Norm, die

$$\|\prod_{i=1}^{n} X_i\| = \|\prod_{i=1}^{n} X_{\pi(i)}\|$$

für alle $X \in \{A, B\}^n$ und Permutationen $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ erfüllt.
Eine derartige Norm wird es jedoch nicht geben. Mainieri [12, 13] und Nielsen [15] verwendet stattdessen ein Gewicht, welches die Normen hinreichend gut approximiert und ähnliche praktische Eigenschaften hat.

Anmerkung 3.5. *Während Mainieri lange umständliche Rechnungen macht und teilweise etwas unverständlich argumentiert um* (3.6) *in eine günstige Form zu bringen, fasst Nielsen zunächst Mainieris Arbeit zusammen und findet eine praktischere Formel mit einem zweizeiler. Er bemerkt fast einwenig spöttisch: „The derivation of the formula is almost trivial“. Wir verfolgen also seinen Ansatz weiter.*

Wir definieren zunächst:

Definition 3.3. *Sei $G \in \mathbb{R}^{d\times d(3\times 3)}$. Sofern der nachfolgende Limes existiert, gelte*

$$|G| := \lim_{n\to\infty} |tr(G^n)|^{\frac{1}{n}}$$

Dieses Gewicht ist zyklisch und multiplikativ in folgendem Sinn:

Satz 3.6. *Seien $G, G_1, G_2 \in \mathbb{R}^{d\times d}$*

(i) Existiert $|G_1G_2|$, so existiert auch $|G_2G_1|$ und es gilt Gleichheit.

(ii) Es gilt $|G^p| = |G|^p$, sofern diese Grenzwerte existieren

(iii) Existiert $|G|$, so gilt $|G| = \rho(G)$, wobei $\rho(G)$ der Spektralradius von G ist.

Beweis. (i) Da $\mathrm{tr}(G_1G_2) = \mathrm{tr}(G_2G_1)$, folgt sofort die Aussage.

(ii) $|G^p| = \lim_{n\to\infty} |\mathrm{tr}(G^{pn})|^{\frac{1}{n}} \overset{pn=:m}{=} \lim_{m\to\infty} |\mathrm{tr}(G^m)|^{\frac{p}{m}} = |G|^p$

(iii) Siehe [15, Bemerkungen 3.4] □

Anmerkung 3.6. *Aus der Literatur wird nicht klar, wann genau $|G|$ existiert. Es scheint kurz ein Wenig deprimierend, dass Nielsen schreibt, dass der Limes nicht existieren muss, wenn G verschiedene Eigenwerte λ_1, λ_2 hat, mit $|\lambda_1| = |\lambda_2| = \rho(G)$. Insbesondere ist das ja der Fall, wenn die betragsgrößten Eigenwerte konjugiert komplex zueinander sind. In allen anderen Fällen jedoch kann $|G|$ bestimmt werden.*
Leider fällt unser B in die „zweifelhafte“ Gruppe. Da jedoch durch B eine lineare Rekursion beschrieben wird, ist es leicht möglich B^n zu bestimmen und manuell zu überprüfen, dass $|B|$ tatsächlich existiert. Dabei ist insbesondere der Zusammenhang des charakteristischen Polynoms zur Bestimmung der Eigenwerte der Matrix B und dem charakteristischen Polynom der Rekursion wesentlich.

Die Rechtfertigung dieses Gewicht, anstatt einer Matrixnorm zu verwenden, kommt von wesentlichen Eigenschaften des Spektralradius.

Satz 3.7. *Seien $G, G_1, G_2 \in R^{d\times d}$*

(i) Ist $\|\cdot\|$ eine Matrixnorm, so gilt für alle Matrizen G:

$$\rho(G) \leq \|G\|$$

(ii) Für alle Matrizen G und alle $\varepsilon > 0$ existiert eine Matrixnorm $\|\cdot\|$ mit

$$\|G\| \leq \rho(G) + \varepsilon$$

Beweis. Siehe [1, S. 36f] □

Damit erhalten wir [4]

$$\mathbb{E}(\gamma_n) = \frac{1}{n \cdot 2^n} \sum_{G \in M_n} \log(|G|)$$

Mit Satz 3.6 haben wir nun ein Werkzeug, die Anzahl der Summanden drastisch zu reduzieren: Es wird nämlich nur noch notwendig sein $|G|$ für jene G zu bestimmen, welche sich nicht als Potenz D^m, $D \in M_{n/m}$ schreiben lassen, wir nennen derartige Matrixprodukte prim. Da außerdem $|X_1 \cdot \ldots \cdot X_n| = |X_n \cdot X_1 \cdot \ldots \cdot X_{n-1}|$, ist es sinnvoll Produkte zu Äquivalzenklassen zusammenzufügen, welche sich durch zyklische Vertauschungen ergeben. Von jeder Klasse muss nur von einem Repräsentanten die $|\cdot|$ berechnet werden.
Daher definieren wir:

Definition 3.4. *(i) Unter einem Zyklus verstehen wir eine der obigen Äquivalenzklasse.*

(ii) Ein Zyklus heißt prim, oder Prim-Zyklus, wenn alle Elemente prim sind.

(iii) P_n sei die Menger aller primen Zyklen aus M_n.

Anmerkung 3.7. *Ein Zyklus ist offensichtlich genau dann prim, wenn ein Repräsentat prim ist. Außerdem ist die Mächtigkeit eines Prim-Zyklus, genau die Länge der einzelnen Elemente, da dies die Anzahl der unterschiedlichen zyklischen Vertauschungen ist.*

Wir befassen uns nun mit

$$\frac{1}{n} \sum_{G \in M_n} \log(|G|)$$

Wie wir gerade gesehen haben, ist jedes $G \in M_n$ entweder prim, oder Potenz eines kleineren primen Matrizenproduktes. Zusammen mit den Überlegungen aus Anmerkung 3.7 erhalten wir.

$$\frac{1}{n} \sum_{G \in M_n} \log(|G|) = \frac{1}{n} \sum_{d|n} \sum_{G_d \in P_d} d \log(|G_d^{n/d}|) = \sum_{d|n} \sum_{G_d \in P_d} \log(|G_d|)$$

Wir erhalten somit das Hauptresulatat dieses Abschnitts:

Korollar 3.1. *Mit den obigen Definitionen gilt:*

$$\mathbb{E}(\gamma_n) = \frac{1}{2^n} \sum_{d|n} \sum_{G \in P_d} \log(|G|) \tag{3.7}$$

Diese Darstellung heißt Zyklendarstellung.

[4] Wie wir bereits Besprochen haben, ist γ_n durchaus Abhängig von der Norm, weshalb ein „=“ an dieser Stelle vielleicht Irreführend sein kann. Da es jedoch letztendlich nur um den Grenzwert $n \to \infty$ geht, ist diese Notation gerechtfertigt.

Um diese Formel nun zu implementieren, benötigen wir eine effiziente Methode, die Primzyklen zu finden. Wir haben dafür einen eigenen Algorithmus entwickelt. Wir stellen zunächst fest, dass natürliche Zahlen als Matrizenprodukte, vermöge der folgenden Abbildung, interpretiert werden können.

Definition 3.5. *Sei $(d_0, d_1, \ldots, d_k)_2$ die eindeutige Binärdarstellung[5] der natürlichen Zahl $\sum_{i=0}^{k} d_i 2^i$. Dann definieren wir*

$$\iota : \begin{cases} \mathbb{N} \to \bigcup_{i\in\mathbb{N}} M_i \\ (d_0, d_1, \ldots, d_k)_2 \mapsto \prod_{i=0}^{k} (d_i \cdot A + (1 - d_i) \cdot B) \end{cases}$$

Anmerkung 3.8. *Diese Funktion multipliziert also für jede Eins mit A und jede Null mit B und ist offensichtlich injektiv. Zum Beispiel ist $\iota(6) = \iota((0, 1, 1)_2) = B \cdot A \cdot A$.*
Umgekehrt hat aber nicht jedes Matrizenprodukt eine Darstellung als natürliche Zahl. Zum Beispiel $A{\cdot}A{\cdot}B$. Hier müssten wir nämlich von $\iota((1, 1, 0)_2) = \iota((1, 1)_2) = A{\cdot}A$ ausgehen. Sehr wohl hat aber jeder Primzyklus einen Repräsentanten, der eine entsprechende Darstellung besitzt. Denn entweder gibt es einen Repräsentanten, der mit A endet, oder es handelt sich um den einzigen Primzykuls, der kein A enthält. Das ist $B = \iota(0)$.

Es ist möglich, eine eindeutigen natürlich Darstellung für Primzyklen G_p festzulegen, indem wir das Maximum $\max_{G\in G_p} \iota^{-1}(G)$ wählen. In diesem Sinn definieren wir $\mathcal{N}$ als die Menge aller natürlichen Zahlen, die einer derartigen eindeutigen Darstellung entsprechen.
Für eine natürliche Zahl n können wir eine sehr einfach und effizienten Algorithmus angeben, der bestimmt, ob $n \in \mathcal{N}$

Data: Natürliche Zahl n in Binärdarstellung $(d_0, \ldots, d_l)_2$
for *i = 1,2,...,l* **do**
  $m = (d_i, d_{i+1}, \ldots, d_l, d_0, d_1, \ldots, d_{i-1})_2$;
  if $m \geq n$ **then**
    return *false*;
  end
end
return *true*;
Result: true, wenn $n \in \mathcal{N}$. false, sonst

Algorithm 2: Pseudocode für Entscheidungsfunktion $n \in \mathcal{N}$.

Der Algorithmus erzeugt also die möglichen natürlichen Darstellungen des Zyklus $\iota(n)$ durch zyklische Vertauschungen der Stellen in der Binärdarstellung. In Matlab lässt sich das effizient mit `bitshift(...)` implementieren.

Satz 3.8. *Der Algorithmus 2 ist korrekt.*

Beweis. Offensichtlich liefert der Algorithmus `false`, wenn $n \neq \max_{G\in\iota(n)} \iota^{-1}(G)$, da sonst ein größeres m gefunden wird.

n	$e^{\mathbb{E}(\gamma_n)}$	CPU time [s]
1	1.143845296172977	0.007683
2	1.069507034185833	0.008614
3	1.070123471345550	0.007196
4	1.039225287707814	0.008726
5	1.001620813416644	0.012709
6	1.023065041794183	0.018103
7	1.020741675222189	0.028551
8	1.019792285658657	0.041830
9	1.018546114393850	0.073944
10	1.008606392921484	0.149561
11	1.011402448497379	0.310221
12	1.011485539289897	0.611023
13	1.010631144427000	1.251223
14	1.010789156079386	2.434384
15	1.011360362613062	5.008696
16	1.010082349433776	10.346435
17	1.010151822089354	21.425144
18	1.010032916974997	43.459323
19	1.009849806255128	91.848916
20	1.009671219435144	187.608668
21	1.009610616601558	399.804437
22	1.009478201073392	836.763051
23	1.009434449253024	1682.990211
24	1.009381287588369	3336.490609
25	1.009343803184316	7192.623995

Tabelle 1: Ergebnisse mittels Zyklendarstellung

Es bleibt zu zeigen, dass auch bei nicht primen Zyklen `false` ausgegeben wird. Ist $\iota(n)$ nicht prim, so gibt es eine Periode $d_0, \ldots, d_{p-1}$, $p|l+1$ in der Binärdarstellung. Dann gilt offensichtlich $m = (d_p, d_{p+1}, \ldots, d_l, d_0, d_1, \ldots, d_{p-1})_2 = n$ und der Algoirthmus terminiert mit `false`. Sei umgekehrt für ein kleinstes i die Bedingung $m = n$ erfüllt, so gilt es also für die zyklische Vertauschung $\delta_i : d_j \mapsto d_{j+i}$. $\delta_i = \delta_0$. Da $(\delta_1^s)_{s=0,\ldots,l}$ eine Untergruppe der Gruppe aller zyklischen Vertauschungen ist, ist die Gruppenordnung ein Teiler von $l+1$. Somit gilt also $i|l+1$, womit eine Periode gefunden ist. □

Um nun alle primzyklen-Darstellungen für Primzyklen der länge n zu finden, muss nun einfach jede natürliche im Intervall $[2^{n-1}, 2^n - 1]$ mit dem obigen Algotithmus getestet werden.
Jetzt können wir alles in Matlab programmieren und bekommen die Ergebnisse aus Tabelle 1. Die wichtigsten Programmteile sind in Listing 6.

Listing 6: Wesentliche Implementierungsteile der Zyklendarstellung

```
function l = binLen(n)
  if n <= 1
    l = 1;
  else
    l = binLen(bitshift(n,-1))+1;
  end
end

function y = lnsp(A)
  y = log(max(abs(eigs(A))));
end



function b = isPrimeRepr(n)
  len = binLen(n);
  b = true;
  lb = bitshift(1,len-1);
  n_ = n;
  for i = 1:(len-1)
    if mod(n_,2) == 1
      n_ = bitshift(n_,-1)+lb;
    else
      n_ = bitshift(n_,-1);
    end
    if n_ >= n
      b = false;
      return
    end
  end
end

function pc = getAllPrimeCycles(n)
  pc = [0,1];
  for i=2:n
    if mod(n,i) == 0
      for j = bitshift(1,i-1):(bitshift(1,i)-1)
        if isPrimeRepr(j)
          pc = [pc,j];
        end
      end
    end
  end
end



function lns = evalCycle(cyc,A,B)
  b = de2bi(cyc);
  R = eye(size(A));
```

```
49    for i = b
50      if i
51        R = R*A;
52      else
53        R = R*B;
54      end
55    end

57    lns = lnsp(R);
58  end

60  A = [[0,1,0];
61  [0,0,1];
62  [1/3,1/3,1]]

64  B = [[0,1,0];
65  [0,0,1];
66  [-1/3,-1/3,1]]


69  n = 16;

71  tic
72    exp(sum(arrayfun(@(c)evalCycle(c,A,B), getAllPrimeCycles(n)))/2^
        n)
73  toc
```

Satz 3.3 sagt uns insbesondere, dass wir obere Schranken gefunden haben, leider aber auch nur Näherungsweise, da wir nur von $|\cdot|$ ausgegangen sind, aber keiner echten Martrixnorm.
Nielsen behauptet zusätzlich, dass $\mathbb{E}(\gamma_n)$ exponentiell abnimmt. Um dies experimentell zu verifizieren, betrachten wir

$$\Delta(n) := \mathbb{E}(\gamma_n) - \mathbb{E}(\gamma_{25}) \quad n = 1, \ldots, 24$$

und Plotten $\log(\Delta(n))$ gegen n. Dieser doch annähernd gerade Verlauf in Abbildung 4 untermauert diese These. Demnach vermuten wir, dass $\sigma = 1.009\ldots$. Leider lassen sich keine exakten Fehlerschätzer finden.

3.9 Random-Walk-Ansatz: Monte Carlo Simulationen

Wie wir bereits in der Einleitung zu diesem Abschnitt besprochen haben, ist der einfachste Ansatz eine Monte Carlo Simulation. Man rechnet also die Folge bis x_n für hinreichend große n aus und mittelt über viele derartige Simulationen. Die Anzahl der Simulationen sei gegeben durch *it*.
Die bisherigen Konvergenzaussagen (z.B. Satz 3.3) garantieren uns den Erfolg der Methode, jedoch ist die Konvergenz üblicherweise eher langsam.
Wir gehen wie in Abschnitt 4.3 vor, um eine Abschätzung für ein benötigtes *it* zu finden. Die Zufallsvariable ist hier γ_n. Unsere ersten Experimente haben gezeigt, dass $\sigma^2_{\gamma_n}$ vermutlich relativ klein ist. Leider lässt sich analytisch aber nur eine eher schlechte Abschätzung finden.

$$\sigma^2_{\gamma_n} \leq \mathbb{E}(\gamma_n^2) = \mathbb{E}(\frac{\log(\prod_{i=1}^n \|T_i\|)^2}{n}) \leq \log(\chi)^2 \approx 0.150236842931682$$

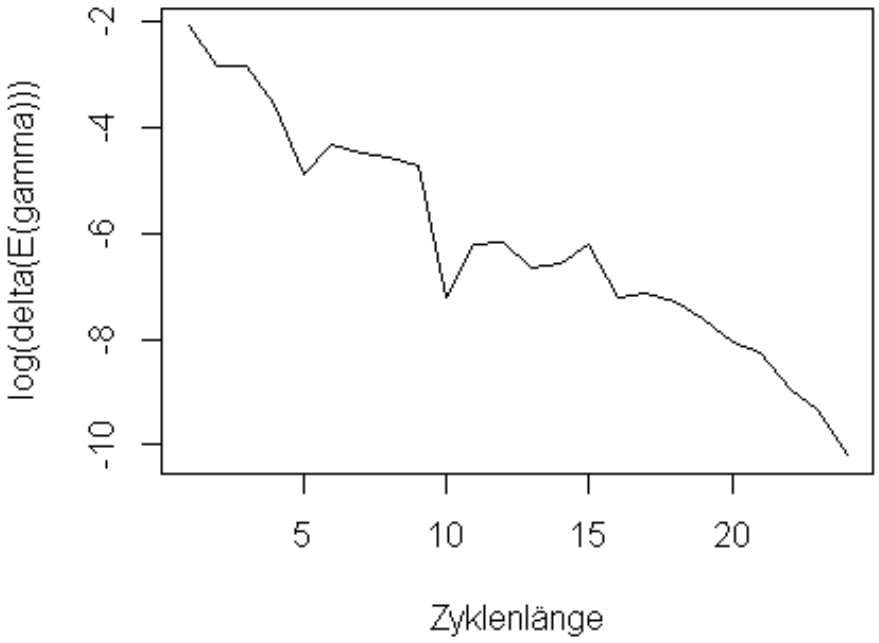


Abbildung 4: Graphische Verifikation der exponentiellen Konvergenz

Wobei wieder $\chi = \|A\| = \|B\| \approx 1.473446150061121$.
Somit erhalten wir die Schätzung $it \geq 0.150236842931682\frac{1}{\alpha\varepsilon^2}$, um mit Wahrscheinlichkeit α höchstens ε vom tatsächlichen Wert abzuweichen. Fpr $\alpha = 10^{-2}$ und $\varepsilon = 10^{-3}$ ergibt das beispielsweise $it > 15 \cdot 10^6$ notwendige Iterationen.
Wir haben aber immer noch ein Problem, wenn n zu groß wird, gibt es einen Overflow in den Daten. Dazu haben wir zwei Lösungen gefunden.
Die erste befasst sich direkt mit der Originalangabe, wie wir wiefolgt umgeformt haben:

$$\frac{x_{n+3}}{x_{n+2}} = 1 \pm \frac{1}{3}\left(\frac{x_{n+1}}{x_{n+2}} + \frac{x_n}{x_{n+2}}\right) \quad n \in \mathbb{N}^+ \tag{3.8}$$

Betrachtet man nun die Folge der Steigungen $u_n := \frac{x_{n+1}}{x_n}$ und $u_0 := 1$, kann man das Problem so anschreiben.

$$u_{n+2} = 1 \pm \frac{1}{3}\left(\frac{1}{u_{n+1}} + \frac{1}{u_n \cdot u_{n+1}}\right)$$

Die Werte u_i bleiben stets klein und x_{i+1} kann über das Produkt der $\prod_{j=0}^{i} u_j$ rückgerechnet werden. Durch Logarithmieren erhalten wir schließlich

$$\sigma_n = e^{1/n \cdot \sum_{j=0}^{n-1} \log(u_j)}$$

Die zweite Lösung, welche wir letztendlich für unsere Simulationen herangezogen haben, geht auf das Problem in Matrizenschreibweise $G_n \cdot t_0$. Sie befasst sich also mit der Rekursion $t_{n+1} = T_n \cdot t_n$, wobei T_n entweder A oder B ist.

Hier ist die wesentliche Idee die Vektoren t_n zu Normieren und erhält eine Rekursion der Art:

$$\begin{aligned} \nu_n &= \frac{t_n}{\|t_n\|} \\ norm_{n+1} &= norm_n \cdot \|t_n\| \\ t_{n+1} &= T_n \cdot \nu_n \end{aligned}$$

Auch hier ist es vorteilhaft nicht das Produkt der Normen zu speichern, sondern die Summe der Logarithmen. Kompakt geschrieben in Matlab ergibt das die folgende Funktion:

Listing 7: Berechnen eines Folgengliedes

```
function val = simSeq(n)

        A = [[0,1,0];
        [0,0,1];
        [1/3,1/3,1]];

        B = [[0,1,0];
        [0,0,1];
        [-1/3,-1/3,1]];

        x = [1;1;1];
        loglen = 0;

        for i=1:n
                sign = randi(2);
                if(sign == 1)
                        x = A*x;
                else
                        x = B*x;
                end

                        len = norm(x);
                        x = x/len;

                        loglen = loglen + log(len);
        end

        val = exp((loglen+log(abs(x(1))))/n);
end
```

Die Simulationen liefern uns folgende Resultate aus Tabelle 2
Es ist also ledigilich eine Frage der Geduld, wie genau man das Ergebnis möchte. Wir haben vergleichsweise wenige Iterationen gemacht. Uns war es nämlich wichtig n groß zu machen.
Wir können nicht wissen, doch wir wollen glauben, dass wir hiermit $\sigma = 1.0090\ldots$ ablesen können. Zufrieden können wir nun allemal sein, weil sich dieses Resultat mit der Zyklenmethode deckt.
Auch wenn wir hier nur Nullereignisse simulieren können, so sagen uns die Konvergenzaussagen doch, dass wir für hinerichend große n und viele Simulationen it, mit großer Wahrscheinlichkeit eine gute Näherung erhalten.

n	it	$\mathbb{E}(\sigma_n)$	CPU time [s]
10^6	10^3	1.009108907048898	7618.18
10^7	500	1.009087635400575	35957.63
10^8	10	1.009090295134524	7041.36

Tabelle 2: Ergebnisse mittels Monte Carlo Simulation

Numerisch gesehen gibt es auch noch keine Probleme, da sich die kummulierten Rundungseffekte bei 10^8 iterativen Berechnungen erst in der siebenten oder achten Nachkommastelle zeigen sollten.

4 Aufgabe 4

4.1 Problembeschreibung

In dieser Aufgabe war folgende reelle Zahl I zu berechnen:

$$I = \int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin\left(\frac{1}{x}\right)}\right)}\right) dx = \int_0^1 f(x)dx \tag{4.1}$$

Der Integrand $f(x)$ ist an abzählbar unendlich vielen Stellen nicht definiert, da abzählbare Mengen jedoch Lebesgue-Nullmengen sind, stellt dies kein Problem für die Wohldefiniertheit dar[6]. Aus der Jensen-Ungleichung für konvexe Funktionen (wir benötigen diese, um Beträge ins Integral zu ziehen, siehe [4, S. 221f]) und der Hölder-Ungleichung (siehe [11, S. 217]) erhalten wir die Abschätzung

$$|I| \leq \|f\|_{L^1([0,1])} \leq \|f\|_{L^\infty([0,1])}\text{meas}([0,1]) = 1 < \infty \tag{4.2}$$

Daraus ist unmittelbar ersichtlich, dass $I \in [-1, 1]$ liegen muss.
Mittels des Befehls `NIntegrate` in Mathematica, der für die numerische Integration zuständig ist, erhalten wir die Fehlermeldung

```
NIntegrate::slwcon: "Numerical integration converging too slowly; suspect one
of the following: singularity, value of the integration is 0, highly
oscillatory integrand, or WorkingPrecision too small."

NIntegrate::ncvb: "NIntegrate failed to converge to prescribed accuracy after
recursive bisections in x near 0.32225008335946764. NIntegrate obtained
0.49698898495984045 and 0.014892075445874203 for the integral and error estimates.
```

Dies ist wenig verwunderlich, zumal die Dichtefunktion (siehe Abbildung) abzählbar unendlich viele Oszillationen aufweist. Allerdings zeigt der Algorithmus in Mathematica schon die ungefähre Größenordnung des Werts des Integrals an. Insbesondere mittels Cross-Referencing mit den Ergebnissen der Monte-Carlo-Integration (siehe Kapitel 4.3) gehen wir davon aus, dass $I \approx 0.497$.

[6] da $f(x)$ λ-f.ü. stetig ist, ist f sowohl Riemann- als auch Lebesgue-integrierbar und die Integrale stimmen überein[11, S. 146f]

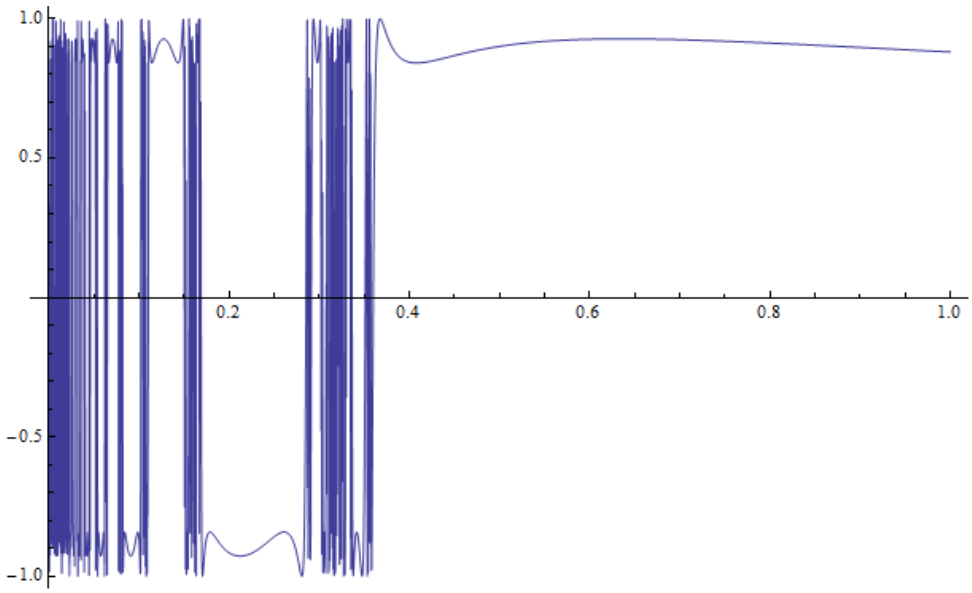


Abbildung 5: Dichtefunktion von $f(x)$ aus (4.1)

4.1.1 Umformulierung des Integrals mittels Substitution

$$
\begin{aligned}
g(x) &:= \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right) \\
g(x) &= g(x+2\pi i) \quad \forall i \in \mathbb{N} \\
g(x-\pi) &= -g(x) \\
I &= \int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin\left(\frac{1}{x}\right)}\right)}\right) dx = \int_1^\infty \frac{1}{x^2} \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right) dx \\
&= \int_1^\pi \frac{1}{x^2} g(x)dx + \sum_{i=1}^{\infty} \int_{2\pi i-\pi}^{2\pi(i+1)-\pi} \frac{1}{x^2} g(x)dx \\
&= \int_1^\pi \frac{1}{x^2} g(x)dx + \sum_{i=1}^{\infty} \int_{-\pi}^{\pi} \frac{1}{(z+2\pi i)^2} g(z+2\pi i)dz \\
&= \int_1^\pi \frac{1}{x^2} g(x)dx + \int_{-\pi}^{\pi} g(z) \sum_{i=1}^{\infty} \frac{1}{(z+2\pi i)^2} dz \\
&= \int_1^\pi \frac{1}{x^2} g(x)dx + \int_0^{\pi} g(z) \left(\sum_{i=1}^{\infty} \frac{1}{(z+2\pi i)^2} - \sum_{i=1}^{\infty} \frac{1}{(z-\pi+2\pi i)^2} \right) dz \\
&= \underbrace{\int_1^\pi \frac{1}{x^2} g(x)dx}_{=:I_1} + \underbrace{\int_0^{\pi} g(z) \left(\frac{\psi_1\left(1+\frac{z}{2\pi}\right)}{4\pi^2} - \frac{\psi_1\left(\frac{z+\pi}{2\pi}\right)}{4\pi^2} \right) dz}_{=:I_2}
\end{aligned}
$$

Mittels dieser Umformulierung ergibt sich eine Dichtefunktion im Integranden, welche in Abbildung 6 dargestellt ist. Es gilt zu beachten, dass die Umformulierung der unendlichen Reihen

mittels des Digamma-Funktion geschieht, welche mittels der logarithmischen Ableitung der Gamma-Funktion

$$\psi_1(z) = \frac{d}{dz}\ln(\Gamma(z)) = \frac{\Gamma'(z)}{\Gamma(z)} \tag{4.3}$$

definiert ist. Bis auf negative ganze Elemente ist diese in ganz $\mathbb{C}$ holomorph.

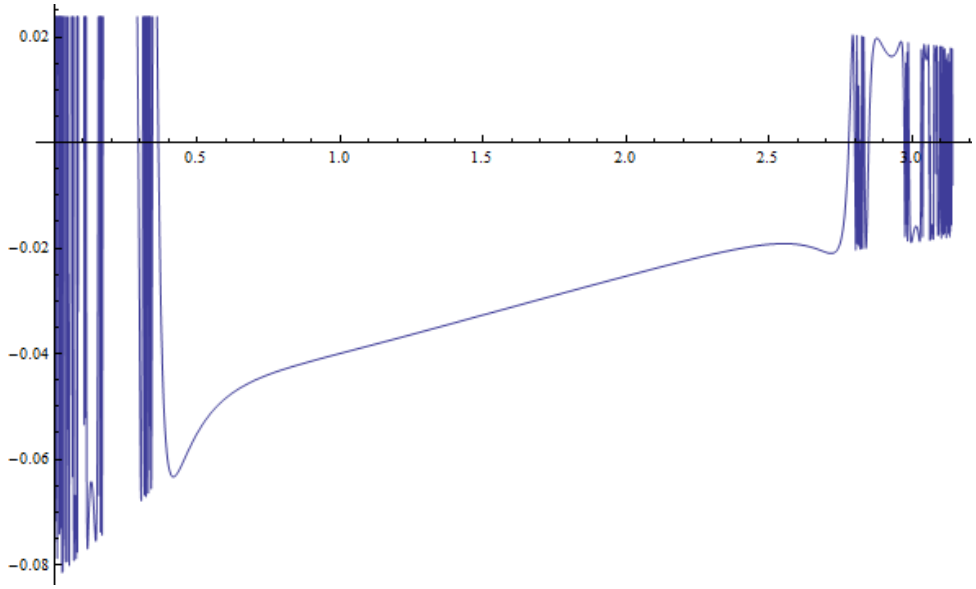


Abbildung 6: Darstellung der Dichtefunktion $g(z)\left(\frac{\psi_1\left(1+\frac{z}{2\pi}\right)}{4\pi^2} - \frac{\psi_1\left(\frac{z+\pi}{2\pi}\right)}{4\pi^2}\right)$ auf $[0,\pi]$, die Oszillationen sind nun auf den linken und rechten Rand des Integrationsgebietes konzentriert.

Mittels Mathematicas numerischer Integrations-Algorithmen ergibt sich mittels

```
NIntegrate[PolyGamma[1, 1 + z/(2 Pi)]/(4 Pi^2)*Sin[1/Sin[1/Sin[z]]], {z, -Pi, Pi},
MaxRecursion -> 300, Method -> {GlobalAdaptive, MaxErrorIncreases -> 10000}]
```

für I_2

$$I_2 \approx -0.0714071$$

und mittels

```
NIntegrate[x^(-2)*Sin[1/Sin[1/Sin[x]]], {x, 1, Pi},
MaxRecursion -> 300, Method -> {GlobalAdaptive,MaxErrorIncreases -> 10000}]
```

für I_1

$$I_1 \approx 0.56762$$

Somit erhalten wir (auch wenn Mathematica, trotz seiner Kollokationsalgorithmen für hochoszillierende Funktionen Warnungen ausgibt, und die Ergebnisse leider mit Vorbehalt zu genießen sind) nach einiger Rechenzeit

$$I \approx 0.4962129 \tag{4.4}$$

4.2 Numerische Quadratur mit hp-Gauß-Integration

Ein weiterer Standard-Ansatz, welcher sich aufdrängt, um I zu berechnen ist ein Gauß-Quadratur-Algorithmus. Wir haben vor einigen Semestern bei Prof. Praetorius in Scientific Computing einen Gauß-Legendre-Quadratur-Algorithmus in einer besonderen Form implementiert. Unser Algorithmus hieß `hpgauss`, da in unserem Code das Integrationsintervall $[a, b]$ durch eine arbiträre Partitionierung $a = a_1 < a_2 < \cdots < a_n = b$ zerlegt wird. Weiters wird dem Algorithmus auf jedem Teilintervall ein Wert p_i übergeben, welcher anzeigt, dass auf jedem Teilintervall Gauß-Legendre-Quadratur $m := p_i$-ter Stufe durchgeführt werden soll. Die Integrationsgewichte ω_k und Stützstellen $x_k \in (a_i, a_{i+1})$ (mit $1 \leq k \leq m$ und $1 \leq i \leq n-1$) für die Quadraturformel

$$G_m[f] = \sum_{k=1}^{m} \omega_k f(x_k) \tag{4.5}$$

werden mittels des Algorithmus von Golub-Welsh[7, S. 222f] über ein Matrix-Eigenwertproblem numerisch für beliebige m berechnet. Weiters gilt für Gauß-Quadratur folgende Abschätzung:

Satz 4.1. *Sei $f \in C^{2m}[a, b]$ und $G_m[f; \omega]$ die m-stufige Gauß-Formel für $I[f; \omega] := \int_a^b f(x)\omega(x)dx$. Dann gilt*

$$|I[f; \omega] - G_m[f; \omega]| \leq \frac{\|f^{(2m)}\|_{\infty,[a,b]}}{(2m)!\gamma_m^2} \tag{4.6}$$

wobei γ_m den Höchstkoeffizienten des Orthogonalpolynoms u_m bezeichnet, der im Fall der Legendre-Polynome durch

$$\gamma_m^2 = \frac{2m+1}{2} \frac{(2m!)^2}{4^m (m!)^4} \tag{4.7}$$

gegeben ist (siehe [8, S. 290f]).

Beweis. Siehe [8, S. 340f]. □

Die Funktion $f(x)$ ist in unserem Beispiel bis auf abzählbar viele Stellen eine glatte Funktion. Wir können die Funktion jedoch nur in endlich viele Intervalle unterteilen, wobei dann in mindestens einem offenen Intervall die Funktion dann nicht global C^∞ ist. Da die Funktion jedoch durch 1 beschränkt ist und $\|f^{(2m)}\|_{L^\infty([a,b])} := \operatorname{esssup}(f^{(2m)}) = 1$. Leider können wir nicht in abzählbar unendlich viele Intervalle unterteilen, weil wenn wir an allen Singularitäten die Intervalle teilen würden wären wir im Inneren glatt genug. Wir sind somit leider analytisch im Blindflug unterwegs. Ein hp-Gauß mit äquidistanter Unterteilung reagiert sehr sensibel auf die Wahl der Stützstellen. Dennoch wollen wir dem geneigten Leser die Ergebnisse nicht vorenthalten, sie sind in Tabelle 3 aufgeführt. Es ergibt sich als Wert für das Integral $I \approx 0.496$, was auch mittels Cross-Referencing mit unserem Mathematica-Ergebnis plausibel erscheint.

4.3 Monte-Carlo-Methoden

Ein beliebter Standard-Ansatz zur numerischen Quadratur ist die so genannte Monte-Carlo-Integration.

N	p	I	CPU time [s]
100	5	0.503243096512242	0.009810473722118
1000	5	0.497309124651590	0.072127810247930
10000	5	0.496572484759540	0.723478667200356
100000	5	0.496078327401465	6.646258466438822
1000000	5	0.496221409580183	63.509018693027564
10000000	5	0.496219282939691	691.585694940753800
100	10	0.486018165316241	0.011398261161451
1000	10	0.497758038643706	0.096733082064462
10000	10	0.496225507874219	0.958849741577029
100000	10	0.496312867761303	8.049283593929701
1000000	10	0.496204063670727	77.971788096182493
10000000	10	0.496213253763635	783.820996587162540
100	50	0.495847926456484	0.091219293971902
1000	50	0.496045332559514	0.605001349317463
10000	50	0.496185461852564	5.409385333070111
100000	50	0.496259436457901	51.068420356763163
1000000	50	0.496217761951309	514.128531091353350

Tabelle 3: Ergebnisse mittels hp-Gauß mit äquidistanter-Intervallzerlegung in N Intervalle, p bezeichnet den (auf jedem Intervall gleich gewählten) Polynomgrad der Gauß-Legendre-Quadratur auf den Teilintervallen

Allgemein ist die Idee auf dem Integrationsbereich Ω eine Zufallsvariable Y mit Dichte p zu definieren und den Erwartungswert von $X := \frac{f}{p}(Y)$ zu bestimmen. Dann gilt nämlich:

$$\mathbb{E}\left(\frac{f(Y)}{p(Y)}\right) = \int_\Omega \frac{f(y)}{p(y)} p(y) d\lambda(y) = \int_\Omega f d\lambda$$

Der Erwartungswert soll nun über endliche Stichproben bestimmt werden. Die Grundlage dafür bietet das schwache Gesetz der großen Zahlen.
Hierbei muss das Verfahren nicht gegen den exakten Wert konvergieren, jedoch mit einer arbiträren Wahrscheinlichkeit α unterscheidet sich der berechnete Wert nur um ε vom „exakten“ Wert. Ein bisschen Theorie wird deutlich machen, was damit gemeint ist.

Satz 4.2 (Schwaches Gesetz der großen Zahlen für iid Zufallsvariablen). *Ist (X_n) eine Folge unabhängig, identisch verteilter Zufallsvariablen aus $\mathfrak{L}_2(\Omega, \mathfrak{S}, P)$ mit dem gemeinsamen Erwartungswert $\mathbb{E}X$, so gilt*

$$P\left(\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}X\right| > \varepsilon\right]\right) \leq \frac{\sigma_X^2}{n\varepsilon^2} \tag{4.8a}$$

$$\lim_{n\to\infty} P\left(\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}X\right| > \varepsilon\right]\right) = 0 \tag{4.8b}$$

Beweis. Siehe [11, S. 255]. □

Insbesondere liefert (4.8a), dass wenn man mit einer Wahrscheinlichkeit von α um maximal ε vom Wert des Integrals abzuweichen (also $P([|\overline{X}_n - \mathbb{E}X| > \varepsilon]) \leq \alpha$), eine Bedingung an die Anzahl der Zufallszahlen erhält, sofern man die Varianz σ_X^2 kennt bzw. schätzen kann:

$$n \geq \frac{\sigma_X^2}{\alpha \varepsilon^2} \tag{4.9}$$

Wir müssen somit $\sigma_X^2 = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$ ermitteln, was die Aufgabe nicht leichter macht.
Wir legen nun fest, dass wir Y gleichverteilt auf $\Omega = [0, 1]$ wählen wollen. Dann ist die Dichte konstant 1.
Es gilt daher $\sigma_X^2 \leq \mathbb{E}(X^2) = \int_0^1 f(x)^2 dx \leq \|f(x)^2\|_{L^\infty([0,1])} \cdot 1 = 1$. Dies kann eine sehr pessimistische Schätzung sein, jedoch ermöglicht uns diese Schätzung n für unsere Genauigkeitsanforderungen zu berechnen. Für $\alpha = 10^{-2}$ und $\varepsilon = 10^{-4}$ ergibt sich beispielsweise $n \geq 10^9$. Es ist ersichtlich, dass die Anzahl der benötigten Zufallszahlen für jede weitere Stelle (also für eine Verkleinerung von ε) quadratisch wächst.

4.3.1 Implementierung der Monte-Carlo-Quadratur in R

Dieses Verfahren ist extremst simpel und schnell zu implementieren, jedoch ist wie in Aufgabe 3 Geduld gefragt.

Listing 8: Monte-Carlo-Integration in R

```
1  options(digits=16)
2  ptm <- proc.time() % zur Zeitmessung
3  x <- runif(1E9,0,1)
4
5  f <- function(x) {
6    sin(1/sin(1/sin(1/x)))
7  }
8
9  print(mean(sapply(x,f)))
10 print(proc.time()-ptm)
```

Dieser Code ist mit größter Vorsicht auszuführen, da er sehr arbeitsspeicherintensiv ist. Anstatt in einer Schleife immer nur eine neue Zufallszahl zu erzeugen und die alte zu überschreiben, wird sofort ein Vektor mit 10^9 Zufallszahlen angelegt. Das hat den Vorteil, dass sowohl die Erzeugung der Zufallszahlen, als auch das Anwenden der Funktion f mit `sapply` parallel durchgeführt werden könnte.
Während wir warten, nützen wir die Zeit wieder für numerische Überlegungen. Können wir $f(x)$ bis auf die allgemeine Rechenungeanuigkeit *eps* exakt bestimmen, so ergibt die Addition von 10^9 mal diesem Fehler, dass wir höchstens 7 unserer ursprünglich 16 Stellen der double precision richtig erhalten werden. Diese Annahme ist aber sehr optimistisch, kleine Fehler im Bereich der Singularitäten von $\sin(1/\sin(1/x))$ können sich bei der nächsten Iteration von $\sin(1/\cdot)$ drastisch auswirken. Das Ergebnis ist also eher kritisch zu sehen und bietet sich als Cross-Referencing zu anderen Verfahren an.
Zu guter letzt erhalten wir auch das Ergebnis $I = 0.4965673019940234$.

5 Aufgabe 5

Aufgabe 5 konfrontierte uns mit folgender Problemstellung aus dem Bereich der nichtlinearen Partiellen Differentialgleichungen:
Sei u die klassische Lösung von

$$u_t - \Delta u = e^u \quad \text{auf } \Omega, t > 0 \tag{5.1a}$$
$$u = 0 \quad \text{auf } \partial\Omega, t > 0 \tag{5.1b}$$
$$u(0, \cdot) = 1 \quad \text{auf } \Omega \tag{5.1c}$$

Wir sollten die reelle Zahl t^* bestimmen, welche als

$$t^* = \min\{t > 0 : \|u(t)\|_{L^\infty(\Omega)} = +\infty\} := \min\{t > 0 : \sup_{x\in\Omega} |u(x,t)| = +\infty\} \tag{5.2}$$

Es stellte sich jedoch im Laufe unserer Seminarsitzungen die Frage, ob das Problem überhaupt wohldefiniert ist. Wir waren daher angehalten, statt (5.1a) auf $\Omega = (0,1)\times(0,1)$ das modifizierte Problem

$$u_t - \Delta u = e^u \quad \text{auf } \Omega, t > 0 \tag{5.3a}$$
$$u = 0 \quad \text{auf } \partial\Omega, t > 0 \tag{5.3b}$$
$$u(0, \cdot) = 20 > 2\pi^2 \quad \text{auf } \Omega \tag{5.3c}$$

zu betrachten, da wir, wie in Kapitel 5.1 nachzulesen ist, für dieses Problem den Blow-Up nachweisen können und somit sicher ein $t^* \in \mathbb{R}$ existiert.

5.1 Beweis der Divergenz der Lösung

Satz 5.1. *Sei u eine klassische also auch schwache Lösung der Differentialgleichung (5.1a) und sei $w_1 > 0$ die normierte Eigenfunktion von $-\Delta$ auf $H_0^1(\Omega)$ zum kleinsten Eigenwert $\lambda_1 > 0$. Wenn $\int_\Omega u(0,x)w_1 dx > \lambda_1$, dann existiert ein $t^* > 0$, so dass*

$$\lim_{t\to t^*} \int_\Omega u(t,x)w_1 dx = \infty$$

Beweis. Läuft analog zu [10, S.50], zu beachten ist nur, da $u(x) > 0$ damit $u(x)^2 \le e^{u(x)}$ und

$$z \le \left(\int_\Omega u^2 w_1 dx\right)^{\frac{1}{2}} \le \left(\int_\Omega e^u w_1 dx\right)^{\frac{1}{2}}$$

gilt. □

Im Beweis ergibt sich die Ungleichung

$$y(t) \ge \frac{y(0)\lambda_1}{\lambda_1 - y(0)(1 - e^{-\lambda_1 t})}$$
$$y(t) = e^{\lambda_1 t} z(t)$$
$$z(t) = \int_\Omega u(t) w_1 dx$$

also

$$\lambda_1 - y(0)(1 - e^{-\lambda_1 \tilde{t}}) = 0 \Rightarrow \lim_{t \to \tilde{t}} \int_\Omega u(t,x) w_1 dx = \infty$$

Eigenfunktionen zu $-\Delta$ sind gegeben durch $w_{k,n} = sin(k\pi x_1) sin(n\pi x_2)$ und damit ist die normierte Eigenfunktion zum kleinsten Eigenwert $\lambda_1 = 2\pi^2$ gegeben durch $w_{1,1} = \frac{\pi^2}{4} sin(\pi x_1) sin(\pi x_2)$. Also gibt $\tilde{t}$, so er existiert, eine obere Schranke für t^*.

$$\begin{aligned} &\lambda_1 - y(0)(1 - e^{-\lambda_1 \tilde{t}}) = 0 \\ \Leftrightarrow &e^{-\lambda_1 \tilde{t}} = 1 - \frac{\lambda_1}{y(0)} \\ \Leftrightarrow &\tilde{t} = -\frac{\ln\left(1 - \frac{\lambda_1}{y(0)}\right)}{\lambda_1} \end{aligned}$$

In unserem Beispiel bedeutet das (da $u(0)$ konstant):

$$\begin{aligned} y(0) = z(0) = \int_\Omega u(0,x) w_1 dx = u(0) \\ t^* \leq \tilde{t} = -\frac{\ln\left(1 - \frac{2\pi^2}{u(0)}\right)}{2\pi^2} \end{aligned}$$

Man bekommt eine hinreichende Bedingung für die Existenz von t^* und eine Abschätzung, wenn der Startwert größer als $2\pi^2$ gewählt wird.

5.2 Implementierung mittels FEM in COMSOL

Unser erster Ansatz war, das kommerzielle FEM-Programm COMSOL zu verwenden, welches einige befreundete Physik-Studenten hoch angepriesen hatten. Es erwies sich jedoch aus mathematischer Sicht als sehr problematisch, dass aus der Menüstruktur nicht klar ersichtlich wurde, welche Art von Basisfunktionen die FEM, die wir für die Ortsdiskretisierung brauchen, intern vom Programm verwendet werden. Für die Zeitschritte wurden BDF-Verfahren der Konvergenzordnung 1 bis 5 verwendet. Als gewünschte Zeitschrittweite wurde 10^{-9} gewählt, wobei der Zeitschrittalgorithmus adaptiv ist und bei Bedarf kleinere Zeitschritte wählt.
Nach einiger Zeit gibt COMSOL folgende Fehlermeldung aus:

```
Repeated error test failures. May have reached a singularity.
Time : 4.954337253702532e-011
Last time step is not converged.
```

Allerdings ist anzunehmen, dass diese untere Schranke für t^* sehr pessimistisch ist, da COMSOL vermutlich nicht dafür ausgelegt ist, Differentialgleichungen mit Blow-Up auszurechnen.

5.3 Implementierung mittels Finiter Differenzen und Explizitem Euler-Timestepping

In erster Annäherung an das Problem diskretisierten wir das Gebiet mit einem äquidistanten Gitter der Schrittweite h und verwendeten Approximationsformeln für den Differenzenquotienten der zweiten Ableitung. Es bezeichne ferner $u^n_{i,j} := u(i \cdot h, j \cdot h, n \cdot \Delta t)$ Es gilt [8, vgl. S. 632]:

N	Δt	t^*	$\|u(t)\|_{L^\infty(\Omega)}$	CPU time [s]
31	9.765625e-010	0.000000005859375	Inf	0.002233
51	3.698225e-010	0.000000003698225	Inf	0.002565
101	9.611688e-011	0.000000002595156	Inf	0.019146
151	4.328255e-011	0.000000002293975	Inf	0.066654
201	2.450740e-011	0.000000002205666	Inf	0.125261
301	1.096443e-011	0.000000002127100	Inf	0.723326
401	6.187966e-012	0.000000002103908	Inf	3.477489
501	3.968191e-012	0.000000002087268	Inf	7.904998
1001	9.960120e-013	0.000000002068717	Inf	123.208158
3001	1.109631e-013	0.000000002062138	Inf	12197.729321

Tabelle 4: Ergebnisse mittels explizitem Euler und 5-Punkt-Stern-Ortsdiskretisierung

$$\frac{\partial^2 u_{i,j}^n}{\partial x^2} \approx D_{h,x}^2[u_{i,j}^n] := \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \mathcal{O}(h^2) \tag{5.4}$$

$$\frac{\partial^2 u_{i,j}^n}{\partial y^2} \approx D_{h,y}^2[u_{i,j}^n] := \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h^2} + \mathcal{O}(h^2) \tag{5.5}$$

$$\Delta u_{i,j}^n \approx L_h^5[u_{i,j}^n] := \frac{u_{i,j-1} + u_{i,j+1} + u_{i+1,j} + u_{i-1,j} - 4u_{i,j}^n}{h^2} + \mathcal{O}(h^2) \tag{5.6}$$

Für das explizite Euler-Verfahren gilt für den Zeitschritt (für autonome Differentialgleichungen wie hier):

$$\begin{aligned} \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} &= f(u_{i,j}^n) \\ u_{i,j}^{n+1} &= u_{i,j}^n + \Delta t f(u_{i,j}^n) \end{aligned} \tag{5.7}$$

In unserem Beispiel war

$$f(u_{i,j}^n) = L_h^5(u_{i,j}^n, u_{i-1,j}^n, u_{i+1,j}^n, u_{i,j-1}^n, u_{i,j+1}^n) + e^{\pi^2 u_{i,j}^n} \tag{5.8}$$

Durch unsere numerischen Experimente kamen wir zu dem Schluss, dass das Maximum von u im Mittelpunkt von Ω liegen muss. Insofern ließen wir Zeitschritte solange laufen, bis der Wert des mittlersten Freiheitsgrades eine bestimmte Schranke η überschritten hatte, womit wir eine untere Schranke für t^* erhalten. Ein wesentlicher Faktor ist natürlich auch die Ortsdiskretisierung, die möglichst fein aufgelöst werden soll. Die Ergebnisse sind in Tabelle 4 zusammengetragen. Mittels Aitken-Extrapolation ergibt sich ein Wert von

$$0.000000002062138$$

Ich halte $t^* \approx 2.0 \cdot 10^{-9}$ für signifikant, aber kann diesen Wert aus Zeitgründen nicht mehr durch Verfahren höherer Ordnung bzw. ein A-stabiles Timestepping-Verfahren überprüfen, da hiezu mehr Zeit und weniger andere Prüfungen notwendig gewesen wären. Auf jeden Fall erhalten wir eine größere Schranke $\widetilde{t} \leq t^*$

Ebenso kann man ein Finite-Differenzenverfahren mit höhere Konsistenzordnung herleiten, nämlich den so genannten 9-Punkte-Stern. Dieser hat eine höhere Konsistenzordnung (Abschneidefehler $\mathcal{O}(h^4)$ für hinreichend glatte Funktionen).

5.4 Ausblick & To do

Leider reichte in der Prüfungsphase die Zeit nicht, um unsere Resultate zu vertiefen. Wir versuchten den 5-Punkte-Stern mit implizitem Euler-Verfahren beziehungsweise A-stabilen Runge-Kutta-Verfahren höherer Ordnung (z.B. Gauß-Radau) hochzuziehen. Auch der 9-Punkte-Stern (Konvergenzordnung $\mathcal{O}(h^4)$) wurde wegen Diskussionen über die Fortsetzung der Randdaten (der 9-Punkte-Stern benötigt an den Rändern 2 Auswertungspunkte) nicht implementiert.
Weiters könnte auch die Ortsdiskretisierung mit adaptiver P1-FEM implementiert werden bzw. FEM mit Polynomgraden höherer Ordnung, da wir ja für $t > 0$ eine klassische Lösung voraussetzen und somit H^1-High-Order-FE-Räume verwenden könnten.

Literatur

[1] Auzinger, W. ; Praetorius, D.: *Numerische Mathematik*. Wien, 2011. – Skriptum

[2] Bornemann, Folkmar: *Vom Lösen numerischer Probleme; ein Streifzug entlang der SSIAM 10x10-Digit Challenge"; mit 44 Tabellen; The SIAM 100-Digit Challenge*. Berlin [u.a.] : Springer, 2007

[3] Bougerol, J. L. Ph.: *Products of Random Matrices with Applications to Schrödinger Operators.* Birkhäuser, 1985

[4] Elstrodt, Jürgen: *Maß- und Integrationstheorie*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011 (Springer-Lehrbuch)

[5] Embree, Mark ; Trefethen, Lloyd N.: Growth and decay of random Fibonacci sequences. 455 (1999), Juli, Nr. 1987, S. 2471–2485

[6] Furstenberg, H. ; Kesten, H.: Products of Random Matrices. In: *The Annals of Mathematical Statistics* 31 (1960), 06, Nr. 2, S. 457–469

[7] Golub, Gene H. ; Welsch, John H.: Calculation of Gauss Quadrature Rules. In: *Mathematics of Computation* 23, Nr. 106, S. pp. 221–230+s1–s10

[8] Hanke-Bourgeois, Martin: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Wiesbaden : Vieweg+Teubner: Wiesbaden, 2009

[9] Hardy, John Edensor ; Pólya G. Godfrey H. ; Littlewood L. Godfrey H. ; Littlewood: *Inequalities*. 2. ed., reprint. Cambridge [u.a.] : Cambridge Univ. Press, 1988

[10] Jüngel, Ansgar: *Nichtlineare partielle Differentialgleichungen*. 2014. – Skriptum, online verfügbar unter `http://www.asc.tuwien.ac.at/~juengel/scripts/nPDE.pdf`

[11] Kusolitsch, Norbert: *Maß- und Wahrscheinlichkeitstheorie*. 1. Auflage. Wien : Springer-Verlag, 2011

[12] Mainieri, Ronnie: Cycle expansion for the Lyapunov exponent of a product of random matrices. In: *Chaos* 2 (1992), Nr. 1, S. 91–97

[13] Mainieri, Ronnie: Zeta function for the Lyapunov exponent of a product of random matrices. In: *Phys Rev Letters* 68 (1992), Nr. 13

[14] Meyer, Carl D.: *Matrix analysis and applied linear algebra*. Philadelphia, Pa. : SIAM, 2000

[15] Nielsen, Jakob L.: *Lyapunov Exponent for Products of Random Matrices*. 1997

[16] Pollicott, Mark: *Lyapunov exponents for random matrix products using determinants*

[17] Riffer-Reinert, Ben: *The Zeta Function and its relation to the prime number theorem*. 2011. – Skriptum, online verfügbar unter `http://www.math.uchicago.edu/~may/VIGRE/VIGRE2011/REUPapers/Riffer-Reinert.pdf`

[18] Viswanath, Divakar: *Lyapunov Exponents From Random Fibonacci Sequences To The Lorenz Equations*, Diss., 1998

[19] Viswanath, Divakar: Random Fibonacci Sequences and the Number 1.13198824. In: *Math. Comput.* 69 (2000), Juli, Nr. 231, S. 1131–1155

Digit Challenge

Nico Amann & Nathanael Skrepek

Wien, 26. Juni 2014

Inhaltsverzeichnis

1 Primzahlenreihe

Die Menge $\mathbb{P}$ bezeichnet im folgenden die Menge aller Primzahlen in den natürlichen Zahlen $\mathbb{N}$.

$$\sum_{p\in\mathbb{P}} \frac{p(p-1)}{p^\pi}$$

Durch Ausmultiplizieren der Summanden und Aufspalten der Reihe erhält man

$$\sum_{p\in\mathbb{P}} \frac{p(p-1)}{p^\pi} = \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-2}} - \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-1}}.$$

Diesen zwei Reihen lässt sich unmittelbar ihre Konvergenz ablesen, da $\pi-2>1$. Außerdem sind diese Reihen Auswertungen der *Primzetafunktion.*

Definition 1.1. Die Primzetafunktion P ist definiert durch

$$P : \begin{matrix} \{z\in\mathbb{C} : \Re(z) > 1\} & \to & \mathbb{C} \\ z & \mapsto & \sum_{p\in\mathbb{P}} \frac{1}{p^z} \end{matrix}$$

Daher lässt sich das Auswerten der ursprünglichen Reihe darauf reduzieren $P(\pi-2)$ und $P(\pi-1)$ auszurechnen. Auf der Internetseite `http://www.wolframalpha.com` ist die Primzetafunktion bereits implementiert und berechnet 50 Stellen. Um jedoch, in der Hoffnung noch mehr Stellen zu erreichen, eine eigene Implementierung der Primzetafunktion schreiben zu können, hilft folgende Identität

Lemma 1.2. Sei μ die Möbiusfunktion der Halbordnung $(\mathbb{N}, |)$ (d.h. die natürlichen Zahlen mit der Teilbarkeitsrelation) und ζ die Riemann'sche Zetafunktion, dann gilt

$$P(z) = \sum_{n=1}^{\infty} \frac{\mu(n)}{n} \log\zeta(nz).$$

Beweis Mit Hilfe des Eulerprodukts für die Riemann'sche Zetafunktion erhält man

$$\zeta(z) = \prod_{p\in\mathbb{P}} \sum_{n=0}^{\infty} \frac{1}{p^{nz}} = \prod_{p\in\mathbb{P}} \frac{1}{1-\frac{1}{p^z}}$$

Auf beiden Seiten den Logarithmus zur Basis e angewandt und dann die Reihendarstellung von $-\log(1-x)$ ausgenützt ergibt

$$\log\zeta(z) = \sum_{p\in\mathbb{P}} -\log(1-\frac{1}{p^z}) = \sum_{p\in\mathbb{P}} \sum_{n=1}^{\infty} \frac{p^{-nz}}{n} = \sum_{n=1}^{\infty} \frac{1}{n} \sum_{p\in\mathbb{P}} \frac{1}{p^{nz}} = \sum_{n=1}^{\infty} \frac{P(nz)}{n}.$$

Das lässt sich umschreiben, wobei mit $z|k$ gemeint ist, dass k ein natürliches Vielfaches von z ist, in

$$\log\zeta(z) = \sum_{z|k} \frac{P(k)}{k} z \quad \text{bzw.} \quad \frac{\log\zeta(z)}{z} = \sum_{z|k} \frac{P(k)}{k}.$$

Diese Relation induziert eine Halbordnung ($a \leq b :\Leftrightarrow a|b$). Nachdem dann ein Intervall $[a,b]$ (bzgl. der HO) isomorph zu $[1, \frac{b}{a}]$ ist, stimmen die Möbiusfunktionen überein. Mittels der Möbiusinversionsformel lässt sich das auflösen zu

$$\frac{P(z)}{z} = \sum_{z|k} \mu(z,k) \frac{\log\zeta(k)}{k} = \sum_{z|k} \mu(1, \frac{k}{z}) \frac{\log\zeta(k)}{k} = \sum_{n=1}^{\infty} \mu(n) \frac{\log\zeta(nz)}{nz}$$

Multipliziert man beide Seiten mit z erhält man die Aussage.

❑

Zunächst sieht die rechte Seite wesentlich unhandlicher aus, da es durch die Riemann'sche Zetafunktion zu einer Doppelreihe wurde, jedoch ist diese Funktion in einigen Programmierumgebungen schon vorimplementiert. Unter anderem in PARI/GP kann man die Zetafunktion auf beliebige Genauigkeit auswerten lassen.

Lemma 1.3. Sei $x \geq 2$ und bezeichne ζ die Riemann'sche Zetafunktion. Dann gilt folgende Abschätzung:

$$\zeta(x) \leq 1 + \frac{1.5^{(1-x)}}{x-1} \tag{1}$$

Beweis Aus der Konvexität der Funktion $f : [1,\infty) \to \mathbb{R}$ mit $f(y) := y^{-x}$ folgt für $a = n+y$ und $b = n-y$

$$f(\frac{a+b}{2}) \leq \frac{f(a)+f(b)}{2} \quad \text{Eingesetzt:} \quad \frac{1}{n^x} \leq \frac{1}{2}\left((n+y)^{-x} + (n-y)^{-x}\right) \tag{2}$$

Somit lässt sich n^{-x} auch durch folgendes Integral nach oben abschätzen:

$$\begin{aligned} \frac{1}{n^x} &= \int_0^{0.5} 2n^{-x} dy \leq \int_0^{0.5} (n+y)^{-x} + (n-y)^{-x} dy \\ &= \int_{-0.5}^{0.5} (n+y)^{-x} dy = \int_{n-0.5}^{n+0.5} y^{-x} dy \end{aligned}$$

Somit gilt unter Berücksichtigung obigen Ungleichungen:

$$\zeta(x) = \sum_{n=1}^{\infty} n^{-x} \leq 1 + \sum_{n=2}^{\infty} \int_{n-0.5}^{n+0.5} y^{-x} dy = 1 + \int_{1.5}^{\infty} y^{-x} dy = 1 + \frac{1.5^{(1-x)}}{x-1}$$

❑

Bemerkung 1.4. Wie man durch Abwandlung der letzten Zeile sieht, gilt sogar allgemeiner, dass $\sum_{n=a}^{\infty} n^{-x} \leq \int_{a-0.5}^{\infty} (y)^{-x} dy = \frac{(a-0.5)^{(1-x)}}{x-1}$.

Korollar 1.5. *Sei $N \geq 2, y > 1$ und bezeichne ζ die Riemann'sche Zetafunktion, sowie μ die Möbiusfunktion. Dann gilt folgende punktweise Abschätzung für die näherungsweise Berechnung der Primzetafunktion P:*

$$\left| P(y) - \sum_{k=1}^{N} \frac{\ln(\zeta(ky))}{k} \mu(k) \right| \leq \frac{2.25}{(y-1) \cdot (N-1)^3} \cdot (1.5^y)^{-N} \tag{3}$$

Beweis Es gilt folgende Gleichheit: $P(y) = \sum_{k=1}^{\infty} \frac{\ln(\zeta(ky))}{k} \mu(k)$ [1]

$$\begin{aligned} \left| P(y) - \sum_{k=1}^{N} \frac{\ln(\zeta(ky))}{k} \mu(k) \right| &\leq \sum_{k=N+1}^{\infty} \left| \frac{\ln(\zeta(ky))}{k} \mu(k) \right| \leq \sum_{k=N+1}^{\infty} \frac{\ln(\zeta(ky))}{k} \\ &\leq \sum_{k=N+1}^{\infty} \frac{\ln(1 + \frac{1.5^{(1-ky)}}{ky-1})}{k} \leq \sum_{k=N+1}^{\infty} \frac{1.5^{(1-ky)}}{(ky-1)k} \end{aligned}$$

Das lässt sich wegen der Monotonie von $\frac{1.5^{(1-xy)}}{(xy-1)x}$ in x für $y > 1$ weiter abschätzen durch

$$\begin{aligned} \sum_{k=N+1}^{\infty} \int_{k-1}^{k} \frac{1.5^{(1-xy)}}{(xy-1)x} dx &= \frac{1.5}{y-1} \int_{N}^{\infty} \frac{1.5^{-xy}}{x^2} \frac{xy-x}{xy-1} dx \leq \frac{1.5}{y-1} \int_{N}^{\infty} \frac{1.5^{-xy}}{x^2} dx \\ &\leq \frac{1.5^2}{N^2(y-1)} \frac{(1.5^y)^{-N}}{N-1} \end{aligned}$$

❑

Bemerkung 1.6. Somit haben wir eine Möglichkeit gefunden, zumindest den Verfahrensfehler analytisch abzuschätzen. Um eine Genauigkeit von 200 Nachkommastellen zu erreichen, bräuchte man bei $y = \pi - 2$ beispielsweise nur die ersten 1000 Glieder aufsummieren, bei $y = \pi - 1$ genügen schon weniger als 600.

Die Reihe wurde in PARI/GP mit einer Genauigkeit von 10000 Nachkommastellen berechnet, wobei nur die ersten 50.000 Elemente berücksichtigt wurden. Somit ergibt eine obere Abschätzung für den Verfahrensfehler von: $V \leq 2 \cdot \frac{2.25}{(pi-3)\cdot(49999)^3} \cdot (1.5^{(\pi-2)})^{-50000} \leq 10^{-10000}$. Dies bedeutet, dass bei diesen Zahlen nur die Rundungsfehler zum Tragen kommen, der Verfahrensfehler aber nicht. Da ab dem 20.000 Element der Zetafunktion diese Werte kleiner als 10^{-10016} sind, werden sie nicht mehr zur berechneten Zahl addiert. Dies ändert aber nichts an unserer Genauigkeit, weil wir wissen, dass die Summe jener Elemente, welche numerisch gesehen 0 sind und hinzugefügt werden sollten, geringer als $30000 \cdot 10^{-10016} \leq 10^{-10011}$ ist.

[1] Vgl. hierzu auch Wolfram Research.

Proposition 1.7. *Der Wert der Reihe beträgt (auf* 5000 *Nachkommastellen gerundet):*

1.
4170252589450246466798947199540426487202893996777526791665026212097223036729279228823074004516422568
7961089332754335640671640263722796672446549688786446744955022457395947107941232376246223753770875706
8676266554210971436934698071100287106377764692948694076190426590074872389061131899391269173751064835
0865402353419018903823058441043554205141792861671409344342790252203399659531918378719618961889033504
6976233294967019365733952012567827307936971814680853706198911819811566086732240615668920901137565701
6696956826221651054329155650243562157089140998466842511698119933815090947662098348612094755490049752
1446719602841165740857230593461294213905805007053845343878087737060584395400649248613301876108178070
4451940960570964557940854548649122910825853130041490527182625736011993328330168799622852610410751374
4347423751586544570461846410651378977066623394426505640369897246203083477436866412369548331226372680
9301807388009239074437561837247709631236155450830264966322648989942371535293843529018119041157634513
5151977091192415121787052614610970449656567634728948936791805594653924810882372702483334504598493183
9650154377134834461070017595090989366000366981574741189716594674981919724276355131936182478969806026
8305014486161084099040661750783717793840623008857046545819436958465057687259953199873519862447713264
1212387607157025139281272616646977284835112476904799703190866673468145837298620147333781274935059617
1920820142639504414065468496711006109128947579907121785344365357293716771142568298848847502039975475
7734058878480847854043409728529107928868065530039359407295000543666349203935996668516348796363315359
2503477655033618040942474231857674867860993989752315626653430422968478433521787217870219360068749398
3533829183560285757713549479981292858250259161860919493257822437048021340610660570153377652507583570
1765180230498836483459050739264360908029655569322323582411349074339640439267437638055062399723111413
9191227005274709180570355228793173967542351141625360625918279332657006184318075633562364759132575359
7343859185982714269682946521300444485741228534865927031778897084693303175950649380276064172148673253
0562247041578841554523640480900963804523361194816990371768189842875546548313050810528807434174847734
5700576899098545688654667150231439063439170649932955220483649824893815499445432219395142528331791299
8590638581704156857730967172490103425480996043270252245785266623477051202765615559957935106451795578
9074528383443676667548602983233560801909043335787361399802931425987693399513296366098960849792375757
4646059804089019393028231343354907807062990083615434311992720969870011513831133332885750102929195121
2955203404415796097201304799641745725056438518868349892979183846661120523465609202140091649461623991
2150323956703671793408530316093543307617638045743697727006051279579607383148093804991887173156788968
0030518144625515992617689799863311968403374150162268808189369671231654493551056615443087410865898164
4111495440245338649528637627231628545292057778167911939933680035679191124677803156644474423022723813
1259289010659861767006531707842856432657176590273811205835861722476846022856405596865622177693713418
4330001919550618694417483954262224189231165962313587041734123961524138412903467507521158172154544137
4117311893262235386928274975386122117671524617058253047510132336822174103812729753431486964107741992
7166381748677558096442942729492506025050105781006880187404567433846776398773863983771842498630108161
1046250983562862670831104934076170227839559239412111901540745968712176717126583792099047587939324393
3700739357552788714323060044790673089544213720151756682842645453394761707706805559319392194556175078
3717794280037699759504867147223048551271923525007381528437204660941809560405521456163518259918146565
8996026373880434108259950426033910903478586178662204338003516609897935914877657442927094363355868298
7022101104175443717730099094911401543288098222240127355108960958715898285680969828596672540327985371
9916102309182000614911433478834432043114449560566676476496730398712859282403625857887532108993963189
5543405763846481301731267997669216414254087952141184813086159889814656653537677263561796706656116336
2871075048172452046064247659298358576197642375885210922842894545463403836007875732916105316984870931
2723917449644281882999780190751984571037444947093015379246203451690475146209230367487609789404148652
3636039200954952022390105879485827476816565647808126934137988373478199908161790712673717177922845819
1707125674277050393091751189305098853751228493834029181103820864850859266727606401451995903259472390
2322667346457942412626388388642336842556340847414948112033082106851283455675364179907846603988928831
4077203612622759663536201376103860444147914878648987842546846709236506709050824829027828079052402004
3078518609336783234949290089233281420541851854138456731508459487574176636000978799565429968157471396
1750606173768388516095070424198709633399246495647141577159823933220218716359094804181662958435689655
7919923438687156228699330998522497378539080186594080940815841857655768533344539016321023681876355925
2

2 Matrixnorm

2.1 Aufgabenstellung

Ein lineare Operator $A : \ell^2 \to \ell^2$ ist gegeben durch die unendliche Matrix $(a_{i,j})_{i,j\in\mathbb{N}}$, wobei $a_{i,j} = \frac{1}{i+j-1}$ für $i + j \in \mathbb{P}$ und $a_{i,j} = \frac{1}{(i+j-1)^2}$ sonst. Gesucht ist die Abbildungsnorm

$$\|A\|_2 = \sup_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

Einer der naheliegendsten Ansätze ist die unendliche Matrix durch endliche Matrizen zu approximieren und die Abbildungsnormen der endlichen Matrizen als Näherung zu bestimmen. Um dieses Verfahren zu rechtfertigen müssten die Folge der Abbildungsnormen der endlichen Matrizen gegen die Abbildungsnorm des Operator konvergieren. Dies und weitere Resultate werden wir im folgenden Abschnitt beweisen.

2.2 Vorbereitung

Die endlichen Matrizen werden nur als solche interpretiert formal sind es endlichdimensionale lineare Operatoren des ℓ^2.

Definition 2.1. Die Folge der endlichdimensionalen Operatoren wird mit $(A_n)_{n\in\mathbb{N}}$ bezeichnet wobei

$$(A_n)_{i,j} = \begin{cases} a_{i,j} & i,j \leq n \\ 0 & \text{sonst} \end{cases}$$

Bemerkung 2.2. Der Operator A ist selbstadjungiert und die A_n ebenfalls. Beziehungsweise als Matrix interpretiert sind die A_n symmetrisch.

Satz 2.3. *Die Abbildungsnorm des Operators ist endlich und mit π beschränkt und die Folge $(\|A_n\|)_{n\in\mathbb{N}}$ konvergiert monoton gegen $\|A\|$.*

Beweis Beschreibe P_n die Orthogonalprojektion auf den Unterraum der von den ersten n Einheitsvektoren aufgespannt wird, dann lässt sich A_n schreiben als P_nAP_n. Da die Einheitsvektoren eine Orthogonormalbasis bilden gilt $\lim_{n\to\infty} P_nx = x$, daher lässt sich folgende Ungleichung formulieren.

$$\|Ax\| = \lim_{n\to\infty} \|P_nAP_nx\| = \lim_{n\to\infty} \|A_nx\| \leq \lim_{n\to\infty} \|A_n\| \, \|x\|$$

Daraus folgt, dass die Abbildungsnorm von A kleiner als $\lim_{n\to\infty} \|A_n\|$ ist.

Für $m \leq n$ gilt dass $P_mP_n = P_nP_m = P_m$ da das Bild der Projektion P_m im Bild von P_n enthalten ist. Außerdem gilt für die Projektionen, dass ihre Abbildungsnorm mit 1 beschränkt ist. Wegen der Submultiplikativität der Abbildungsnorm folgt

$$\|A_m\| = \|P_mP_nAP_nP_m\| \leq \|P_m\|^2 \, \|P_nAP_n\| \leq \|A_n\| \leq \|A\| \, .$$

Nachdem man nun Ungleichungen in beide Richtungen hat konvergiert $(\|A_n\|)_{n\in\mathbb{N}}$ gegen $\|A\|$ und das sogar monoton. Zur Erinnerung gibt es folgende Identität für Abbildungsnormen

$$\|A_n\| = \sup\{(A_n x, y) \,|\, x, y \in \ell^2,\ \|x\| = \|y\| = 1\}.$$

Da alle Einträge der Matrix positiv sind reicht es das Supremum über x, y mit positiven Einträgen laufen zu lassen. Dadurch lässt sich mit Hilfe der Hilbert Ungleichung [3, Problem 10.1] folgende Abschätzung machen

$$(A_n x, y) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j} x_i y_i \leq \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \frac{x_i y_i}{i+j-1} \leq \pi \|x\| \|y\| .$$

Daraus ergibt sich $\|A_n\| \leq \pi$ für alle $n \in \mathbb{N}$ und damit gilt die selbe Abschätzung für $\|A\|$.

❑

Konvergenz in der Operatornorm hat man damit nicht gezeigt. Der Versuch das nachzuweisen hat als Resultat leider nur einige vollgeschriebene Zettel hervorgebracht. Da meist in irgendeiner Form die Hölderungleichung verwendet wurde, endeten die meisten Versuchen bei der Divergenten Reihe $\sum_{i=N}^{\infty} \frac{1}{i}$.

Auch der Versuch mit den Abschätzungen, die im Beweis der Hilberts Inequality verwendet werden, bessere Abschätzungen oder Operatornormkonvergenz zu erhalten, scheiterte.

Die Matrizen A_n sind *Hankelmatrizen*. Das heißt jeder Eintrag $(A_n)_{i,j}$ hängt nur von der Summe $i + j$ ab. Solche Matrizen kann man speicherplatzsparend in einen Vektor der Länge $2n$ speichern, da n auch die Spalten- und Zeilendimension der Matrix A_n angibt.

Um nun die Abbildungsnorm von $\|A_n\|$ für ein konkretes n zu bestimmen, sei erwähnt, dass A_n als symmetrische Matrix, normal und diagonalisierbar ist. Daher stimmt die Abbildungsnorm von A_n mit dem betragsgrößten Eigenwert überein. Zur Berechnung des größten Eigenwertes bietet sich die Potenzmethode an, da diese für diagonalisierbare Matrizen sicher konvergiert.

Neben der Vektoridentifikation haben Hankelmatrizen auch die nette Eigenschaft, dass sich die Matrixvektormultiplikation in $O(n \log n)$ durchführen lässt, indem man eine *Fast-Fourier-Transformation* verwendet [2].

2.3 Auswerten

Wir wenden nun dieses Verfahren für steigende Dimensionen an und hoffen so anhand der Entwicklung der Zahlen abschätzen zu können, wieviele Nachkommastellen genau sind.

Trotz der vermuteten Konvergenzordnung von $\mathbb{O}(\frac{1}{\sqrt(n)})$, wobei n der Dimension entspricht, zeigt sich hier ein schnelleres Stabilisieren der Nachkommastellen. Wie aus Tabelle 1 ersichtlich, bleibt bei jedem Zehnersprung in der Potenz auch eine weitere Stelle 'stehen'. Erst zum Ende, wenn die Dimensionen nur mehr verdoppelt werden, benötigt man 3 dieser Erhöhungen für die nächste Stelle, was wegen $2^3 = 8$ durchaus mit der Hypothese einer schnelleren Konvergenzordnung übereinstimmt.

In Tabelle 1 haben wir der Übersichtlichkeit halber für niedrigere Dimensionen immer einen Dimensionssprung in der Zehnerpotenz gemacht. Des Weiteren

Dimension	Ergebnis
10^0	**1**.0000000000
10^1	**1.39**24202045
10^2	**1.395**501314
10^3	**1.3957**48167
10^4	**1.3957**69812
10^5	**1.395771**776
10^6	**1.3957719**55
2^{23}	**1.3957719**63
2^{24}	**1.3957719**68
2^{25}	**1.3957719**71

Tabelle 1: Werte für $\|A_n\|$

haben wir dasselbe für die Dimensionen $2^n, n \in [0, 25] \cap \mathbb{N}$ gerechnet und auf diese Werte den Wynn'schen Epsilon-Algorithmus angewandt. Dieser liefert für den Extrapolationswert

$$\widehat{\|A\|} = 1.395771973657402$$

Außerdem liefert er für die einzelnen Schritte die Fehlerentwicklung. Dies ist graphisch in Abbildung 1 dargestellt.

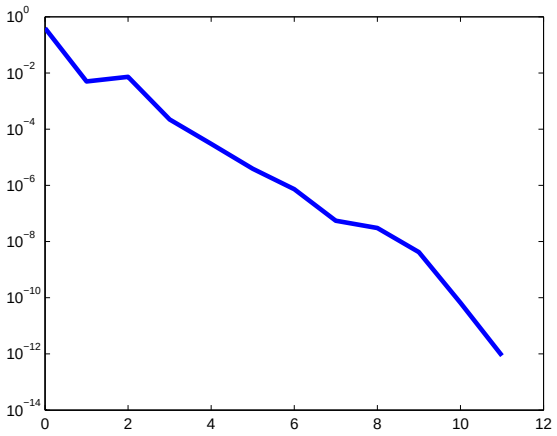


Abbildung 1: Fehlerentwicklung beim Wynn'schen Epsilon Algorithmus

3 Stochastische Folge

3.1 Aufgabenstellung

Die Folge $(x_n)_{n\in\mathbb{N}}$ ist rekursiv definiert durch

$$x_{n+3} = x_{n+2} \pm \frac{1}{3}(x_{n+1} + x_n), \quad x_0 = x_1 = x_2 = 1,$$

wobei das Vorzeichen $\pm$ mit Wahrscheinlichkeit $\frac{1}{2}$ wechselt.

Berechnen Sie

$$\sigma := \lim_{n\to\infty} |x_n|^{\frac{1}{n}}$$

Bemerkung 3.1. Zunächst einmal ist die Aufgabenstellung nicht eindeutig: Es steht lediglich, dass das Vorzeichen mit Wahrscheinlichkeit $\frac{1}{2}$ positiv oder negativ ist, allerdings nichts über die Abhängigkeiten der Vorzeichen voneinander, bzw. von den Folgengliedern. Eine mögliche Definition wäre es, das erste Vorzeichen durch eine Zufallsvariable auf $\{-1, 1\}$ mit den Punktwahrscheinlichkeiten $\frac{1}{2}$zu modellieren und danach alle anderen Vorzeichen gleich dem ersten zu setzten. Dann gilt für das i-te Vorzeichen $\mathbb{X}_i$:

$$\mathbb{P}(\mathbb{X}_i = +1) = \mathbb{P}(\mathbb{X}_1 = +1) = \frac{1}{2}$$

Wie man sich leicht überzeugen kann, nimmt im Falle eines positiven Vorzeichens σ den (gerundeten) Wert 1.4058 an und im umgekehrten Fall den Wert $\frac{1}{3}(\sqrt[3]{10} - 1) \approx 0.3848$ an. Somit erhalten wir für den Grenzwert:

$$\lim_{n\to\infty} \|x_n\|^{\frac{1}{n}} \to^d \widehat{\sigma}, \widehat{\sigma}(\Omega) = \{1.4058, 0.3848\}, \mathbb{P}(\widehat{\sigma} = 1.4058) = \frac{1}{2} \tag{4}$$

Davon möchten wir aber absehen und verstehen die Aufgabe wie folgt: Seien $(\mathbb{X}_n)_{n\in\mathbb{N}}$ unabhängig, identisch verteilte Zufallszahlen auf $\{-1, 1\}$ mit den Punktwahrscheinlichkeiten $\frac{1}{2}$. Dann gilt für die rekursive Darstellung:

$$x_{n+3} = x_{n+2} + \mathbb{X}_{n+3}\frac{1}{3}(x_{n+1} + x_n) \tag{5}$$

3.2 Analytischer Zugang

Bemerkung 3.2. Wie man leicht mittels vollständiger Induktion nachweist, gilt auch folgende Darstellung[2] für x_n:

$$x_n = 1 + \sum_{i=2}^{n-1} \frac{\mathbb{X}_n}{3} \left(x_{n-i} + x_{n-i-1}\right), \; n \geq 3$$

[2]Definiert man die Summe über die leere Menge wie üblich durch 0, so stimmt obige Gleichheit sogar für beliebige $n \in \mathbb{N}$

Diese Darstellung verdeutlicht auch die Unabhängigkeit der x_{n-i} von $\mathbb{X}_n$ für alle positiven i. Ein weiterer Vorteil dieser Darstellung besteht darin, dass die einzelnen Summanden paarweise unkorreliert sind und Erwartungswert 0 besitzen. Wie sich zeigen wird, können wir sogar deren zweiten Momente explizit bestimmen.

Wegen der Linearität des Erwartungswertes gilt auch, dass dieser für alle Folgenglieder konstant gleich 1 ist. Würden die Varianzen der Summanden nicht exponentiell ansteigen, wie sich später herausstellen wird, könnte man nun einfach ein Gesetz der großen Zahlen für unkorrelierte Zufallszahlen anwenden.

Lemma 3.3. Sei $(w_n)_{n\in\mathbb{N}}$ eine rekursiv definierte Folge auf den reellen Zahlen mit $w_n = w_{n-1} + \frac{w_{n-2}}{9} + \frac{w_{n-3}}{3}$ und den Startwerten $w_0 = w_1 = w_2 = 1$.

Bezeichnet

$$\gamma := \frac{1}{27}\left(9 + \sqrt[3]{4374 - 162\sqrt{681}} + 3\sqrt[3]{6(27+\sqrt{681})}\right) \approx 1.2874$$

so können wir wie folgt abschätzen:

$$\gamma^{n-2} \leq w_n \leq \gamma^n \quad \text{für alle } n \in \mathbb{N} \tag{6}$$

Beweis γ ist die einzige reelle Nullstelle des Polynoms $x^3 - x^2 - \frac{x}{9} - \frac{1}{3}$. Daher gilt klarerweise $\gamma^2 + \frac{\gamma}{9} + \frac{1}{3} = \gamma^3$. Wir beweisen zunächst die Abschätzung $w_n \leq \gamma^n \ \forall n \in \mathbb{N}$ mittels vollständiger Induktion nach n.

Induktionsanfang: Für $i \in \{0,1,2\}$ gilt $w_i = 1 \leq \gamma^i$.

Induktionsschritt: Gelte nun $w_k \leq \gamma^k$ für alle $k \leq n$. Dann lässt sich folgende Ungleichung machen

$$w_{n+1} := w_n + \frac{1}{9}w_{n-1} + \frac{1}{3}w_{n-2} \leq \gamma^{n-2}(\gamma^2 + \frac{1}{9}\gamma^1 + \frac{1}{3}) = \gamma^{n+1}$$

Die zweite Abschätzung werden wir erneut mittels vollständiger Induktion beweisen. Der Induktionsanfang erfolgt durch $w_i = 1 \geq \gamma^{i-2}$ für $i \in \{0,1,2\}$. Für den Induktionsschritt machen wir eine analoge Umformung wie oben und erhalten:

$$w_{n+1} \geq \gamma^{n-4}(\gamma^2 + \frac{1}{9}\gamma^1 + \frac{1}{3}) = \gamma^{n-1}$$

❑

Lemma 3.4. Sei $(x_n)_{n\in\mathbb{N}}$ wie in (5) definiert und sei v_n die rekursiv definierte Folge aus 3.3. Dann ist $\mathbb{E}(x_n^2) = v_n$. Für das asymptotische Verhalten ergibt sich:

$$\lim_{n\to\infty} \left[\mathbb{E}(x_n^2)\right]^{\frac{1}{n}} = \gamma \tag{7}$$

Beweis Sei $(w_n)_{n\in\mathbb{N}} := \mathbb{E}(x_n)^2$. Wir wollen nun zeigen, dass die beiden Folgen w_n und v_n übereinstimmen. Dazu werden wir eine Rekursionsformel für w_n herleiten und zeigen, dass diese und die Startwerte der Folgen übereinstimmen.

Da $\mathbb{X}_n$ unabhängig von x_i $\forall i \leq n-1$ ist und Erwartungswert 0 hat, folgt zunächst durch Einsetzen der Rekursion:

$$\mathbb{E}(x_i x_{i-1}) = \mathbb{E}(x_{i-1}^2) + \frac{1}{3}\mathbb{E}\left(\mathbb{X}_n x_{i-1}(x_{i-2} + x_{i-3})\right) \tag{8}$$

$$= \mathbb{E}(x_{i-1}^2) + \frac{1}{3}\mathbb{E}(\mathbb{X}_n)\mathbb{E}\left(x_{i-1}(x_{i-2} + (x_{i-3})\right) = \mathbb{E}(x_{i-1}^2) \tag{9}$$

Bringt man in (5) x_{n-1} auf die linke Seite, quadriert beide Seiten und betrachtet den Erwartungswert erhält man die Gleichheit

$$\mathbb{E}\left[(x_n - x_{n-1})^2\right] = \mathbb{E}[(\frac{1}{3}\mathbb{X}_n(x_{n-2} + x_{n-3}))^2]$$

Der linke Seite entspricht jedoch genau $w_i - w_{i-1}$, weil sich das unter Berücksichtigung von (8) auch wie folgt schreiben lässt:

$$w_i - w_{i-1} := \mathbb{E}\left[x_n^2\right] - \mathbb{E}\left[x_{n-1}^2\right] =$$
$$\mathbb{E}\left[x_n^2\right] - 2\mathbb{E}\left[x_{n-1}x_n\right] + \mathbb{E}\left[x_{n-1}^2\right] = \mathbb{E}\left[(x_n - x_{n-1})^2\right]$$

Die rechte Seite kann unter erneuter Ausnützung der Unabhängigkeit der $\mathbb{X}_i$ und (8) umgeformt werden zu:

$$\mathbb{E}\left[\left(\frac{1}{3}\mathbb{X}_n(x_{n-2} + x_{n-3})\right)^2\right] = \frac{\mathbb{E}(\mathbb{X}_n^2)}{9}\mathbb{E}\left(x_{n-2}^2 + x_{n-3}^2 + 2x_{n-2}x_{n-3}\right) =$$
$$\frac{1}{9}\mathbb{E}\left(x_{n-2}^2 + 3x_{n-3}^2\right) = \frac{w_{n-2}}{9} + \frac{w_{n-3}}{3}$$

Somit können wir w_i auch durch $w_i = \mathbb{E}(x_i)^2 = x_i^2 = 1 = v_n,\ i \in \{0,1,2\}$ und der Rekursion

$$w_i = w_{i-1} + \frac{w_{n-2}}{9} + \frac{w_{n-3}}{3}\ \forall i \geq 3$$

ausdrücken. Dies entspricht jedoch genau der rekursiven Definition von v_n. Somit stimmen die beiden Folgen überein. Behauptung (7) folgt nun unmittelbar aus 3.3 und $\lim_{n\to\infty} \gamma^{\frac{n-2}{n}} = \gamma$.

❑

Satz 3.5. *Für den Grenzwert σ, wie in (5) definiert, gilt folgende Abschätzung:*

$$\sigma \leq \sqrt{\gamma} < 1.1347 \tag{10}$$

Beweis Zunächst stellen wir $\|x_n\|^{\frac{1}{n}} \geq 0$ fest und können nach dem Lemma von Fatou den Grenzwert aus dem Erwartungswert herausziehen.

$$\sigma = \mathbb{E}(\sigma) = \mathbb{E}(\liminf_{n\to\infty} \|x_n\|^{\frac{1}{n}}) \leq \liminf_{n\to\infty} \mathbb{E}(\|x_n\|^{\frac{1}{n}}) \leq \lim_{n\to\infty} \left(\mathbb{E}(x_n^2)\right)^{\frac{1}{2n}}$$

Für die letzte Zeile wurde die Ungleichung von Jensen verwendet. Aus der Stetigkeit der Wurzelfunktion und Lemma 3.4 ergibt sich nun die Abschätzung

$$\sigma \leq \sqrt{\lim_{n\to\infty} \left(\mathbb{E}(x_n^2)\right)^{\frac{1}{n}}} = \sqrt{\gamma}$$

❑

Proposition 3.6. *Die stochastische Folge $(x_n)_{n\in\mathbb{N}}$, versehen mit ihrer natürlichen Filtration $\mathbb{A}_n$, bildet ein Martingal. Daher gilt auch $\mathbb{E}x_n = \mathbb{E}x_1 = 1\ \forall n \in \mathbb{N}$.*

Beweis Aus der Existenz des zweiten Moments folgt auch die Integrierbarkeit der x_n. Für den bedingten Erwartungswert folgt:

$$\mathbb{E}(x_n|\mathbb{A}_n) = x_{n-1} + \mathbb{E}(\mathbb{X}_n)(x_{n-2} + x_{n-3}) = x_{n-1}\ \forall n \in \mathbb{N}$$

Hierbei wurde die Eigenschaft $\mathbb{E}\mathbb{X}_n = 0$ und deren Unabhängigkeit von $x_i, i < n$ verwendet. Die letzte Behauptung folgt nun aus: $\mathbb{E}x_n = \mathbb{E}(\mathbb{E}(x_n|\mathbb{A}_n)) = \mathbb{E}x_{n-1}$, vollständiger Induktion nach n und dem Umstand, dass $\mathbb{E}x_1 = \mathbb{E}1 = 1$. ❑

Bemerkung 3.7. Aus der verallgemeinerten Ungleichung von Kolmogoroff für Submartingale folgt nun sogar wegen:

$$\mathbb{P}\left(\max_{1\leq i\leq n} \|x_i\| \geq (\sqrt{\gamma} + \epsilon)^n\right) \leq (\sqrt{\gamma} + \epsilon)^{-2n}\mathbb{E}x_n^2$$

die Aussage

$$\lim_{n\to\infty} \mathbb{P}\left(\max_{1\leq i\leq n} \|x_i\|^{\frac{1}{n}} \geq \sqrt{\gamma} + \epsilon\right) = 0\ \forall \epsilon > 0$$

3.3 Numerischer Lösungsansatz

Bezeichne in diesem Abschnitt

$$y_n := \|x_n\|^{\frac{1}{n}}$$

Ziel der folgenden Überlegungen wird es sein, ohne allzu großes Vorwissen über die Verteilung dieser Zufallsvariable ein Konfidenzintervall aufzustellen.

Konvergiert y_n gegen eine Konstante, so liegt die Vermutung nahe, dass der Erwartungswert sich ebenfalls dieser Konstante nähert und die Varianz verschwindet, also $\mathbb{V}(y_n)$ eine Nullfolge bildet. Analytisch lässt sich aus obigen Sätzen nur folgende, schwächere Aussage ableiten:

$$\mathbb{V}(y_n) := \mathbb{E}(y_n^2) - (\mathbb{E}y_n)^2 \leq \mathbb{E}\left((x_n^2)^{\frac{1}{n}}\right) \leq \left(\mathbb{E}x_n^2\right)^{\frac{1}{n}} \leq \gamma$$

Zieht man nun eine Stichprobe bzw. simuliert mehrere Realisationen dieser Zufallsvariable und berechnet den Mittelwert, so ändert sich der Erwartungswert

nicht, die Varianz nimmt jedoch indirekt proportional zur Stichprobengröße ab. Da wir die exakte Verteilung von y_n nicht kennen, greifen wir auf eine allgemein gültige Ungleichung (Tschebyscheff) zurück:

$$\mathbb{P}\left(\|\overline{y_n} - \mathbb{E}(y_n)\| \geq \epsilon\right) \leq \frac{\mathbb{V}(\overline{y_n})}{\epsilon^2} \leq \frac{\gamma}{n\epsilon^2}$$

Somit können wir aus obigen Überlegungen folgende Aussage schlussfolgern:

Proposition 3.8. *Mit* $C_\alpha := \left(\frac{\gamma}{\alpha}\right)^{\frac{1}{2}}$*, sowie den Bezeichnungen des obigen Abschnitts bildet*

$$\left[\ \overline{y_n} - \frac{C_\alpha}{\sqrt{n}}\ ,\ \overline{y_n} + \frac{C_\alpha}{\sqrt{n}}\ \right]$$

ein Konfidenzintervall mit einer Überdeckungswahrscheinlichkeit von mindestens $1 - \alpha$*.*

3.4 Auswertung

Da wir für diese Auswertung ein sehr große Anzahl an Zufallszahlen benötigen, haben wir dieses Programm nicht nur in Matlab, sondern auch in dem Statistik Programm R programmiert. Interessanterweise kamen bei unseren Simulationsläufen für das 2^16-te bzw. 10^5-te Folgenglied (Matlab/R) mit 2^13 bzw. 50000 Simulationen recht unterschiedliche Schätzwerte heraus. Wie eine Analyse anderer Folgenglieder in den jeweiligen Programmen ergab, lag dieser Unterschied nicht an dem tatsächlich berechneten Folgenglied. Wir schließen daraus, dass eines der beiden Programme in diesem Bereich numerisch bereits instabil arbeitet. In Matlab ergab sich ein Schätzwert von 1.022, während in R sich selbst bei unterschiedlichen Folgendliedern ein Wert um 1.0091 ergab. Somit vermuten wir, dass der tatsächliche Wert mit $1.0\ldots$ beginnt.

3.5 Weiterführende Überlegungen

Im hiesigen Abschnitt betrachten wir eine mögliche Verallgemeinerung von (5):

$$x_n^{(\delta)} := x_{n-1}^{(\delta)} + \delta\mathbb{X}_n(x_{n-2}^{(\delta)} + x_{n-3}^{(\delta)}),\ n \geq 3 \tag{11}$$

$$x_0^{(\delta)} = x_1^{(\delta)} = x_2^{(\delta)} := 1 \tag{12}$$

Für $\delta = \frac{1}{3}$ ergibt sich genau die uns bekannte Folge. Da $(-\mathbb{X}_n)$ der gleichen Verteilung folgt wie $\mathbb{X}_n$, ist der Grenzwert $\sigma^{(\delta)} := \lim_{n\to\infty} \|x_n^{(\delta)}\|^{\frac{1}{n}}$ - so er überhaupt existiert - symmetrisch um die Null. Des Weiteren lassen sich die Beweise des obigen Abschnittes für $\delta \in [-2, 2]$ abwandeln [3]. Das zur Abschätzung der Varianz benötigte $\gamma^{(\delta)}$ berechnet sich im allgemeinen Fall als Nullstelle des Polynoms $x^3 - x^2 - \frac{x+3}{\delta^2}$. Zieht man nun noch die Wurzel aus diesem Wert, so erhält man ein Abschätzung nach oben für den Grenzwert $\sigma^{(\delta)}$ in Abhängigkeit von δ. Dieser Zusammenhang ist in Abbildung 2 graphisch dargestellt. Es zeigt sich auch deutlich, dass eine Abschätzung für $\sigma^{(\delta)}$ durch 1 nur für $\delta = 0$ möglich ist, weil in diesem Fall die ursprüngliche Folgen konstant bleibt.

[3] Zumindest solange das zu betrachtende Polynom eine eindeutige Nullstelle echt größer 1 hat, können die Beweise übernommen werden. Dies ist für das oben angegebene Intervall tatsächlich der Fall, auch wenn man es durchaus noch vergrößern könnte.

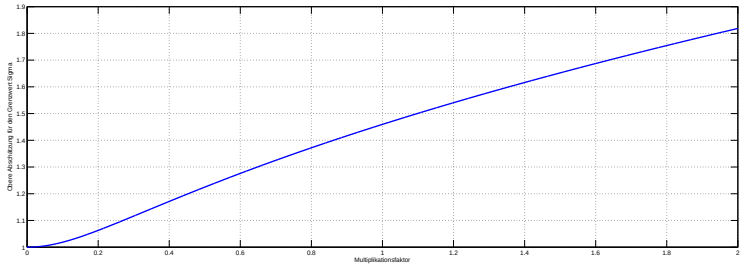


Abbildung 2: Obere Abschätzung für $\sigma^{(\delta)}$

3.6 Irrwege

Wir wissen für die Folge der Zufallsvariablen x_n, dass sie konstanten Erwartungswert hat, kennen ihre Varianz und dass sie ein Martingal bildet. Definiert man nun ein Folge $y_{n,j} := (x_j - x_{j-1})\gamma^{-\frac{n}{2}}$, so ist deren Varianz nahezu konstant und ihr Erwartungswert 0. Des Weiteren handelt es sich um eine Dreiecksfolge von Martingaldifferenzen, für die es einen zentralen Grenzwertsatz gibt. Dieser würde besagen, dass $x_n\gamma^{-\frac{n}{2}}$ gegen eine Standardnormalverteilung konvergiert. Man könnte dann weiters hoffen, so genauere Aussagen über σ treffen zu können. Leider ließ sich die konditionierte Lindeberg-Bedingung nicht zeigen. Auch statistische Tests ergaben für endliche Simulationen, dass keine Ähnlichkeit mit einer Normalverteilung vorliegt.

4 Oszillierendes Integral

Die vierte Aufgabe ist es das Integral I zu berechnen.

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

4.1 Einleitende Worte

Der Integrand ist bis auf abzählbar viele Stellen stetig und daher eine Lebesgue-meßbare Funktion. Außerdem lässt sich der Integrand mit 1 beschränken und die Menge, über die integriert wird, hat endliches Maß. Daraus folgt die Lebesgue-Integrierbarkeit und damit auch die Riemann-Integrierbarkeit. Als erste Abschätzung kann

$$\left\| \int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(\frac{1}{x})}\right)}\right) dx \right\| \leq (1-0) \sup_{x\in[0,1]} \left\| \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(\frac{1}{x})}\right)}\right) \right\| = 1$$

gemacht werden.

Betrachtet man Abbildung 3, würde man für das Integral einen Wert in der Nähe von 0.5 vermuten, da sich die Funktion ab $x = 0.4$ relativ gutmütig verhält

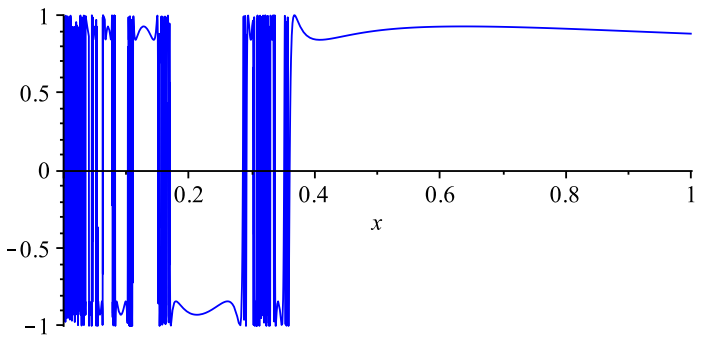


Abbildung 3: Ursprünglicher Integrand vor Transformation

und nahe 1 liegt. Außerdem gibt es ein Intervall mit einer Länge von ungefähr 0.1, auf der die Funktion sich ebenso verhält, allerdings mit umgedrehten Vorzeichen. Man könnte hoffen, dass sich im restlichen Intervall der Integrand wieder wegkürzt.

Als ersten Richtwert berechnen wir in MATLAB die Legendre-Gauß Knoten und deren Gewichte in unterschiedlichen Ordnungen für die Intervalle $[0, 1]$ und $[1, 1000]$. Beim ersten Intervall berechnen wir so das numerische Integral für die normale Funktion, beim zweiten Intervall verwenden wir eine Transformation der Funktion, welche wir später genauer erläutern werden. Wir brechen hier willkürlich bei der oberen Schranke 1000 ab, werden aber später in Bemerkung 4.2 mathematisch beweisen können, dass das restliche Integral geringer als $2 \cdot 10^{-6}$ ist. Somit könnten wir darauf hoffen, dass zumindest die ersten 5 Nachkommastellen mit dem Integral über das unbeschränkte Intervall übereinstimmen. In Tabelle 2 haben wir die Ergebnisse zusammengefasst und jene Nachkommastellen, welche mit unserem späteren Ergebnis übereinstimmen, fett gedruckt.

Ordnung	Ursprüngliches Integral	Transformiertes Integral
10^2	**0.4**89267188439361	**0.**357423390762866
10^3	**0.49**282719316007	**0.4**64987453167806
10^4	**0.4962**85293604327	**0.49**5399195407471

Tabelle 2: Werte bei naiver Integration

Überraschenderweise zeigt sich schon bei dieser naiven Herangehensweise, dass der Wert des Integrals etwa bei 0.49 liegen dürfte und die ursprüngliche Funktion immer um mindestens eine Nachkommastelle genauer als ihr transfomiertes Pendant ist. Dennoch wird sich die Transformation als fruchtbar herausstellen. Nimmt man an, dass mit zunehmender Ordnung die Werte zuverlässiger sind, so könnte man sogar die Vermutung aufstellen, dass der gesuchte Wert im Intervall $[0.495, 0.497]$ liegt. Erstaunlicherweise ist diese Aussage trotz des ein-

fachen Zugangs korrekt.

Somit stellt sich in diesem Beispiel nicht die Frage, ob man dieses Integral überhaupt sinnvoll auswerten kann, sondern mit welcher Genauigkeit dies geschehen kann. Die Herausforderung in diesem Beispiel ist es daher, analytische Aussagen über dieses Integral zu treffen, um so auch mathematisch exakt die Genauigkeit der berechneten Zahl nach oben abschätzen zu können. Dies werden wir nun im folgenden Abschnitt machen.

4.2 Vorbereitung

Um die unangenehmen Stellen der Funktion, die sich zur 0 häufen (siehe Abbildung 3) ein bisschen zu strecken kann eine Koordinatentransformation mit $\frac{1}{x}$ angewandt werden. Mit dieser erhält man

$$\int_0^1 \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(\frac{1}{x})}\right)}\right) dx = \int_1^\infty \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right) \frac{1}{x^2}\, dx.$$

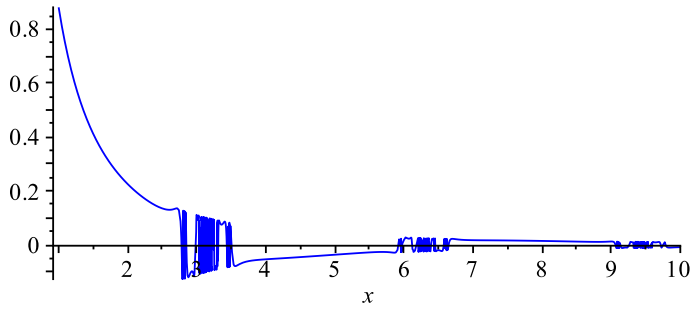


Abbildung 4: Integrand nach Transformation

Vernachlässigt man den Ableitungsterm $\frac{1}{x^2}$, so erhält man eine ungerade π-periodische Funktion. Weswegen das Integral über letztere Funktion auf jedem Intervall der Länge 2π verschwindet. Abbildung 5 veranschaulicht dies. Außerdem verschwindet das Integral auch für Intervalle der Form $[n\pi - c, n\pi + c]$. Für große n ist es daher naheliegend, dass das Integral über die transformierte Funktion (mit Ableitungsterm) über diese Intervalle nahe 0 ist. Satz 4.1 formuliert dies in einer mathematischen Aussage und gibt einen Fehlerschätzer.

Satz 4.1. *Sei* $f \in L_\infty([x-c, x+c], \mathfrak{B}([x-c, x+c], \lambda)$ *mit* $0 < c < x, c, x \in \mathbb{R}^+$. *Weiters gelte:* $\|f\|_\infty \leq 1$, f *sei bezüglich* x *eine* λ *fast überall ungerade Funktion, also* $f(x-y) = -f(x+y)$ *für alle* $y \in M$, *wobei* $M := [-c, c] \cap N^c$ *und* $\lambda(N) = 0$. *Dann gilt folgende Abschätzung:*

$$\| \int_{x-c}^{x+c} \frac{1}{y^2} f(y) dy \| \leq \frac{2xc^2}{x^2(x^2-c^2)} \leq \frac{2c^2}{x(x^2-c^2)} \sim O(\frac{1}{x^3}) \tag{13}$$

Beweis

$$\int_{x-c}^{x+c} \frac{1}{y^2} f(y) dy = \int_{-\frac{c}{2}}^{\frac{c}{2}} \frac{f(y+x-\frac{c}{2})}{(y+x-\frac{c}{2})^2} dy + \int_{-\frac{c}{2}}^{\frac{c}{2}} \frac{f(y+x+\frac{c}{2})}{(y+x+\frac{c}{2})^2} dy =$$

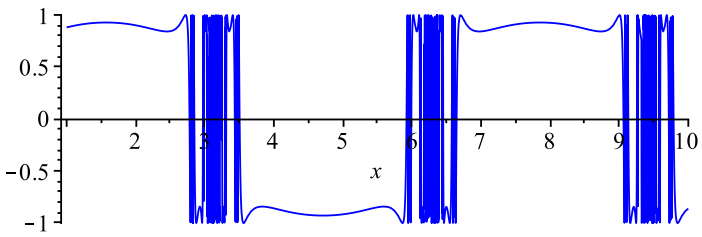


Abbildung 5: Integrand nach Transformation ohne Ableitungsterm $\frac{1}{x}$

Nun nützen wir für den ersten Integranden die schiefe Punktsymmetrie um x aus (da ja $y - \frac{c}{2} \in [-c, c]\ \forall y \in [-\frac{c}{2}, \frac{c}{2}]$ gilt).

$$= \int_{-\frac{c}{2}}^{\frac{c}{2}} \frac{-f(x-y+\frac{c}{2})}{(y+x-\frac{c}{2})^2} dy + \int_{-\frac{c}{2}}^{\frac{c}{2}} \frac{f(y+x+\frac{c}{2})}{(y+x+\frac{c}{2})^2} dy =$$
$$\int_{-\frac{c}{2}}^{\frac{c}{2}} f(y+x+\frac{c}{2}) \cdot \left(\frac{1}{(y+x+\frac{c}{2})^2} - \frac{1}{(x-y-\frac{c}{2})^2} \right) dy$$

Nun betrachten wir die Norm des Integrals und schätzen diese ab, wobei wir die Eigenschaft $\|f\|_\infty \leq 1$ ausnützen.

$$\| \int_{x-c}^{x+c} \frac{1}{y^2} f(y) dy \| \leq \int_{-\frac{c}{2}}^{\frac{c}{2}} \| \frac{1}{(y+x+\frac{c}{2})^2} - \frac{1}{(x-y-\frac{c}{2})^2} \| dy =$$
$$= \int_{-\frac{c}{2}}^{\frac{c}{2}} \frac{4x(y+\frac{c}{2})}{(x^2-(y+\frac{c}{2})^2)^2} dy = \int_0^c \frac{4xy}{(x^2-y^2)^2} dy =$$
$$= 2x \left(\frac{1}{x^2-c^2} - \frac{1}{x^2} \right) = \frac{2xc^2}{x^2(x^2-c^2)}$$

❑

Bemerkung 4.2. Die transformierte Funktion ohne Ableitungsterm erfüllt alle Bedingungen von Satz 4.1, wobei $x = n\pi, n \in \mathbb{N}$. Somit können wir gleich zwei Abschätzungen machen:

1. Berechnet man den Wert des Integrals nur bis zum Punkt $x = 2n\pi$, so lässt sich der restliche Term wie folgt abschätzen:

$$\| \int_{2n\pi}^{\infty} \frac{1}{y^2} f(y) dy \| \leq \sum_{k=n}^{\infty} \| \int_{2k\pi}^{(2k+2)\pi} \frac{1}{y^2} f(y) dy \| \tag{14}$$

Schreibt man noch $2k\pi$ als $(2k+1)\pi - \pi$ und nutzt die Schiefsymmetrie bei $(2k+1)\pi$ aus, so kann man Satz 4.1 mit $c = \pi$ anwenden und erhält als Abschätzung

$$(14) \leq \pi^3 \sum_{k=n}^{\infty} \frac{4k+2}{(2k\pi)^4} \leq \frac{1}{4\pi} \sum_{k=n}^{\infty} \frac{k+0.5}{k^4}$$

Letzteren Term teilen wir auf zwei Reihen auf und können gemäß Bemerkung 1.4 wie folgt abschätzen:

$$\| \int_{2n\pi}^{\infty} \frac{1}{y^2} f(y)dy \| \leq \frac{(n-0.5)^{(-2)}}{8\pi} + \frac{(n-0.5)^{(-3)}}{24\pi}$$

2. Aber auch als Abschätzung des Verfahrenfehlers bei jeder einzelnen Singularität lässt sich obiger Satz anwenden:

$$\int_{n\pi-\epsilon}^{n\pi+\epsilon} \frac{1}{y^2} f(y)dy \leq 2\epsilon^2 \frac{n\pi}{((n\pi)^2 - \epsilon^2)(n\pi)^2}$$

Der Unterschied zur naiven Abschätzung über die Supremumsnorm besteht einerseits darin, mit welcher Ordnung die Stelle, um deren Epsilon-Umgebung integriert wird, in der Abschätzung eingeht. Andererseits fließt hier die Intervalllänge quadratisch ein. Mit der klassischen Abschätzung erhielte man lediglich:

$$\int_{n\pi-\epsilon}^{n\pi+\epsilon} \frac{1}{y^2} f(y)dy \leq \frac{2\epsilon}{(n\pi - \epsilon)^2}$$

Somit unterscheiden sich die beiden Abschätzungen genau um den Term $\frac{\epsilon}{2n\pi} < 1$.

Satz 4.3. *Die Nullstellen von* $\sin\left(\frac{1}{\sin(\frac{1}{\sin(x)})}\right)$ *sind gegeben durch*

$$\pi \cdot m - \arcsin\left(\frac{1}{\pi \cdot n - \arcsin(\frac{1}{k\pi})}\right) \qquad m, n, k \in \mathbb{Z},\ n, k \neq 0$$

Beweis Die Nullstellen der Sinusfunktion sind genau bei ganzzahligen Vielfachen von π:

$$\sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right) = 0 \Leftrightarrow \frac{1}{\sin\left(\frac{1}{\sin(x)}\right)} = k\pi, \quad k \in \mathbb{Z}$$

Da der Zähler des Bruches immer eins ist kann der Fall $k = 0$ ausgeschlossen werden. Daher kann auch der Kehrwert $\sin\left(\frac{1}{\sin(x)}\right) = \frac{1}{k\pi}$ gebildet werden. Daraus ergibt sich für $n \in \mathbb{Z}$ folgende Gleichung für die Nullstellen

$$\frac{1}{\sin(x)} = 2\pi \cdot n + \arcsin(\frac{1}{k\pi}) \quad \text{oder} \quad \frac{1}{\sin(x)} = 2\pi \cdot n + \pi - \arcsin(\frac{1}{k\pi})$$

Da $k \in \mathbb{Z}\backslash\{0\}$ lassen sich diese zwei Fälle zu

$$\frac{1}{\sin(x)} = \pi \cdot n - \arcsin(\frac{1}{k\pi}) \quad k \in \mathbb{Z}\backslash\{0\}$$

zusammenfassen. Nimmt man nochmals den Kehrwert und löst die Gleichung auf so erhält man

$$x = 2\pi \cdot m + \arcsin\left(\frac{1}{\pi \cdot n - \arcsin(\frac{1}{k\pi})}\right) \quad \text{oder}$$

$$x = 2\pi \cdot m + \pi - \arcsin\left(\frac{1}{\pi \cdot n - \arcsin(\frac{1}{k\pi})}\right)$$

wobei für $n = 0$ das Argument im arcsin größer als 1 ist und damit im reellen nicht auflösbar. Diese zwei Fälle lassen sich wieder durch herausheben des Vorzeichens aus den arcsin Funktionen zur Aussage zusammenfassen.

❑

Bemerkung 4.4. Das Vorzeichen von n bestimmt, ob die Nullstelle rechts oder links von $m\pi$ ist und das Vorzeichen von k bestimmt, rechts oder links von $m\pi - \arcsin(\frac{1}{n\pi})$. Positives bedeutet links, negatives Vorzeichen rechts.

Satz 4.5. *Die Unstetigkeitsstellen*[4] *von* $\sin\left(\frac{1}{\sin(\frac{1}{\sin(x)})}\right)$ *sind bei*

$$\pi \cdot m,\ m \in \mathbb{N} \quad \textit{und} \quad \pi \cdot m - \arcsin(\frac{1}{n\pi}),\ n, m \in \mathbb{Z}$$

Beweis Da die Funktion eine Komposition aus sin und $\frac{1}{(.)}$ ist können Unstetigkeitsstellen nur dort auftreten, wo $\frac{1}{(.)}$ nicht definiert ist. Also genau dann wenn

$$\sin(x) = 0 \quad \text{oder} \quad \sin(\frac{1}{\sin(x)}) = 0$$

Im ersten Fall sind das genau die ganzzahliges Vielfachen von π. Im zweiten Fall sind das genau die Werte wo $\frac{1}{\sin(x)}$ ein ganzzahliges Vielfaches von π ist. Daher

$$\sin(x) = \frac{1}{n\pi} \quad \Leftrightarrow \quad \begin{cases} x = 2\pi \cdot m + \arcsin(\frac{1}{n\pi}) \\ x = 2\pi \cdot m + \pi - \arcsin(\frac{1}{n\pi}) \end{cases} \qquad n, m \in \mathbb{Z}$$

Diese zwei Mengen von Unstetigkeitsstellen lassen sich zur der im Satz genannten zusammenfassen.

❑

Definition 4.6. Als *Auszucker* bezeichnen wir ein Intervall der Form

$$\left[\pi m - \arcsin\left(\frac{1}{\pi - \arcsin(\frac{1}{\pi})}\right), \pi m + \arcsin\left(\frac{1}{\pi - \arcsin(\frac{1}{\pi})}\right)\right],\ \text{für ein } m \in \mathbb{N}$$

Für festes $m \in \mathbb{N}$ spricht man vom m-ten Auszucker.

Bemerkung 4.7. Jede Unstetigkeits- und Nullstelle des Integranden befindet sich in einem Auszucker. Das heißt zwischen benachbarten Auszuckern gibt es keine Unstetigkeits- und keine Nullstelle.

Zwischen zwei Auszuckern lässt sich das Integral daher, ohne viele Probleme zu erwarten, numerisch berechnen. Abbildung 4 hat das bereits dargestellt. Das motiviert folgende Herangehensweise zur numerischen Berechnung des Integrales:

[4] streng genommen sind es keine Unstetigkeitsstellen sondern Stellen an denen die Funktion gar nicht definiert ist

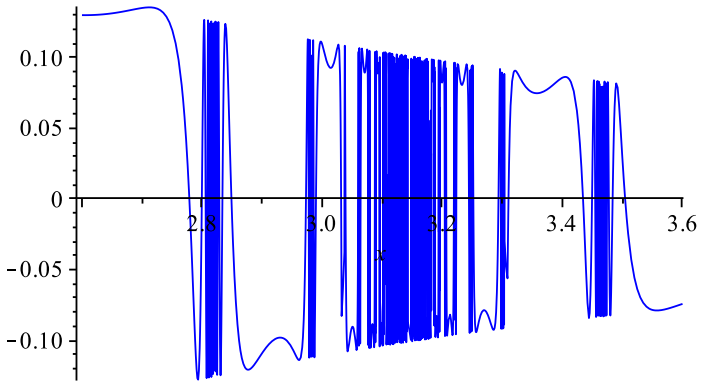


Abbildung 6: erster Auszucker

Zu nächst integriert man den *glatten Teil* der Funktion, also zwischen den Auszuckern. Das entspricht den Intervallen

$$\left[\pi m + \arcsin\left(\frac{1}{\pi - \arcsin(\frac{1}{\pi})}\right), \pi(m+1) - \arcsin\left(\frac{1}{\pi - \arcsin(\frac{1}{\pi})}\right)\right] \quad m \geq 1$$

$$\text{und} \quad \left[1, \pi - \arcsin\left(\frac{1}{\pi - \arcsin(\frac{1}{\pi})}\right)\right]$$

Und dann versucht man die etwas problematischeren Auszucker zu integrieren. Da sich die Funktion an den Auszucker nur um den $\frac{1}{x^2}$ Term unterscheiden reicht es eine Methode zur numerischen Integration zu finden. Diese lässt sich dann für alle Auszucker anwenden. Dabei bietet sich eine summierte Gaußquadratur an, wobei jede einzelne Gaußquadratur zwischen zwei aufeinanderfolgende Nullstellen durchgeführt wird.

Um die Notation ein wenig kompakter zu gestalten wird folgende Konventionen verwendet

$$f(x) := \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)$$

$$a_{m,n,k} := \pi m - \arcsin(\frac{1}{\pi n - \arcsin(\frac{1}{k\pi})}) \quad \text{(die Nullstellen)}$$

$a_{0,n,k}$ wird als 1 umdefiniert um die untere Integralgrenze zu beachten. Nun lässt sich das Integral in die eben genannten Teile aufspalten

$$\int_1^\infty f(x)\frac{1}{x^2}\,dx = \sum_{i=0}^{\infty}\int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x)\frac{1}{x^2}\,dx + \sum_{i=1}^{\infty}\int_{i.\text{Auszucker}} f(x)\frac{1}{x^2}\,dx$$

Definition 4.8. Ein *Auszuckerchen* ist ein Intervall der Form

$$\left[\pi m - \arcsin\left(\frac{1}{\pi n - \arcsin(\frac{1}{\pi})}\right), \pi m - \arcsin\left(\frac{1}{\pi n + \arcsin(\frac{1}{\pi})}\right)\right]$$

für $m \in \mathbb{N}$ und $n \in \mathbb{Z}\backslash\{0\}$. Für festes m und n spricht man vom m,n-ten Auszuckerchen.

Bemerkung 4.9. Für festes m häufen sich die Auszuckerchen von beiden Seiten zu πm. Außerdem sind für festes m alle m,n-ten Auszuckerchen im m-ten Auszucker enthalten. Das m,n-te Auszuckerchen enthält genau $\pi \cdot m - \arcsin(\frac{1}{n\pi})$ als einzige Unstetigkeitsstelle.

Lemma 4.10. Für eine monoton fallende Nullfolge $(a_n)_{n\in\mathbb{N}}$ gilt

$$0 \leq \sum_{n=1}^{\infty}(-1)^{n+1}a_n \leq a_1$$

Beweis Die Aussage wird zunächst für alle Partialsummen mittels vollständiger Induktion gezeigt. Daraus folgt unmittelbar die Aussage selbst.

Der Induktionsanfang ist mit $a_n \leq a_n$ erfüllt.

Induktionsannahme: Für alle natürlichen Zahlen kleiner als N gilt die Aussage bereits. Dabei treten für den Induktionsschritt auf $N+1$ zwei Fälle auf

1. Fall N gerade: Da die a_n monoton fallend sind ist die Differenz $a_{N+1} - a_N \leq 0$ und wendet man dann noch die Induktionsannahme an erhält man

$$\sum_{n=1}^{N}(-1)^{n+1}a_n + (-1)^{N+2}a_{N+1} = \sum_{n=1}^{N-1}(-1)^{n+1}a_n - a_N + a_{N+1} \leq a_1$$

2. Fall N ungerade: Für diesen Fall kann man unmittelbar die Induktionsannahme anwenden da $(-1)^{N+2}a_{N+1} \leq 0$

$$\sum_{n=1}^{N}(-1)^{n+2}a_n - a_{N+1} \leq a_1$$

Damit ist die Induktion vollständig. Für die Abschätzung von unten mit 0 verläuft der Beweis analog.

❑

Korollar 4.11. *Für eine monoton wachsende Nullfolge $(a_n)_{n\in\mathbb{N}}$ gilt*

$$a_1 \leq \sum_{n=1}^{\infty}(-1)^{n+1}a_n \leq 0$$

Beweis Multipliziert man die Ungleichung aus dem vorherigen Lemma mit -1 und betrachtet $(-a_n)_{n\in\mathbb{N}}$, so erhält man genau die Aussage des Korollars.

❑

Lemma 4.10 ist für die Fehlerabschätzung des Verfahrensfehlers enorm hilfreich. Nachdem $f(x+\pi) = -f(x)$ kann man für den *glatten Teil*

$$\sum_{i=0}^{\infty}\int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x)\frac{1}{x^2}\,dx = \sum_{i=0}^{\infty}(-1)^i\Big|\int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x)\frac{1}{x^2}\,dx\Big|$$

N	Ergebnis	Fehler
10^4	**4.988325**340391153e-01	$\leq 10^{-8}$
10^5	**4.9883253296**43207e-01	$\leq 10^{-10}$
10^6	**4.98832532953**5709e-01	$\leq 10^{-12}$
10^7	**4.98832532953463**5e-01	$\leq 10^{-14}$

Tabelle 3: glatter Teil

die Restreihe $\sum_{i=N}^{\infty}(-1)^i\left|\int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x)\frac{1}{x^2}\,dx\right|$ durch $\left|\int_{s_N}^{t_N} f(x)\frac{1}{x^2}\,dx\right| \leq \frac{1}{\pi N^2}$ abschätzen.

Jetzt fehlt noch die Integration über die Auszucker

$$\sum_{i=1}^{\infty}\int_{i.\text{Auszucker}} f(x)\frac{1}{x^2}\,dx = \sum_{i=1}^{\infty}(-1)^i\left|\int_{i.\text{Auszucker}} f(x)\frac{1}{x^2}\,dx\right|$$

Bemerkung 4.12. Nun kann man sich wieder Lemma 4.10 zur Fehlerabschätzung der Restreihe ab N bedienen. Unter Verwendung von Satz 4.1 kann man eine Abschätzung der Größenordnung $\frac{1}{N^3}$ erwarten, was jedoch noch eine ziemlich grobe Abschätzung ist.

Um später eine bessere Fehlerschranke zu bekommen wird der Wert des Integrals über den $N+1$. Auszucker numerisch berechnet. Das Integral über einen m. Auszucker wird aufgeteilt in das Integral über die Auszuckerchen in jenem Auszucker und den Teilen zwischen den Auszuckerchen. Zwischen den Auszuckerchen sind keine Null- und Unstetigkeitsstellen, daher lassen sich diese Teile noch ohne große Komplikation numerisch berechnen. So erhält man für die *glatten Teile des m-ten Auszuckers* folgende Darstellung, wobei die erste Reihe die Teile links von $m\pi$ sind und die zweite Reihe rechts davon.

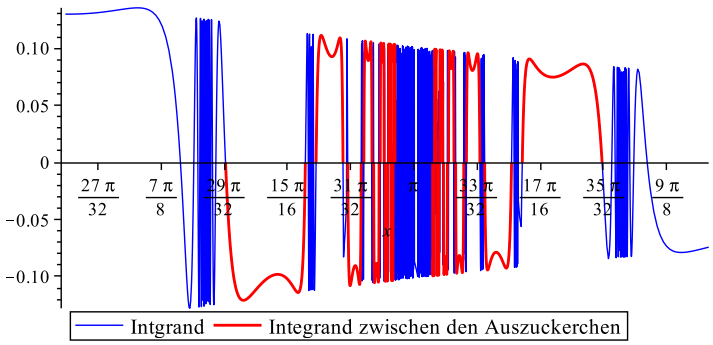


Abbildung 7: glatte Teile im Auszucker

$$\sum_{i=1}^{\infty} \int_{a_{m,i,-1}}^{a_{m,i+1,1}} \frac{f(x)}{x^2}\,dx + \sum_{i=1}^{\infty} \int_{a_{m,-i-1,-1}}^{a_{m,-i,1}} \frac{f(x)}{x^2}\,dx$$

$A_{m,i}$ beschreibt das (m,i)-ten Auszuckerchen

$$\sum_{i=1}^{\infty} \int_{A_{m,i}} \frac{f(x)}{x^2}\,dx + \sum_{i=1}^{\infty} \int_{A_{m,-i}} \frac{f(x)}{x^2}\,dx$$

Wobei die Restreihen ab N für den Auszuckerchen und die Teile dazwischen für große N einem Integral um $m\pi$ mit Radius von ungefähr $(\pi N)^{-1}$ entspricht. Denn für die Grenzen eines $A_{m,n}$ verhalten sich, wegen der Taylorapproximation für große n wie

$$\pi m - \arcsin\Big(\frac{1}{\pi n - \arcsin(\frac{1}{\pi})}\Big) \approx \pi m - \frac{1}{\pi n - \arcsin(\frac{1}{\pi})} \approx \pi m - \frac{1}{\pi n}$$

Das ergäbe dann aufsummiert über alle Auszucker einen Fehler von

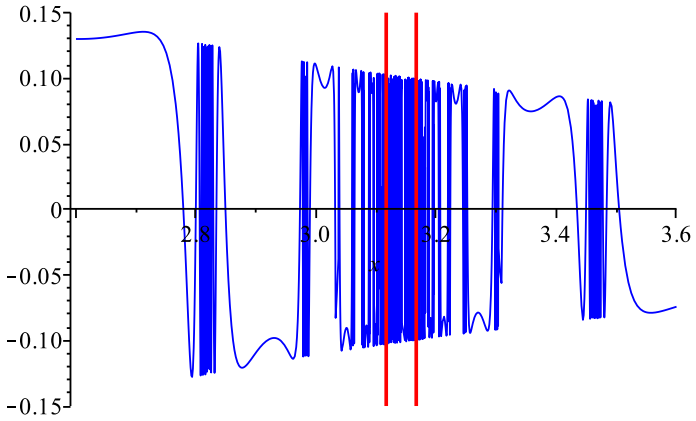


Abbildung 8: Bereich des Fehlers

$$\sum_{m=1}^{\infty} (-1)^m \Big| \int_{m\pi-\frac{1}{N\pi}}^{m\pi+\frac{1}{N\pi}} \frac{f(x)}{x^2}\,dx \Big|$$

Dieser lässt sich wieder mit Hilfe von Lemma 4.10 und Satz 4.1 betragsmäßig Abschätzen durch

$$\Big| \int_{\pi-\frac{1}{N\pi}}^{\pi+\frac{1}{N\pi}} \frac{f(x)}{x^2}\,dx \Big| \leq \frac{8}{\pi(\pi^2 N - 2)^2} \approx \frac{8}{\pi^5 N^2} \tag{15}$$

Daher kann man für jeden Auszucker die einzelnen Summen ohne weiteres *nur* bis N laufen lassen, ohne eine Aufsummierung eines Fehlers befürchten zu müssen.

Bemerkung 4.13. Will man ab dem m-ten Auszucker die *Tiefe* der Integration ändern (also die Reihe bei einem früheren N abbrechen lassen) kann man

pessimistisch einen Fehler von

$$\frac{8}{m^3\pi^5N^2} \tag{16}$$

dazurechnen.

Die Integration eines Auszuckerchen wird wieder aufgeteilt, sodass man nur zwischen Nullstellen integriert und das aufsummiert, erhält man

$$\int_{A_{m,n}} \frac{f(x)}{x^2}\,dx = \sum_{k=1}^{\infty} \int_{a_{m,n,k}}^{a_{m,n,k+1}} \frac{f(x)}{x^2}\,dx + \sum_{k=1}^{\infty} \int_{a_{m,n,-k-1}}^{a_{m,n,-k}} \frac{f(x)}{x^2}\,dx,$$

Innerhalb eines Auszuckerchen ist $g(x) := \left(\sin\left(\frac{1}{\sin(x)}\right)\right)^{-1}$ monoton, daher verhält sich $f(x) = \sin(g(x))$ ähnlich wie die Sinusfunktion je nach Steigung verzerrt. Um das einzusehen betrachtet man zu nächst die Ableitung von g

$$\frac{d}{dx}\left(\sin\left(\frac{1}{\sin(x)}\right)\right)^{-1} = \cot(x)\cot\left(\frac{1}{\sin(x)}\right)\frac{1}{\sin(x)}\frac{1}{\sin(\frac{1}{\sin(x)})}$$

Durch geeignetes Umformen, ähnlich wie im Beweis von Satz 4.3, sieht man, dass die Nullstellen der Ableitung bei

$$\pi m - \arcsin\left(\frac{1}{\pi n - \frac{\pi}{2}}\right)$$

sind. Zwischen den zwei Nullstellen ist genau eine Polstelle der Ableitung $\pi m - \arcsin(\frac{1}{\pi n})$. Da aber das Vorzeichen der Ableitung vor und nach der Polstelle gleicht bleibt und die Ableitung ansonsten zwischen den Nullstellen stetig ist ist das Vorzeichen zwischen den Nullstellen gleichbleibend, daher ist g auf dem Bereich monoton. Jedes Auszuckerchen ist innerhalb zweier dieser Nullstellen.

Zur numerischen Berechnung ist es sinnvoll die einzelnen Integrale alle auf das Einheitsintervall zu transformieren, da dann für eine Quadraturformel die Stützstellen und Gewichte nur einmal bestimmt werden müssen.

$$\int_{a_{m,n,k}}^{a_{m,n,k+1}} \sin(g(x))\,dx = d_{n,k}\int_0^1 \sin(g(d_{n,k}x + a_{m,n,k}))\,dx,$$

wobei $d_{n,k} := a_{m,n,k+1} - a_{m,n,k} = \arcsin(\frac{1}{\pi n - \arcsin(\frac{1}{k\pi})}) - \arcsin(\frac{1}{\pi n - \arcsin(\frac{1}{(k+1)\pi})})$. Für negative k erhält man ein analoges Resultat.

Bemerkung 4.14. $d_{n,k}$ ist für steigendes k monoton fallend, da $\arcsin(\frac{1}{(.)})$ konvex ist.

Lemma 4.15. Bezeichne g' die erste Ableitung von $\left(\sin\left(\frac{1}{\sin(x)}\right)\right)^{-1}$ und seien $a_{m,n,k}$, $b_{m,n,k}$ und $d_{n,k}$ so wie zuvor definiert, so gilt

$$\lim_{k\to\infty} g'(a_{m,n,k})d_{n,k} = \pi \quad \text{und} \quad \lim_{k\to\infty} g'(a_{m,n,k+1})d_{n,k} = \pi.$$

Wobei $g'(a_{m,n,k})d_{n,k}$ monoton von unten konvergiert und $g'(a_{m,n,k+1})d_{n,k}$ monoton von oben. Außerdem liegt eine Konvergenzordnung von $\frac{1}{k}$ vor.

Beweis Durch Einsetzen und Verwenden der Periodizitätseigenschaft der Trigonometrischenfunktion erhält man

$$g'(a_{m,n,k}) = (k\pi)^2(n\pi - \arcsin(\frac{1}{k\pi}))\sqrt{(n\pi - \arcsin(\frac{1}{k\pi}))^2 - 1}\sqrt{1 - \frac{1}{(k\pi)^2}}.$$

Wenn man $d_{n,k}$ in eine Taylorreihe bezüglich k bis zum quadratischen Term entwickelt, erhält man bis auf einen Fehler der Größe $O(\frac{1}{k^3})$

$$d_{n,k} \approx \frac{1}{n\pi\sqrt{(n\pi)^2-1}}\frac{1}{k(k+1)\pi} + \frac{2(n\pi)^2-1}{((n\pi)^2-1)(n\pi)^2\sqrt{(n\pi)^2-1}}\frac{2k-1}{k^2(k+1)^2\pi^2}$$

Das Produkt von $g'(a_{m,n,k})$ und dem zweiten Summanden der obigen Darstellung von $d_{n,k}$ geht mit $O(\frac{1}{k})$ gegen Null. Daher wird nur noch mit dem ersten Summanden ausmultipliziert.

$$\frac{\pi k}{k+1}\sqrt{1 - \frac{1}{(k\pi)^2}}\Big(1 - \frac{\arcsin(\frac{1}{k\pi})}{n\pi}\Big)\sqrt{\frac{(n\pi - \arcsin(\frac{1}{k\pi}))^2 - 1}{(n\pi)^2 - 1}}$$

Diesem Term kann man ablesen, dass der erste Faktor gegen π und alle anderen gegen 1 konvergieren. Für $a_{m,n,k+1}$ verläuft der Beweis analog

❑

Satz 4.16. *Für das Integral zwischen zwei Nullstellen in einem Auszuckerchen gibt es folgende Approximation*

$$\big(1 - O(\frac{1}{k})\big)\int_{a_{m,n,k}}^{a_{m,n,k+1}} f(x)\,dx \approx (-1)^{m+n+k}d_{n,k}\frac{2}{\pi}.$$

Beweis Da $g(x) := \frac{1}{\sin(\frac{1}{\sin(x)})}$ wie vorher schon erwähnt zwischen zwei Nullstellen in einem Auszuckerchen stetig und monoton ist, existiert auch die Inverse g^{-1}.

$$\int_{a_{m,n,k}}^{a_{m,n,k+1}} \sin(g(x))\,dx = \int_{(-1)^{m+n}k\pi}^{(-1)^{m+n}(k+1)\pi} \sin(x)\frac{1}{g'(g^{-1}(x))}\,dx$$

Wendet man den Mittelwertsatz der Integralrechnung an und substituiert mit πx, so erhält man für ein $\xi \in [a_{m,n,k}, a_{m,n,k+1}]$

$$\frac{\pi}{g'(\xi)}\int_{(-1)^{m+n}k}^{(-1)^{m+n}(k+1)} \sin(\pi x)\,dx \quad \text{bzw.} \quad (-1)^{m+n+k}\frac{d_{n,k}}{d_{n,k}}\frac{\pi}{g'(\xi)}\frac{2}{\pi}$$

Nachdem laut Lemma 4.15 $g'(\xi)d_{n,k}$ mit Ordnung $\frac{1}{k}$ gegen π konvergiert, erhält man die Aussage.

❑

Ein analoges Resultat bekommt man im Auszuckerchen rechts von der Polstelle integriert.

Bemerkung 4.17. Wenn man im Beweis den Ableitungsterm $\frac{1}{x^2}$ mitschleppt dann erhält man, dass es ein $\xi \in [a_{m,n,k}, a_{m,n,k+1}]$ gibt, sodass

$$\left(1 - O(\frac{1}{k})\right) \int_{a_{m,n,k}}^{a_{m,n,k+1}} \frac{f(x)}{x^2}\, dx \approx \frac{1}{\xi^2}(-1)^{m+n+k} d_{n,k} \frac{2}{\pi}.$$

ξ kann als Näherung mit dem Halbierungspunkt des Intervalls gleichgesetzt werden.

Für große k kann daher dieser Wert anstatt einer Quadratur Methode verwendet werden, da erstens die Approximation ziemlich gut ist und zweitens durch $d_{n,k}$ die Zahl ohnehin ziemlich klein ist. Nun lässt sich das Integral über den (m,n)-ten Auszucker als Summe zweier alternierenden Reihen darstellen.

$$(-1)^{m+n} \left(\sum_{k=1}^{\infty} (-1)^k \Big| \int_{a_{m,n,k}}^{a_{m,n,k+1}} \frac{f(x)}{x^2}\, dx \Big| + \sum_{k=1}^{\infty} (-1)^{k+1} \Big| \int_{a_{m,n,-k}}^{a_{m,n,-k-1}} \frac{f(x)}{x^2}\, dx \Big| \right).$$

Hört man wiederum bei K auf zu summieren passiert ein Fehler kleiner als

$$\Big| \int_{a_{m,n,K}}^{a_{m,n,K+1}} \frac{f(x)}{x^2}\, dx \Big| + \Big| \int_{a_{m,n,-K-1}}^{a_{m,n,-K}} \frac{f(x)}{x^2}\, dx \Big|$$

Allerdings kann man noch wesentlich mehr herausholen, da man für jedes Auszuckerchen und für jeden Auszucker diese Art von Fehler macht und sich daher einiges aufsummierte. Exemplarisch wird dies für den Fehler der ersten Reihe (links zur Unstetigkeitsstelle) und für Auszuckerchen links von $m\pi$ $(n > 0)$ verdeutlicht.

$$\sum_{m=1}^{\infty} (-1)^m \sum_{n=1}^{\infty} (-1)^n \sum_{k=K}^{\infty} (-1)^k \Big| \int_{a_{m,n,k}}^{a_{m,n,k+1}} \frac{f(x)}{x^2}\, dx \Big|$$

Als Reihe in m sind alle Voraussetzung von Lemma 4.10 erfüllt daher lässt sich die Reihe durch den Wert des ersten Summanden $(m = 1)$ Abschätzen. Analoges gilt für n, wodurch sich sogar der gesamte Fehler jetzt durch $\left| \int_{a_{1,1,K}}^{a_{1,1,K+1}} \frac{f(x)}{x^2}\, dx \right|$ abschätzen lässt. Beziehungsweise verhält sich der Fehler wie

$$d_{1,K} \frac{2}{\pi} \frac{4}{(a_{1,1,K} + a_{1,1,K+1})^2} \tag{17}$$

4.3 Auswerten

Nachdem nun das Integral bereits in lauter Reihen aufgeteilt wurde und für Approximationen über endliche Summen einige Fehlerabschätzungen gemacht wurden, kann nun die numerische Auswertung beginnen.

Zunächst wird das Integral in die erwähnten Reihen aufgeteilt und diese werden separat ausgewertet. Für den sogenannten *glatten Teil*

$$\sum_{i=0}^{\infty} \int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x) \frac{1}{x^2}\, dx = \sum_{i=0}^{10^7} \int_{a_{i,-1,-1}}^{a_{i+1,1,1}} f(x) \frac{1}{x^2} + R \approx 0.4988325329534635$$

wurde bereits die Tabelle 3 gezeigt, wobei $R \leq \pi^{-1} 10^{-14}$. Für die Auswertung wurde eine summierte Gaußquadratur mit 1000 Stützstellen pro Summand verwendet. Da der Integrand in diesen Bereichen analytisch ist, ist der Fehler der

Gaußquadratur im Bereich der Rechengenauigkeit und daher vernachlässigbar klein.

Die Reihe über die Auszucker wurde in die Reihe über die Auszuckerchen

$$\sum_{m=1}^{\infty} \sum_{n\in\mathbb{Z}\setminus\{0\}} \int_{A_{m,n}} \frac{f(x)}{x^2}\,dx$$

und die Reihe über die Teile dazwischen (siehe Abbildung 7).

$$\sum_{m=1}^{\infty} \left(\sum_{n=1}^{\infty} \int_{a_{m,n,-1}}^{a_{m,n+1,1}} \frac{f(x)}{x^2}\,dx + \sum_{n=1}^{\infty} \int_{a_{m,-n-1,-1}}^{a_{m,-n,1}} \frac{f(x)}{x^2}\,dx \right)$$

Wegen Abschätzung (15) kann man die inneren Reihen (mit dem Laufindex n) bei 10^5 abbrechen, um die zehnte Nachkommastelle zu garantieren. (Für die erste Reihe ist damit eine Einschränkung von $\mathbb{Z}\setminus\{0\}$ auf $[-10^5, 10^5]\cap\mathbb{Z}\setminus\{0\}$ gemeint) Nachdem das jedoch einen hohen Rechenaufwand mit sich bringt, kann man sich damit behelfen, dass der Fehler durch weniger nahes heranbringen an $m\pi$ bei größeren m wesentlich kleiner wird. Ab $m = 10$ muss man gemäß Bemerkung 4.13 nur bis 10^3 aufsummieren (die Summe mit Laufindex n), um den Fehler mit 10^{-10} zu beschränken. Für die numerischen Berechnung der Teile zwischen den Auszuckerchen werden die Integrale in der Summe auf das Einheitsintervall transformiert. Damit muss für die anschließende Gaußquadratur die Stützstellen und Gewichte nur einmal berechnet werden.

$$(a_{m,n+1,1} - a_{m,n,-1}) \int_0^1 \frac{f((a_{m,n+1,1} - a_{m,n,-1})x + a_{m,n,-1})}{(a_{m,n+1,1} - a_{m,n,-1})x + a_{m,n,-1})^2}\,dx$$

Analoges gilt für die Integrale von rechts. Wertet man nun diese Integrale aus und summiert sie auf erhält man die Werte aus Tabelle 4. Hier sieht man auch, dass zumindestens einmal für die Teile zwischen den Auszuckerchen die Abschätzung der Restreihe über m ab 1001 wesentlich besser als $\frac{1}{1001^3}$ ist, wie Bemerkung 4.12 schon angedeutet hat. Die Restreihe lässt sich eben gemäß Lemma 4.10 durch den ersten Summanden abschätzen. Gemäß Bemerkung 4.13 wurden die Fehler in der Tabelle angegeben.

m läuft	n läuft	Wert	Fehler
1 bis 1	1 bis 10^6	-2.725432509867055e-03	$\leq 3\cdot 10^{-14}$
2 bis 10	1 bis 10^5	2.670641294848690e-04	$\leq 4\cdot 10^{-13}$
11 bis 100	1 bis 10^4	-1.145800633072389e-06	$\leq 3\cdot 10^{-13}$
101 bis 1000	1 bis 10^3	-1.324083840144348e-09	$\leq 3\cdot 10^{-14}$
		$\sum$ = -2.459515505099098e-03	$\sum \leq 7\cdot 10^{-13}$
1001 bis 1001	1 bis 10^5	-2.683155066043680e-12	$\leq 3\cdot 10^{-21}$

Tabelle 4: Integral zwischen den Auszuckerchen

Für die Auszuckerchen wird das ganze ein wenig unangenehmer da man mit der Strategie von Nullstelle zu Nullstelle zu integrieren dann bereits bei

drei ineinander geschachtelten Summen ist. Jedoch ist mit der Abschätzung (17) schon einiges gerettet. Auch die Approximation durch Bemerkung 4.17 für späte Teile eines Auszuckerchens erspart Aufwand. Für den Fehler der bei dem Abbruch der innersten Reihe passiert

$$\sum_{m=1}^{\infty}(-1)^m \sum_{n=1}^{\infty}(-1)^n \sum_{k=K}^{\infty}(-1)^k \Big| \int_{a_{m,n,k}}^{a_{m,n,k+1}} \frac{f(x)}{x^2}\, dx \Big|$$

gibt es die bereits erwähnte Abschätzung (17), da jedoch für wachsende n unterschiedliche kleinere K verwendet wurden addiert man pessimistisch die Abschätzung (17) bei jeder Änderung dazu. Durch Auswerten erhält man Tabelle 5.

m läuft	n läuft	Wert	Fehler1	Fehler2
1 bis 1	1 bis 10^5	-1.75902592519e-04	$\leq 3 \cdot 10^{-12}$	$\leq 6 \cdot 10^{-13}$
2 bis 10	1 bis 10^4	1.69289752129e-05	$\leq 4 \cdot 10^{-11}$	-
11 bis 100	1 bis 10^3	-7.21384457370e-08	$\leq 3 \cdot 10^{-11}$	$\leq 2 \cdot 10^{-11}$
101 bis 1000	1 bis 10^3	-8.33445584430e-11	$\leq 3 \cdot 10^{-14}$	$\leq 2 \cdot 10^{-11}$
		$\sum$ = -1.59045838636e-04	$\sum \leq 7 \cdot 10^{-11}$	$\sum \leq 4 \cdot 10^{-11}$
1001 bis 1001	1 bis 10^5	-1.68873623658e-13	$\leq 3 \cdot 10^{-21}$	

Tabelle 5: Integral über Auszuckerchen

Fehler1 in Tabelle 5 steht für den Abbruch in der Reihe über n und Fehler2 für den Abbruch in der Reihe über k. Der zweite Fehler in der zweiten Zeile ist schon beim zweiten Fehler in der ersten Zeile inkludiert, da an den selben Stellen die gleiche Änderungen der Anzahl der Summanden gemacht wurde.

Nimmt man jetzt alle Zahlen der aufgepalteten Teile des Integrals und addiert diese und ihre Fehlerabschätzungen zusammen so erhält man endlich den Wert

$$I = 0.4962139716097281$$

wobei aufgrund der Fehlerabschätzungen die 10. Nachkommastelle noch exakt ist.

4.4 Monte-Carlo Methode

Einen anderen Zugang bildet die Monte-Carlo-Integration. Die grundlegende Idee ist es, an zufällig gewählten Punkten die Funktion auszuwerten und dessen Mittelwert zu berechnen. Aus wahrscheinlichkeitstheoretischer Sicht sieht dieses Verfahren wie folgt aus:[5]

Sei $f : [a, b] \to \mathbb{R}$ eine integrierbare Funktion und bezeichne U_i unabhängige, stetig gleich verteilte Zufallsvariablen auf dem Intervall $[a, b]$. Dann besitzt

[5] Vgl. hierzu Kusolitsch, Maß- und Wahrscheinlichkeitstheorie - Eine Einführung

$X_i := f(U_i)$ den Erwartungswert $\frac{1}{b-a}\int_a^b f(x)dx$. Für das um die Intervalllänge bereinigte Stichprobenmittel $Y_n := \frac{b-a}{n}\sum_{i=1}^n X_i$ folgt dann $\mathbb{E}(Y_n) = \int_a^b f(x)dx$ und $\mathbb{V}(Y_n) = \frac{\sigma_X^2(b-a)^2}{n}$, wobei σ_X^2 die Varianz der X_i bezeichnet. Dann folgt wegen der Ungleichung von Tschebyscheff:

$$\mathbb{P}\left(\|Y_n - \int_a^b f(x)dx\| \geq \epsilon\right) \leq \frac{\sigma_X^2(b-a)^2}{n\epsilon^2} \tag{18}$$

Da wir die Varianz der X_i nicht kennen, können wir sie entweder nach oben abschätzen oder diese erneut mittels Monte-Carlo simulieren[6]. Bei ersterer Methode erhalten wir dadurch Konfidenzintervalle, deren tatsächliche Überdeckungswahrscheinlichkeit größer als die nominelle ist. Da für die Funktion $\|f\|_\infty = 1$ gilt, schätzen wir die Varianz durch 1 ab. Dies mag im ersten Moment recht grob erscheinen, allerdings wird sich später zeigen, dass $E(X_i)^2 < 0.01$ und daher $\mathbb{V}(X_i) \approx \mathbb{E}(X_i^2)$. Des Weiteren nimmt die quadrierte Funktion tatsächlich auf großen Teilen des Intervalls einen Wert über 0.7 an. Eine Monte-Carlo Simulation liefert für σ_X einen Schätzwert 0.29915, was gar nicht so viel besser als $\sigma_X^2 \leq 1$ ist.

Bei einer Simulation mit 10^{11} Zufallszahlen, ergab das Monte-Carlo Integral $\int_0^{0.4} \sin(\frac{1}{\sin(\frac{1}{\sin(\frac{1}{x})})})\,dx$ den Wert $I := -0.04639392$. Das Integral über das verbleibende Intervall $[0.4, 1]$ lässt sich numerisch relativ gut berechnen und ergibt einen gerundeten Wert von 0.5426090464. Unter der Annahme, dass dieser Wert zumindest auf sieben Nachkommastellen genau ist, erhalten wir als Schätzwert für das gesamte Integral 0.4962151. Dieser stimmt immerhin auf 5 Nachkommastellen mit der von uns berechneten Zahl überein. Als Konfidenzintervalle zu den Überdeckungswahrscheinlichkeiten α (mit einer abgeschätzten Varianz $\sigma_X^2 \leq 0.3$ erhalten wir:

ÜW $1-\alpha$	linker Rand	rechter Rand
0.1	**0.4962**144	**0.4962**159
0.5	**0.4962**141	**0.4962**161
0.9	**0.4962**129	**0.4962**173
0.95	**0.4962**120	**0.4962**182
0.99	**0.4962**082	**0.4962**221
0.999	**0.496**1932	**0.4962**370

Tabelle 6: Konfidenzintervalle der Monte-Carlo Simulation

Man beachte hierbei, dass in den ersten beiden Konfidenzintervallen der Wert 0.4962139716 nicht enthalten ist. Wir können aber mit der Ungleichung (18) die Wahrscheinlichkeit abschätzen, mit der wir - unter der Annahme, dass letztere Zahl tatsächlich der gesuchte Wert ist - ein stärker abweichendes Ergebnis in der Monte-Carlo Simulation erhalten. Diese in der Statistik als $p - Wert$

[6]Man beachte, dass man in diesem Fall die Varianz von X_i^2 schätzen müsste - hier wäre dann eine Abschätzung durchaus sinnvoll.

bezeichnete Kenngröße liefert die Zahl 0.377, was im Allgemeinen als nicht signifikant bezeichnet wird. Vereinfacht gesagt, sind die Unterschiede in der sechsten Nachkommastelle zufälliger Natur und können nicht in Richtung einer falschen Berechnung interpretiert werden. Dies wäre ohnedies mehr als verwunderlich gewesen, weil wir analytisch die Exaktheit von 0.4962139716 auf mehr als acht Stellen beweisen können.

4.5 Irrelevante Ausführungen

Nun betrachten wir folgende allgemeinere Funktion:

$$I(n) := \int_0^1 \frac{1}{f_n(x)} dx$$

wobei

$$f_n(x) := \csc \circ \csc \circ \cdots \circ \csc(\frac{1}{x}) \tag{19}$$

die n-fache Hintereinanderführung der Kosekans-Funktion ist. Somit stimmt der oben behandelte Fall mit $I(3)$ überein.

Wir werden allerdings hier nicht für jedes $I(n)$ eine eigene mathematische Theorie herleiten sondern uns der Monte-Carlo Methode bedienen, wobei wir die Anzahl der Zufallszahlen auf 10^9 reduzieren werden. Für verschieden Werte von n simulieren wir nun das Parameterintegral. Aus der Abschätzung (18) erhalten wir für die Wahrscheinlichkeit von 0.05 ein $\epsilon = \sqrt{2} \cdot 10^{-4}$. Wir könnten nun aus dem Schätzer, sowie $\pm\epsilon$ ein 95% Konfidenzintervall kreieren.

$I(n)$	Schätzer
$I(1)$	$\sin(1) - C_i(1) \approx 0.5040670619$
$I(2)$	0.4827738
$I(3)$	0.4962151
$I(4)$	0.4890532
$I(5)$	0.4962648
$I(6)$	0.4908792
$I(10)$	0.4915751
$I(20)$	0.4915746
$I(30)$	0.4915114

Tabelle 7: Werte für das Parameterintegral

Interessanterweise liegen alle Werte relativ nahe beieinander, wobei $I(5)$ sogar auf 4 Nachkommastellen mit $I(3)$, also dem ursprünglich betrachteten Integral, übereinstimmt. Auch der Wert von $I(10)$ stimmt sehr genau mit jenem von $I(20)$ überein. Auch $I(30)$ liegt (unter Berücksichtigung des Umstands, dass

$\epsilon = \sqrt{2} \cdot 10^{-4}$) sehr nahe bei diesen Werten. Wir erhoffen uns dennoch keine Konvergenz von $\lim_{n\to\infty} I(n)$, sondern vermuten, dass für den Computer f_{10} und f_{20} numerisch ident sind.

5 Differenzialgleichung

Zweiundvierzig

Literatur

[1] N. Kusolitsch. *Maß- und Wahrscheinlichkeitstheorie: Eine Einführung.* Springer, 2011.

[2] Franklin T Luk and Sanzheng Qiao. A fast eigenvalue algorithm for hankel matrices. *Linear Algebra and Its Applications*, 316(1):171–182, 2000.

[3] J. Michael Steele. *The Cauchy-Schwarz master class.* MAA Problem Books Series. Mathematical Association of America, Washington, DC; Cambridge University Press, Cambridge, 2004. An introduction to the art of mathematical inequalities.

Aufteilung

1. Primzahlreihe - Nico Amann
2. Matrixnorm
   (a) Vorbereitung - Nathanael Skrepek
   (b) Auswertung - Nico Amann
3. Stochastische Folge - Nico Amann
4. Oszillierendes Integral
   (a) Vorbereitung - Nathanael Skrepek
   (b) Auswertung - Nathanael Skrepek
   (c) Monte Carlo - Nico Amann
   (d) Irrelevante Ausführungen - Nico Amann
5. Differenzialgleichung - Chuck Norris

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Seminararbeit

Digit Challenge

Ausgeführt am Institut für

Analysis und Scientific Computing

unter Anleitung von

Prof. Dr. Ansgar Jüngel

durch

Maximilian Bernkopf, Axel Böhm, Lisa Mayer

1 Aufgabe 1

Work smart, not hard.

Jugendspruch

Ausarbeitung: Maximilian Bernkopf

Zu bestimmen ist der Wert

$$s := \sum_{p\in\mathbb{P}} \frac{p\,(p-1)}{p^{\pi}} = \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-2}} - \sum_{p\in\mathbb{P}} \frac{1}{p^{\pi-1}} = P\,(\pi-2) - P\,(\pi-1)\,,$$

wobei P die Primzetafunktion bezeichnet. Damit ist eigentlich schon alles getan, denn diese ist für beliebige Genauigkeit zum Beispiel bereits in Mathematica implementiert. Es seien an dieser Stelle nur 1000 Nachkommastellen angegeben, da wir ja kein Papier verschwenden wollen. Der interessierte Leser kann sich, sofern Mathematica ihm nicht zugänglich ist, die kostenlose Testversion herunterladen und sich mehr Stellen ausgeben lassen. Ist M die gewünschte Anzahl der Nachkommastellen so gelingt dies beispielsweise mit dem Befehl:

```
N[PrimeZetaP[Pi-2] - PrimeZetaP[Pi-1], M]
```

Hier nun also die versprochenen Nachkommastellen:

$s = 1,41702525894502464667989471995404264872028939967775267916650262120972230$
$367292792288230740045164225687961089332754335640671640263722796672446549688786$
$446744955022457395947107941232376246223753770875706867626655421097143693469807$
$110028710637776469294869407619042659007487238906113189939126917375106483508654$
$023534190189038230584410435542051417928616714093443427902522033996595319183787$
$196189618890335046976233294967019365733952012567827307936971814680853706198911$
$819811566086732240615668920901137565701669695682622165105432915565024356215708$
$914099846684251169811993381509094766209834861209475549004975214467196028411657$
$408572305934612942139058050070538453438780877370605843954006492486133018761081$
$780704451940960570964557940854548649122910825853130041490527182625736011993328$
$330168799622852610410751374434742375158654457046184641065137897706662339442650$
$564036989724620308347743686641236954833122637268093018073880092390744375618372$
$477096312361554508302649663226489899423715352938435290181190411576345 1\ldots$

2 Beispiel 2

Ausarbeitung: Axel Böhm

2.1 Einleitung

Wir betrachten die *unendliche* Matrix $A = (a_{ij})_{i,j=1}^{\infty}$ definiert durch:

$$a_{ij} = \begin{cases} \frac{1}{i+j-1} & \text{, falls } i+j \text{ prim} \\ \frac{1}{(i+j-1)^2} & \text{, sonst} \end{cases} \tag{1}$$

Diese Matrix bildet einen Operator von $(\ell^2, \|.\|_2)$, auf sich selbst. Gesucht ist die Operatornorm, welche definiert ist durch:

$$\|A\| = \sup_{\|x\|_2=1} \|Ax\|_2 \tag{2}$$

Da eine analytische Berechnung eher aussichtlos erscheint und die Implementierung einer unendlich großen Matrix am Computer nicht möglich ist wählen wir zunächst den wahrscheinlich intuitivsten Zugang: Die Approximation unseres Operators über $n \times n$ Matrizen mit möglich großem n, welche wir von nun an als A_n bezeichnen werden. Diese Matrizen kann man als Operatoren auf ℓ^2 interpretieren, indem man sie sich mit Null fortgesetzt denkt, oder als lineare Abbildungen von $\mathbb{R}^n$ auf sich selbst.

Bemerkung 2.2. *$\mathbb{R}^n$ ist im folgenden immer ausgestattet mit der üblichen euklidischen Norm, welche ebenfalls mit $\|.\|_2$ bezeichnet wird. Dies sollte jedoch zu keinerlei Verwirrung führen.*

Des weiteren werden wir nicht immer genau unterscheiden auf welchem der Räume wir die Matrizen betrachten, da die Abbildungsnormen in beiden Fällen die gleichen sind.
Da die Abbildungsnorm einer linearen Abbildung von $(\mathbb{R}^n, \|.\|_2)$ auf sich selbst gleich dem größten Singularwert der Matrix ist die diese Abbildung repräsentiert, haben wir schon unser erste Aufgabe gefunden: Diesen Singulärwert zu bestimmen. Zunächst erinnern wir uns an folgendes:

Definition 2.3. *Sei $M \in \mathbb{R}^{m \times n}$ eine Matrix, dann sind die Singulärwerte von M die Wurzeln aus den positiven Eigenwerten von $M^T M$.*

Da $\forall n \in \mathbb{N}$ gilt, dass die von uns zu untersuchende Matrix A_n quadratisch und symmetrisch ist, fällt der größte Singulärwert mit dem Betragsgrößten Eigenwert zusammen.
Für die Freunde der Funktionalanalysis sei hier noch erwähnt, dass man die gleiche Aussage auch aus dem folgenden Satz erhält.

Theorem 2.4. *Sei $(H, (.,.)_H)$ ein Hilbertraum und $T \in \mathcal{B}(H)$ ein normaler Operator, dann gilt, dass die Operatornorm von T gleich seinem Spektralradius ist.*

2.5 Berechnung des Eigenwertes mittels Poweriteration

Nun müssen wir uns dem Problem widmen, wie der Eigenwert am besten zu berechnen ist. Ein erster Versuch diesen mit Matlabs `norm`() Funktion zu berechnen scheitert sehr schnell an der stark anwachsenden Rechenzeit. Ein zweiter Anlauf unter Verwendung der Funktion `normest`() erlaubt zwar größere Dimensionen, doch auch hier wird schnell die größe der Matrizen problematisch. In weiterer Folge wollen wir uns die besondere Gestalt der Matrizen A_n zu nutze machen. Es fällt nämlich auf dass der Eintrag in Zeile i und Spalte j immer nur von $i+j$ abhängt. Matrizen dieser Art werden zu ehren von Hermann Hankel auch Hankelmatrizen genannt.

Definition 2.6. *Eine Matrix $B = (b_{ij})_{i,j=0}^{n-1}$ wird Hankelmatrix genannt, genau dann wenn $b_{ij} = f(i+j)$, für eine Funktion $f : \mathbb{N}_0 \to \mathbb{R}$.*

Eine Hankelmatrix der Dimension $m \times m$ hat also nur $2m-1$ unterschiedliche Einträge, was die Möglichkeit einer effizienteren Speicherung nahelegt. Nun muss man sich allerdings wieder selber Gedanken über die Brechnung der Norm machen und das Verfahren unserer Wahl wird die Poweriteration sein. Dieser Algorithmus basiert im wesentlichen auf einer wiederholten Multiplikation der Matrix auf einen Vektor. Dieser Vektor dreht sich in Richtung des Eigenraumes des Betragsgrößten Eigenwertes und durch Auswertung der Länge des Vektors vor und nach der Matrixmutiplikation lässt sich ebendieser Eigenwert ermitteln.
Aufgrund der effizienten Speicherung der Matrix in einem $2n-1$ langen Vektor muss man die Matrix-Vektor-Multiplikation nun selbst implementieren. Hierbei kommt man selbst bei der Verwendung von Matlabs Vektorarithmetik (welche sehr schnell ist da die zugruneliegenden Routinen in C programmiert sind) nicht um die Verwendung einer for-Schleife nicht herum. Daher wurde an dieser Stelle der Versuch gestartet die eben beschriebene Prozedur in einer maschinennäheren Programmiersprache umzusetzen. Hierbei wurde C++ gewählt, was sich aufgrund der fehlenden Vorkenntnisse des Autors als mühsames Unterfangen herausstellen und nur in einer unwesentlich besseren Rechenzeiten resultieren sollte. Daraufhin wurde wieder ausschließlich in Matlab programmiert.
Da für die Dauer der Poweriteration, die Anzahl der Matrix-Vektor-Multiplikation entscheidend ist, möchte man natürlich mit einem Vektor starten der schon möglichst in die gleiche Richtung zeigt wie der Eigenvektor der zum betragsgrößten Eigenwert gehört. Hierfür wird besagter Eigenvektor zwischen Berechnungen immer abgespeichert und bei Erhöhung der Dimension einfach mit Nullen aufgefüllt. Man könnte versuchen sich hier eine geschicktere Auffüllstrategie zu überlegen, dass man dem neuen Eigenvektor schon möglichst nahe kommt. Da jedoch schon das simple Anhängen von Nullen zu nur sehr wenigen Iterationen führt, kann diese Idee getrost verworfen werden.

2.7 Fast Fourier Transform

Da momentan die meiste Rechenzeit für die reine Matrix-Vektor-Multiplikation verbraucht wird werden wir versuchen diese weiter zu optimieren. Hierfür verwenden wir den in [3] propagierten Algorithmus, der auf FFT basiert. Wir wollen hier dieses Verfahren kurz erläutern.

Sei H die von uns betrachtete Hankelmatrix und bezeichne $h = (h_0, h_1, \cdots, h_{2n-2})^T$ den Vektor der alle Matrixeinträge beinhaltet, wobei folgender Zusammenhang gilt:

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_1 & h_2 & h_3 & \cdots & h_n \\ h_2 & h_3 & h_4 & \cdots & h_{n+1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ h_{n-1} & h_n & h_{n+1} & \cdots & h_{2n-2} \end{pmatrix} \tag{3}$$

Bezeichne w den Vektor auf den wir die Matrix anwenden wollen und p das Ergebnis dieser Operation. Also:

$$p = Hw \tag{4}$$

Dann seien $\hat{c}, \hat{w} \in \mathbb{R}^{2n-1}$ folgendermaßen definiert:

$$\hat{c} = \begin{pmatrix} h_n & h_{n+1} & h_{n+2} & \cdots & h_{2n-1} & h_1 & h_2 & \cdots & h_{n-1} \end{pmatrix}^T \tag{5}$$

$$\hat{w} = \begin{pmatrix} w_n & w_{n-1} & w_{n-2} & \cdots & w_1 & 0 & \cdots & 0 \end{pmatrix}^T \tag{6}$$

und

$$y = \texttt{ifft}(\texttt{fft}(\hat{c}) .* \texttt{fft}(\hat{w})) \tag{7}$$

wobei $.*$ die punktweise Multiplikation darstellt.
Der gesuchte Vektor p lässt sich nun schreiben als:

$$p = \begin{pmatrix} y_1 & y_2 & \cdots & y_{n-1} & y_n \end{pmatrix}^T \tag{8}$$

Speziell die Permutierungen in diesem Algorithmus wirken zunächst ein wenig merkwürdig. Diese rühren jedoch von der Transformation auf eine Töplitzmatrix her.
Nichtsdestoweniger beschleunigt dieses Verfahren die Berechnungen ungemein und lässt bei gleicher Rechenzeit plötzlich über 100 mal so große Dimensionen n zu. Des weiteren beziffert [3] den Aufwand einer Multiplikation (inklusive Hin- und Rücktransformation) mit $30n\log(n) + O(n)$ flops, welcher sich in der nur *moderat* ansteigenden Rechenzeit wiederspiegelt (siehe Tabelle 2).

2.8 Hankeloperatoren

In diesem Abschnitt wollen wir nun ein wenig die Theorie beleuchten. Zum Beispiel stellt sich die Frage ob der zu untersuchende Operator tatsächlich nach ℓ^2 abbildet und ob er in diesem Fall überhaupt beschränkt ist. Hierfür benötigen wir zuerst folgende Begriffsbildung:

Definition 2.9. *Sei H eine unendliche Hankelmatrix, dann bezeichnen wir den von ihr definierten Operator auf ℓ^2 mit Γ_h und nennen diesen Hankeloperator, wobei $h = (h_n)_{n\in\mathbb{N}_0}$ die Folge der Koeffizienten ist.*

Bemerkung 2.10. *Von nun an bezeichnet $\mathbb{T}$ die Einheitskreislinie in $\mathbb{C}$, i.e. $\{z \in \mathbb{C} : |z| = 1\}$.*

Aus [6] wissen wir, dass Γ_h tatsächlich nach ℓ^2 abbildet, falls $(h_n)_{n\in\mathbb{N}_0} \in \ell^2$. Erfreulicherweise wird dies von A beziehungsweise der dazugehörigen Folge $(a_n)_{n\in\mathbb{N}_0}$ offensichtlich erfüllt.
Zur weiteren Untersuchung benötigen wir folgenden Satz:

Theorem 2.11. *Eine unendliche Hankelmatrix H betrachtet als Operator von ℓ^2 auf sich selbst ist genau dann beschränkt wenn es eine Funktion $\psi \in L^\infty(\mathbb{T})$ gibt sodass $h_m = \hat{\psi}(m)\,\forall m \in \mathbb{N}_0$. In diesem Fall gilt sogar*

$$\|H\| = \inf\{\|\psi\|_\infty : \psi \in L^\infty(\mathbb{T}),\, \hat{\psi}(m) = h_m\,\forall m \in \mathbb{N}_0\} \tag{9}$$

Für einen Beweis siehe [6].

□

Da nicht klar ist wie eine solche Funktion gefunden werden kann betrachten wir zunächst ein Beispiel:

Beispiel 2.12. *Die unendliche Hankelmatrix I die von der Folge $(\alpha_n)_{n\in\mathbb{N}}$, wobei $\alpha_n = \frac{1}{n+1}\,\forall n \in \mathbb{N}_0$, erzeugt wird nennt sich Hilbertmatrix.*

Die Koeffizientenfolge wird zum Beispiel von der Funktion ϕ erzeugt.

$$\phi(z) = \sum_{n=0}^{\infty} \frac{1}{n+1} z^n \tag{10}$$

welche jedoch schon auf $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$ unbeschränkt ist. Nichtsdestoweniger ist Γ_α ein beschränkter Operator. Betrachten wir hierzu die Funktion ψ definiert auf $\mathbb{T}$:

$$\psi(e^{it}) = ie^{-it}(\pi - t) \quad t \in [0, 2\pi) \tag{11}$$

Integration zeigt, dass tatsächlich folgendendes gilt:

$$\hat{\psi}(n) := \frac{1}{2\pi}\int_0^{2\pi} \psi(e^{it})e^{-it}\,\mathrm{d}t = \frac{1}{1+n} \quad \forall n \in \mathbb{N}_0 \tag{12}$$

Offensichtlich gilt, dass $\|\psi\|_\infty = \pi$ woraus nach Theorem 2.11 folgt, dass $\|\Gamma_\alpha\| \leq \pi$. Man kann hier sogar Gleichheit zeigen, siehe [4].
Wir haben zwar noch keine passende Funktion für unsere Matrix A gefunden jedoch sehen wir, da A von der Folge

$$a_n = \begin{cases} \frac{1}{n+1} & \text{, falls } n+2 \text{ prim} \\ \frac{1}{(n+1)^2} & \text{, sonst} \end{cases} \tag{13}$$

erzeugt wird, dass $a_n \leq \alpha_n \quad \forall n \in \mathbb{N}_0$.

Theorem 2.13. *Mit den Bezeichnungen dieses Kapitels gilt: $\|\Gamma_a\| \leq \|\Gamma_\alpha\|$.*
Beweis.

Da sowohl I also auch A nur positive Einträge haben, kann man sich bei der Spremumsbildung in der Operatornorm auf die Menge $M^+ = \{x \in \ell^2 : \|x\|_2 = 1\,,\, 0 \leq x_j \,\forall j \in \mathbb{N}\}$ beschränken. Aus der oben angesprochenen Ungleichung $a_n \leq \alpha_n$ folgt natürlich, dass

$$\|\Gamma_a x\|_2 \leq \|\Gamma_\alpha x\|_2 \quad x \in M^+ \tag{14}$$

daher

$$\sup_{x\in M^+} \|\Gamma_a x\|_2 \leq \sup_{x\in M^+} \|\Gamma_\alpha x\|_2 \tag{15}$$

und daraus die gewünschte Ungleichung.

□

Korollar 2.14. $\|\Gamma_a\| \leq \pi$

Da die empirischen Berechnungen suggerieren, dass die wahre Operatornorm eher in der Nähe von 1.4 liegt, ist die momentane Abschätzung im Hinblick auf Nachkommastellen die mit Sicherheit richtig sind nicht hilfreich. Aus diesem Grund begeben wir uns nun doch auf die Suche nach einer Funktion aus $L^\infty(\mathbb{T})$ deren Fourierkoeffizienten die Koeffizientenfolge der Matrix erzeugt. Der Einfachkeit halber teilen wir die Folge $(a_n)_{n\in\mathbb{N}_0}$ zunächst auf in die Summe aus $(b_n)_{n\in\mathbb{N}_0}$ und $(c_n)_{n\in\mathbb{N}_0}$ wobei

$$b_n := \frac{1}{(n+1)^2} \quad \forall n \in \mathbb{N}_0 \tag{16}$$

und

$$c_n = \begin{cases} \frac{1}{n+1} - \frac{1}{(n+1)^2} & \text{, falls } n+2 \text{ prim} \\ 0 & \text{, sonst} \end{cases} \tag{17}$$

Aus der Dreieckungleichung folgt dann, dass $\|\Gamma_a\| \leq \|\Gamma_b\| + \|\Gamma_c\|$. Eine beschränkte Funktion zu finden die $(b_n)_{n\in\mathbb{N}_0}$ erzeugt ist leicht, da die Reihe darüber konvergent ist:

$$\varphi : z \mapsto \sum_{n=0}^{\infty} \frac{1}{(n+1)^2} z^n \tag{18}$$

Da $\|\varphi\|_\infty = \frac{\pi^2}{6}$ folgt aus Theorem 2.11 dass $\|\Gamma_b\|_\infty \leq \frac{\pi^2}{6}$. Da die Reihe über die Reziprokwerte aller Primzahlen jedoch divergiert, ist die Frage nach einer beschränkten Funktion auf der Einheitskreislinie deren Fourierkoeffizienten gleich $(c_n)_{n\in\mathbb{N}_0}$ sind nicht so einfach zu lösen. Außerdem suggerieren die Berechnungen, dass $\|\Gamma_a\|$ in der Nähe von 1.4 liegt, womit $\frac{\pi^2}{6} \approx 1.6$ alleine schon nicht die beste Abschätzung ist.
Tatsächlich ist es dem Autor nicht gelungen eine solche Funktion zu finden.

2.15 Konvergenz

Nachdem wir gezeigt haben, dass A tatsächlich einen beschränkten Operator definiert, können wir uns darüber Gedanken machen in welchem Sinn A_n gegen A konvergiert. Dass die Konvergenz Punktweise, also in der starken Operatortopologie stattfindet ist einfach

zu sehen. Jedoch benötigen wir etwas mehr um unser Verfahren der Approximation über die endlichen Matrizen zu rechtfertigen. Konkret hätten wir gerne, dass $\lim_{n\to\infty}\|A_n\| = \|A\|$. Denn dann wüssten wir, dass die Folge der berechneten Eigenwerte tatsächlich gegen den wahren Wert konvergiert. Hierfür formulieren wir folgendes Lemma.

Lemma 2.16. *Sei $(X, \|.\|_X)$ ein Banachraum und $(B_n)_{n\in\mathbb{N}}$ eine Folge von Operatoren aus $\mathcal{B}(X)$ die in der starken Operatortopologie gegen ein B konvergiert, sodass es eine Menge $M \subset \{x \in X : \|x\|_X = 1\}$ gibt für die gilt, dass*

$$\sup_{\|x\|_X=1} \|B_n x\|_X = \sup_{x\in M} \|B_n x\|_X \quad \forall n \in \mathbb{N} \tag{19}$$

und

$$\sup_{\|x\|_X=1} \|Bx\|_X = \sup_{x\in M} \|Bx\|_X \tag{20}$$

und

$$\|B_n x\|_X \leq \|B_m x\|_X \quad \forall x \in M \quad \forall m \geq n \tag{21}$$

Dann gilt das Folgende:

$$\lim_{n\to\infty} \|B_n\| = \|B\| \tag{22}$$

Beweis.

Zuerst bemerken wir, dass aus den Vorraussetzungen folgt:

$$\|B_n\| \leq \|B_m\| \quad \forall m \geq n \tag{23}$$

Nun zum eigentlichen Beweis:

$$\begin{aligned}
&\lim_{n\to\infty} \|B_n\| \\
&\overset{i.}{=} \lim_{n\to\infty} \sup_{\|x\|_X=1} \|B_n x\|_X \\
&\overset{ii.}{=} \lim_{n\to\infty} \sup_{x\in M} \|B_n x\|_X \\
&\overset{iii.}{=} \sup_{n\in\mathbb{N}} \sup_{x\in M} \|B_n x\|_X \\
&\overset{iv.}{=} \sup_{x\in M} \sup_{n\in\mathbb{N}} \|B_n x\|_X \\
&\overset{v.}{=} \sup_{x\in M} \lim_{n\to\infty} \|B_n x\|_X \\
&\overset{vi.}{=} \sup_{x\in M} \|Bx\|_X \\
&\overset{vii.}{=} \sup_{\|x\|_X=1} \|Bx\|_X \\
&\overset{viii.}{=} \|B\|
\end{aligned}$$

Ad (i.): Definition der Abbildungnorm
Ad (ii.): Voraussetzung
Ad (iii.): Aufgrund der in (23) gezeigten Monotonie können wir sup statt lim schreiben
Ad (iv.): Vertauschung von Suprema, da beide endlich
Ad (v.): Voraussetzung über die punktweise Monotonie
Ad (vi.): Konvergenz in der Operatortopologie
Ad (vii.): Voraussetzung
Ad (viii.): Definition der Abbildungnorm

□

Unsere Folge A_n erfüllt offensichtlich die Vorraussetzungen von Lemma 2.16 mit $M = M^+ := \{x \in \ell^2 : \|x\|_2 = 1\,,\, 0 \leq x_j \,\forall j \in \mathbb{N}\}$. Somit haben wir gezeigt, dass die Approximationen tatsächlich gegen den gesuchten Wert konvergieren.

2.17 Ergebnisse

Dimension	Wert	Lösungsverfahren
3000	1.3956879908882631	norm
4000	1.3957097729681185	norm
30×10^3	1.3957637881213329	normest
50×10^3	1.3957671901628703	zeilenweise Speicherung + PI
60×10^3	1.3957683009842248	zeilenweise Speicherung + PI
70×10^3	1.3957688295896986	zeilenweise Speicherung + PI
90×10^3	1.3957695583942458	zeilenweise Speicherung + PI
100×10^3	1.3957698116044472	zeilenweise Speicherung + PI
100×10^3	1.3957613403979254	C++
120×10^3	1.3957701827925473	C++
130×10^3	1.3957636924714846	C++

Tabelle 1: naive Berechnungen in Matlab und C++

Aufgrund der jeweils nur einmaligen Berechnung sind die Zeitangaben in Tabelle 2 erheblichen Schwankungen unterworfen. Der Trend sollte sich jedoch ohne große Probleme dennoch ablesen lassen.

Dimension	Wert	Dauer der FFT	Dauer der PI
2×10^5	1.3957709216798986		
3×10^5	1.3957712820777992		
4×10^5	1.3957714616563668		
1.1×10^6	1.3957717942058072		
1.5×10^6	1.3957718436194240		
2×10^6	1.3957718771227372		
3×10^6	1.3957719101956252		
4.5×10^6	1.3957719319442097		
7×10^6	1.3957719472428607		
10×10^6	1.3957719553819170		
11×10^6	1.3957719570992424	5.4009	43.62
12×10^6	1.3957719585244253	19.1436	148.70
13×10^6	1.3957719597261562	6.48405	53.50
14×10^6	1.3957719607522778	10.4954	79.47
15×10^6	1.3957719616400077	25.71070	157.057
16×10^6	1.3957719624134530	7.278	66.83
17×10^6	1.3957719630936742	7.528	54.52
20×10^6	1.3957719647269542	23.8277	184.89
23×10^6	1.3957719659532670	13.471	113.96
26×10^6	1.3957719668969126	19.690	245.71
29×10^6	1.3957719676166498	15.637	187.97
32×10^6	1.3957719681759937	12.791	180.62
37×10^6	1.3957719688639150	22.941	209.06
40×10^6	1.3957719689949872	28.11	134.95
45×10^6	1.3957719695923136	87.88	293.83
48×10^6	1.3957719696638482	23.09	98.026
55×10^6	1.3957719701632312	55.75	807.04
65×10^6	1.3957719705184446	53.19	608.04
75×10^6	1.3957719705855287	86.57	255.90
100×10^6	1.3957719711806107	75.27	289.04
120×10^6	1.3957719711863887	232.01	716.79

Tabelle 2: Berechnungen mittels Power-Iteration und FFT in Matlab

3 Aufgabe 3

Jedermann, der Zufallszahlen mit einer arithmentischen Methode erzeugen will, ist nicht ganz bei Trost.
John von Neumann

Ausarbeitung: Maximilian Bernkopf

Dieser Abschnitt behandelt die folgende Problemstellung laut Angabe: Es sei die stochastische Folge $(x_n)_{n\in\mathbb{N}}$ definiert durch $x_1 = x_2 = x_3 = 1$ und

$$x_{n+3} = x_{n+2} \pm \frac{1}{3}(x_{n+1} + x_n) \qquad n \geq 1,$$

wobei das Vorzeichen jeweils mit Wahrscheinlichkeit $\frac{1}{2}$ positiv oder negativ ist. Zu bestimmen ist der Grenzwert

$$\sigma := \lim_{n\to\infty} \sqrt[n]{|x_n|}.$$

Vorab wollen wir die Fragestellung etwas präzisieren: Sei $(\Omega, \mathcal{F}, \mathbb{P})$ ein Wahrscheinlichkeitsraum. Weiters sei $(Y_n)_{n\in\mathbb{N}}$ eine Folge unabhängiger und identisch verteilter[1] Zufallsvariablen mit $\mathbb{P}(Y_1 = 1) = \mathbb{P}(Y_1 = -1) = \frac{1}{2}$. Die stochastische Folge $(x_n)_{n\in\mathbb{N}}$ ist nun definiert durch $x_1 = x_2 = x_2 = 1$ und

$$x_{n+3} = x_{n+2} + \frac{1}{3}Y_n(x_{n+1} + x_n) \qquad n \geq 1.$$

zu bestimmen ist der Grenzwert - in welchem Sinne dieser zu verstehen ist wird im nächsten Abschnitt behandelt -

$$\sigma := \lim_{n\to\infty} \sqrt[n]{|x_n|}.$$

3.1 Theoretische Vorüberlegungen

Für die nun folgenden Ausführungen benötigen wir einiges an Theorie über stochastische Matrizen. Die nötigen Definitionen werden in aller Kürze und sämtliche Sätze ohne Beweis gebracht. Der interessierte Leser sei auf [1] verwiesen.

Definition 3.2. Sei G eine topologische Halbgruppe. Ein topologischer Raum B *agiert* auf der Halbgruppe G, falls man stetig jedes Element $(g, x) \in G \times B$ mit einem Element $g \cdot x \in B$ identifizieren kann und zwar derart, dass

$$(g_1 g_2) \cdot x = g_1 \cdot (g_2 \cdot x) \qquad \forall g_1, g_2 \in G, x \in B.$$

Falls G darüberhinaus eine topologische Gruppe mit Einheit e ist und weiters $e \cdot x = x$ für jedes $x \in B$ gilt, so nennt man B einen *G-Raum.*

[1] von nun an kurz i.i.d.

Definition 3.3. Sei G eine auf B agierende topologische Halbgruppe. Eine stetige Abbildung $\sigma : G \times B \to \mathbb{R}$ heißt *additiver Kozyklus*, falls

$$\sigma(g_1 g_2, x) = \sigma(g_1, g_2 \cdot x) + \sigma(g_2, x)$$

für alle $g_1, g_2 \in G, x \in B$.

Beispiele 3.4. Wählt man $G = Gl(n, \mathbb{R})$ so ist S^{n-1} die Einheitssphäre im $\mathbb{R}^n$ ein G-Raum, wenn man

$$M \cdot x = \frac{Mx}{\|Mx\|}$$

für $M \in Gl(n, \mathbb{R}), x \in S^{n-1}$ wählt. Weiters bildet $\sigma(M, x) = \ln(\|Mx\|)$ einen additiven Kozyklus.

In der gesamten Ausarbeitung zu Aufgabe 3 gelte von nun an

$$A := \begin{pmatrix} 1 & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad B := \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Bemerkung 3.5. Man beachte, dass $\det(A) = \frac{1}{3}$ sowie $\det(B) = -\frac{1}{3}$, womit sich mit Multiplikation einer Konstante beide Matrizen zugleich auf Matrizen mit betragsmäßiger Determinante gleich 1 transformieren lassen. Dieser Zusammenhang wird später noch eine untere Abschätzung liefern.

Wir wollen nun das Problem der Aufgabenstellung umschreiben. Es sei $(\Omega, \mathcal{F}, \mathbb{P})$ ein Wahrscheinlichkeitsraum und $(M_n)_{n\in\mathbb{N}}$ eine Folge unabhängiger und identisch verteilter Zufallsvariablen mit Werten in $\mathbb{R}^{3\times 3}$ mit $\mathbb{P}(M_1 = A) = \mathbb{P}(M_1 = B) = \frac{1}{2}$. Diese Überlegungen ermöglichen es die stochastische Folge in der Form

$$\begin{pmatrix} x_{n+3} \\ x_{n+2} \\ x_{n+1} \end{pmatrix} = M_n M_{n-1} \cdots M_1 \begin{pmatrix} x_3 \\ x_2 \\ x_1 \end{pmatrix}$$

zu schreiben. Man beachte weiters, dass die Verteilung von M_1 durch das Maß

$$\mu = \frac{1}{2}\delta_A + \frac{1}{2}\delta_B$$

gegeben ist, wobei δ_x das Dirac-Maß an x bezeichne.
Bekannte Resultate aus der Theorie der stochastischen Matrizen geben uns die Existenz des fast sicheren Grenzwertes

$$\gamma = \lim_{n\to\infty} \frac{1}{n} \ln(\|M_n \cdots M_1\|)$$

Darüber hinaus gilt auch

$$\gamma = \lim_{n\to\infty} \frac{1}{n} \mathbb{E}[\ln(\|M_n \cdots M_1\|)], \tag{24}$$

sowie

$$\exp(\gamma) = \lim_{n\to\infty} \sqrt[n]{|x_n|}.$$

Gleichung (24) liefert uns also eine deterministische Möglichkeit den gesuchten Grenzwert zu approximieren. Gilt darüberhinaus $|\det(M_n)| = 1$ so gilt auch $\gamma > 0$. Mit Bemerkung 3.5 erhalten wir also für $\tilde{M}_n := \sqrt[3]{3}M_n$ zusammen mit

$$\tilde{\gamma} := \lim_{n\to\infty} \frac{\ln\left(\left\|\tilde{M}_n \cdots \tilde{M}_1\right\|\right)}{n} = \ln\left(\sqrt[3]{3}\right) + \lim_{n\to\infty} \frac{\ln\left(\|M_n \cdots M_1\|\right)}{n} = \ln\left(\sqrt[3]{3}\right) + \gamma,$$

die untere Schranke $\exp\left(-\ln\left(\sqrt[3]{3}\right)\right) = \frac{1}{\sqrt[3]{3}} \approx 0.69336127435 < \sigma$.

Bemerkung 3.6. Man beachte, dass der Grenzwert γ offensichtlich nicht von der gewählten Matrixnorm abhängt, da bekanntlich auf endlichdimensionalen Räumen ohnehin alle Normen äquivalent sind.

Definition 3.7. Sei G eine auf B agierende topologische Halbgruppe, μ ein Wahrscheinlichkeitsmaß auf G und ν ein Wahrscheinlichkeitsmaß auf B. Dann bezeichne $\mu\dot{*}\nu$ eine Verteilung auf B, welche

$$\int_B f(x)\, d\mu\dot{*}\nu(x) = \int_B \int_G f(g\cdot x)\, d\mu(g)\, d\nu(x)$$

für jedes borelmessbare $f : B \to \mathbb{R}$ erfüllt. Weiters nennt man ν ein μ-*invariantes* Maß, falls $\mu\dot{*}\nu = \nu$.

Zwei Vektoren $x, y \in \mathbb{R}^d \setminus \{0\}$ haben dieselbe *Richtung* falls es ein $\lambda \in \mathbb{R}$ gibt, sodass $x = \lambda y$. Dies definiert eine Äquivalenzrelation Γ auf dem $\mathbb{R}^d \setminus \{0\}$.

Definition 3.8. Die Menge aller Richtungen im $\mathbb{R}^d$ ist durch den Faktorraum von $\mathbb{R}^d \setminus \{0\}/\Gamma$ gegeben. Diesen bezeichnet man als *projektiven Raum*. Für ein $x \in \mathbb{R}^d \setminus \{0\}$ bezeichne $\overline{x}$ seine Richtung. Weiters sei $M \cdot \overline{x} := \overline{Mx}$ für ein $M \in Gl(d, \mathbb{R})$.

Bemerkung 3.9. Die Richtungen $\overline{x}$ im $\mathbb{R}^3$ lassen sich in natürlicher Weise mit der oberen Halbkugel mit Radius 1 identifizieren.

Ein weiteres wichtiges Resultat der stochastischen Matrizen liefert nun die Existienz eines μ invarianten Wahrscheinlichkeitsmaßes ν auf dem projektiven Raum von $\mathbb{R}^3$, sodass

$$\gamma = \int\int \ln\left(\frac{\|Mx\|}{\|x\|}\right) d\mu(M)\, d\nu(\overline{x})$$

Diese Überlegung führt auf die Invarianzbedingung

$$\nu(S) = \frac{1}{2}\nu\left(\overline{A^{-1}S}\right) + \frac{1}{2}\nu\left(\overline{B^{-1}S}\right)$$

für alle Borelmengen S. Parametrisiert man nun die obere Halbkugel durch das Einheitsrechteck $[0,1]^2$ so führt dies auf eine entsprechende Gleichung für Borelmengen von $[0,1]^2$.
Ein zentrales Ergebnis, welches für den gesuchten Grenzwert sowohl obere als auch untere Schranken liefern wird, ist das folgende

Lemma 3.10. *Der fast sichere Grenzwert*

$$\gamma = \lim_{n\to\infty} \frac{1}{n} \ln\left(\|M_n \cdots M_1\|\right)$$

erfüllt für beliebiges $m \in \mathbb{N}$ und jeder submultiplikativen[2] *Matrixnorm $\|\cdot\|$ die Abschätzungen*

$$\frac{1}{m}\mathbb{E}\left[\ln\left(\sigma_3\left(M_m \cdots M_1\right)\right)\right] \leq \gamma \leq \frac{1}{m}\mathbb{E}\left[\ln\left(\|M_m \cdots M_1\|\right)\right].$$

Beweis. Für die erste Ungleichung beachte man die Tatsache $\|PQ\| \geq \sigma_3(P)\,\sigma_3(Q)$. Hiermit ergibt sich

$$\begin{aligned}
\gamma &= \lim_{n\to\infty} \frac{1}{n} \ln\left(\|M_n \cdots M_1\|\right) \\
&= \lim_{n\to\infty} \frac{1}{nm} \ln\left(\|M_{nm} \cdots M_1\|\right) \\
&\geq \lim_{n\to\infty} \frac{1}{nm} \ln\left(\sigma_3\left(M_{nm} \cdots M_{(n-1)m+1}\right) \cdots \sigma_3\left(M_m \cdots M_1\right)\right) \\
&= \frac{1}{m} \lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{n} \ln\left(\sigma_3\left(M_{im} \cdots M_{(i-1)m+1}\right)\right) \\
&= \frac{1}{m}\mathbb{E}\left[\ln\left(\sigma_3\left(M_m \cdots M_1\right)\right)\right].
\end{aligned}$$

Man beachte hierbei, dass die Zufallsvariablen $\left(\sigma_3\left(M_{nm} \cdots M_{(n-1)m+1}\right)\right)_{n\in\mathbb{N}}$ unabhängig und identisch verteilt und sämtliche Integrierbarkeitsbedingungen zur Anwendung des Gesetzes der großen Zahlen erfüllt sind.
Für die zweite Ungleichung benutzt man in analoger Weise die Submultiplikativität, somit folgt

$$\begin{aligned}
\gamma &= \lim_{n\to\infty} \frac{1}{n} \ln\left(\|M_n \cdots M_1\|\right) \\
&= \lim_{n\to\infty} \frac{1}{nm} \ln\left(\|M_{nm} \cdots M_1\|\right) \\
&\leq \lim_{n\to\infty} \frac{1}{nm} \ln\left(\left\|M_{nm} \cdots M_{(n-1)m+1}\right\| \cdots \|M_m \cdots M_1\|\right) \\
&= \frac{1}{m} \lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{n} \ln\left(\left\|M_{im} \cdots M_{(i-1)m+1}\right\|\right) \\
&= \frac{1}{m}\mathbb{E}\left[\ln\left(\|M_m \cdots M_1\|\right)\right].
\end{aligned}$$

Womit das Lemma bewiesen ist. □

Die Implementierung der obigen Abschätzungen ist insofern einfach, da der Erwartungswert nur das arithmetische Mittel der Normen beziehungsweise kleinster Singulärwerte

[2] Die Forderung der Submultiplikativität ist für den Beweis dieses Lemmas essenziell, man beachte dennoch die Gültigkeit von Bemerkung 3.6

aller möglichen m-langen Matrixmultiplikationen bestehend aus den Matrizen A und B ist. Für $m = 3$ sind also zum Beispiel nur die Matrizen $AAA, AAB, ABA, ABB, BAA, BAB, BBA, BBB$ zu berechnen. Für allgemeines m hat man jedoch 2^m Matrizen zu berechnen, womit der numerische Aufwand exponentiell ist.

3.11 Viele Ideen - Viel Frustration

Der nun folgende Abschnitt sei all jenen Ideen gewidmet, welche aus diversen Gründen keine Resultate erbrachten. Sie seien dennoch an dieser Stelle erwähnt, da sie sehr wohl interessante Ergebnisse lieferten als auch für einen Wissenzuwachs sorgten.

Geschlossene Form Sämtliche Versuche das Matrixprodukt $M_n \cdots M_1$ in geschlossener Form darzustellen sind kläglich gescheitert.

Eine vollständig multiplikative Matrixnorm? Bei der Berechnung der Matrixnormen zur Bestimmung des gesuchten Grenzwertes stellte sich die Frage nach einer vollständig multiplikativen Matrixnorm, also einer Norm $\|\cdot\|$ auf dem $\mathbb{R}^{n\times n}$ mit der Eigenschaft $\|CD\| = \|C\|\,\|D\|$ für alle $C, D \in \mathbb{R}^{n\times n}$.
Diese Frage ist natürlich mit einem klaren Nein zu beantworten, wie das einfach Beispiel

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

zeigt, denn $CD = 0$ aber $C, D \neq 0$. Für unsere Betrachtung ist aber nicht der ganze $\mathbb{R}^{n\times n}$ relevant sondern lediglich die von den Matrizen A und B erzeugte Halbgruppe, also die Menge $G := \{\prod_{i=1}^{n} Y_i : Y_i = A \vee Y_i = B, n \in \mathbb{N}\}$. Die Frage nach einer derartigen Norm auf dieser Halbgruppe blieb längere Zeit unbeantwortet bis schließlich die zündende Idee kam. Bevor wir endlich die Antwort geben sei an den folgenden Satz erinnert.

Satz 3.12. *Für jede Matrixnorm $\|\cdot\|$ gilt*

$$\rho(A) = \lim_{k\to\infty} \left\|A^k\right\|^{\frac{1}{k}},$$

wobei ρ den Spekralradius bezeichnet.

Der Satz 3.12 leifert nun also die Antwort auf unsere Frage, denn gäbe es eine vollständig multiplikative Matrixnorm so müsste diese schon auf G mit dem Spektralradius übereinstimmen. Daher existiert die gewünschte Norm dann und nur dann wenn $\rho(CD) = \rho(C)\,\rho(D)$ für alle $C, D \in G$. In unserem Fall gilt jedoch leider $\rho(AB) = \rho(A)\,\rho(B)$.
Ein simples Beispiel, worauf die obige Überlegung anwendbar wäre ist die Halbgruppe der doppelt stochastischen Matrizen.

Schärfere Abschätzungen Eine Verschärfung des Lemmas 3.10 wäre wünschenswert gewesen. Aber von welcher Norm man auch ausgeht, so sind die besten unteren Abschätzungen für die Norm eines Produktes dennoch die Singulärwerte. Eine schärfere untere Schranke wurde versucht zu erreichen, indem die folgenden bekannten Ungleichungen verwendet wurden.

Lemma 3.13. *Für $C, D \in \mathbb{R}^{n\times n}$ gelten für jedes $t = 1 \dots n$ die Abschätzung*

$$\min_{i+j=t+1} \sigma_i(C)\,\sigma_j(D) \geq \sigma_t(CD) \geq \max_{i+j=t+n} \sigma_i(C)\,\sigma_j(D)\,, \tag{25}$$

sowie

$$\sum_{t=1}^{n} \sigma_t(C)\,\sigma_t(D) \geq \sum_{t=1}^{n} \sigma_t(CD) \geq \sum_{t=1}^{n} \sigma_t(C)\,\sigma_{n-t+1}(D)\,, \tag{26}$$

Man beachte, dass in der Mitte von 26 genau die Schattennorm steht. Dennoch führt ein analoges Vorgehen wie im Beweis von Lemma 3.10 nur auf dieselbe untere Schranken.

3.14 Numerische Ergebnisse

Im folgenden werden wir sämtliche numerische Herangehensweisen näher betrachen, ihre Vor- und Nachteile besprechen und dabei weitgehend chronologisch vorgehen. Die offensichtlich größte Schwierigkeit an unserem Problem ist die Stochastik. Möchte man also verlässliche Ergebnisse erziehlen, so wird man einer deterministischen Methode nicht ausweichen können.

Monte Carlo Experimente Eine der natürlichsten Herangehensweisen ist klarerweise die Monte Carlo Simulation.
Ein wichtiger Grund der gegen eine Monte Carlo Simulation spricht ist die Tatsache, dass *jede* zufällig erzeugte Abfolge von Vorzeichen zu einem Ereignis fortgesetzt werden kann welches nicht in jener Teilmenge von Ω liegt, auf welcher die fast sichere Konvergenz stattfindet. Diese Ergebnisse sind daher mit Vorsicht zu genießen. Dennoch ergab die Monte Carlo Simulation für eine große Anzahl an Versuchen und Iterationsschritten einen ersten Richtwert. Dieser betrug $\sigma \approx 1.009\dots$.

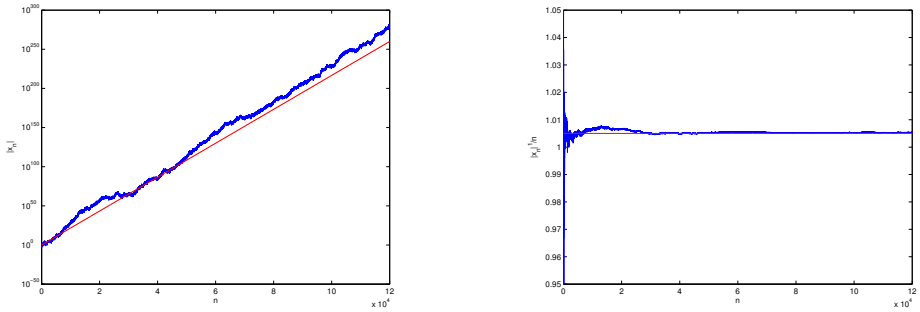


Abbildung 1: Die linke Graphik zeigt einen Semilogplot von $|x_n|$ gegen n. Die rechte Graphik zeigt einen Plot von $\sqrt[n]{|x_n|}$ gegen n. Die rote Gerade entspricht dem Richtwert der Monte Carlo Simulation.

Man sieht klar das zu vermutende exponentielle Verhalten der stochastischen Folge. Nichtsdestoweniger ist die Monte Carlo Simulation ein sehr starkes Hilfsmittel und hat auch bei der Ausarbeitung dieses Beispiels eine Richtung aufgezeigt.

Schranken mit Hilfe von Lemma 3.10 Im folgenden präsentieren wir die berechneten Schranken für den gesuchten Grenzwert. Dazu wurden die Erwartungswerte der Normen beziehungsweise der kleinsten Singulärwerte berechnet und anschließend darauf die Exponentialfunktion angewendet. Man beachte jedoch, dass für jedes m schließlich 2^m Matrixprodukte jeweils von der Länge m berechnet wurden, was wiederum in einem großen Rechenaufwand endete.
Es sei im Folgeneden

$$O_m := \exp\left(\frac{1}{m}\mathbb{E}\left[\ln\left(\|M_m \cdots M_1\|\right)\right]\right),$$

sowie

$$L_m := \exp\left(\frac{1}{m}\mathbb{E}\left[\ln\left(\sigma_3\left(M_m \cdots M_1\right)\right)\right]\right)$$

für $m \in \mathbb{N}$.

m	O_m	L_m
1	1.473 446 150 061 121	0.226 227 021 136 474
2	1.356 046 510 529 092	0.385 853 109 479 126
3	1.258 293 753 721 496	0.401 919 445 608 827
4	1.194 753 976 090 192	0.423 265 352 893 921
5	1.155 117 474 771 068	0.438 517 068 453 596
6	1.137 434 900 967 937	0.447 368 226 686 288
7	1.121 584 743 597 686	0.453 151 773 824 405
8	1.108 811 552 152 987	0.459 090 678 307 338
9	1.098 185 945 030 631	0.463 964 064 969 593
10	1.089 965 068 839 594	0.467 407 634 662 162
11	1.083 155 435 279 399	0.470 693 832 798 313
12	1.076 903 454 894 401	0.473 551 686 489 396
13	1.071 683 127 576 348	0.475 896 986 701 561
14	1.067 371 275 195 560	0.478 015 694 318 395
15	1.063 648 343 223 164	0.479 885 939 663 842
16	1.060 292 382 600 037	0.481 552 538 812 567
17	1.057 304 472 600 572	0.483 033 748 264 990
18	1.054 648 833 998 510	0.484 378 275 184 787
19	1.052 275 349 125 813	0.485 583 306 607 088
20	1.050 124 682 754 533	0.486 689 068 338 873
21	1.048 168 872 375 301	0.487 712 674 347 518
22	1.046 385 873 610 220	0.488 756 489 086 709
23	1.044 760 791 953 723	0.490 123 253 662 041
24	1.043 269 461 946 565	0.490 254 719 380 127
25	1.041 895 521 450 953	0.490 982 209 143 130
26	1.040 626 647 832 980	0.491 658 237 187 779

Mit diesen Daten und der bereits erhaltenen unteren Schranke können wir das folgende Lemma formulieren.

Lemma 3.15. *Der fast sichere Grenzwert σ liegt im Intervall* $(0.693361, 1.040626)$.

Man beachte, dass die Berechnung der kleinsten Singulärwerte also keine neuen Erkenntnisse gebracht hat.

Mit Kanonen auf Spatzen - Die Suche nach einem invarianten Maß Wir verfolgen in diesem Abschnitt die bereits erwähnte Idee eines invarianten Wahrscheinlichkeitsmaßes ν, welches die Bedingung

$$\nu(S) = \frac{1}{2}\nu\left(\overline{A^{-1}S}\right) + \frac{1}{2}\nu\left(\overline{B^{-1}S}\right)$$

für alle Borelmengen der oberen Einheitkugel S_+^2 erfüllt. Da mit dieser Bedingung alle Versuche, dieses explizit zu bestimmen gescheitert sind, wurde die Invarianzbedingung mittels der Kugeltransformation umgeschrieben. Dazu sei $\phi : (0,1)^2 \to S_+^2$ die homöomorphe Einbettung in die obere Einheitskugel[3] sowie $\pi : \mathbb{R}^3 \to S_+^2$ die Projektion auf die obere Einheitskugel. Damit erhalten wir für das entsprechend transformierte Maß, welches wieder mit ν bezeichnet sei, die Bedingung

$$\nu(S) = \frac{1}{2}\nu\left(\phi^{-1}\left(\pi\left(A^{-1}\phi(S)\right)\right)\right) + \frac{1}{2}\nu\left(\phi^{-1}\left(\pi\left(B^{-1}\phi(S)\right)\right)\right) \tag{27}$$

für alle Borelmengen $S \subset (0,1)^2$. Der Übersicht halber sei von nun an $f := \phi^{-1} \circ \pi \circ A^{-1} \circ \phi$ sowie $g := \phi^{-1} \circ \pi \circ B^{-1} \circ \phi$. Man beachte, dass unsere Invarianzbedingung 2 dimensional ist, da es sich um eine stochastische 3-Term Rekursion handelt. Wäre diese nur eine 2-Term Rekursion so wäre dies eine eindimensionale Bedingung, welche weitaus einfach zu behandeln ist, aber bereits hier entsteht ein sehr abstrakte Maße ν, der interessierte Leser sei auf [7] verwiesen.
Wir wollen nun das Maß ν diskretisieren, dazu seien $(I_{i,j})_{i,j=1}^n$ die Teilrechtecke von $(0,1)^2$ der Form $I_{i,j} = \left(\frac{i-1}{n}, \frac{i}{n}\right) \times \left(\frac{j-1}{n}, \frac{j}{n}\right)$. Mit $\nu_{i,j} = \nu(I_{i,j})$ erhalten wir die diskrete Bedingung

$$\nu_{i,j} = \frac{1}{2}\sum_{k,l=1}^{n} c_{k,l}\nu_{k,l},$$

wobei $c_{k,l} \in [0,2]$, die Überlappung der Menge $I_{k,l}$ mit den Mengen der rechten Seite in (27) angeben. Dies führe also auf ein Gleichungssystem in n^2 Variablen vom Rang $n^2 - 1$. Indem man die letzte Bedinugn durch $\sum_{i,j=1}^n \nu_{i,j} = 1$ ersetzt ist dieses eindeutig lösbar. Offensichtlich ist die Bestimmung der $c_{k,l}$ ein essentieller Punkt in dieser Berechnung.

Approximation des Bildes durch Rechtecke Im Folgenden illustrieren wir die Approximation des Bildes unter den Abbildungen f und g. Da sämtliche Versuche das Bild eines Rechtecks $[a,b] \times [c,d] \subset [0,1] \times [0,1]$ explitzit anzugeben gescheitert sind, musste bereits dieses numerisch approximiert werden.

[3]Man beachte, dass man sich auf $(0,1)^2$ beschränken kann und nicht $[0,1]^2$ benutzen muss, da sie sich nur um eine ν Nullmenge unterscheiden

Die Grundidee war es einen Zufallspunkt $x \in [a,b] \times [c,d]$ zu wählen, den Funktionswert dieses Punktes zu berechnen und schließlich jenes Teilrechteck, in welchem der Funktionswert liegt zu vierteln. Dieses Verfahren wird M mal iteriert. Zur graphischen Veranschaulichung betrachte man die untenstehende Abbildung.

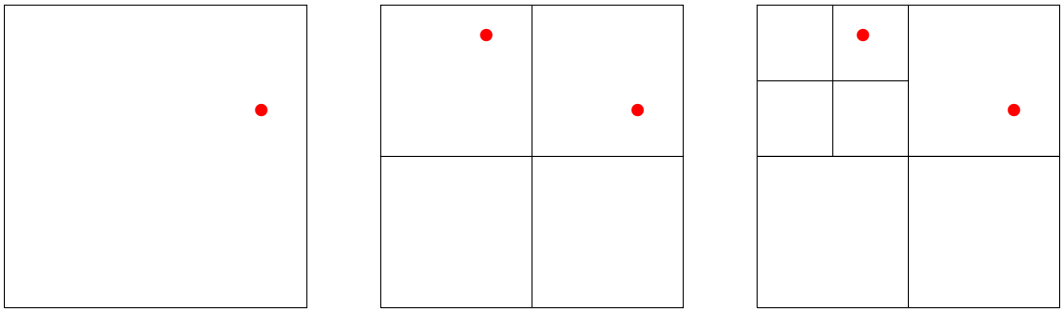


Abbildung 2: Veranschaulichung der ersten 2 Iterationen auf dem Rechteck $[0,1]^2$

Abschließend werden all jene Teilrechtecke entfernt, deren Durchmesser kleiner ist als eine vorgegebene Toleranzgrenze ε. Dies ermöglicht nicht nur eine Approximation sonderen auch eine einfache Berechnung der approximerten $c_{k,l}$. Um nun schließlich das Integral

$$\gamma = \int_{(0,1)^2} \frac{1}{2} \ln\left(\|A\phi\left(x\right)\|\right) + \frac{1}{2} \ln\left(\|B\phi\left(x\right)\|\right) d\nu\left(x\right)$$

approximativ zu bestimmen, bedarf es einer näheren Untersuchung des Integranden. Dazu sei

$$h\left(x\right) := \frac{1}{2} \ln\left(\|A\phi\left(x\right)\|\right) + \frac{1}{2} \ln\left(\|B\phi\left(x\right)\|\right).$$

An dieser Stelle wollen wir die Funktion h graphisch veranschaulichen.

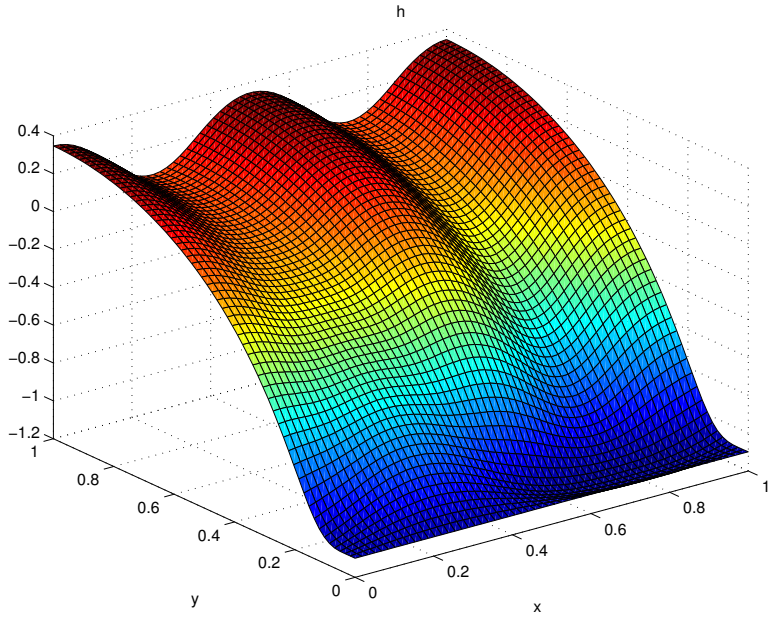


Abbildung 3: Plot der Funktion h auf dem Einheitsrechteck.

Simple Kurvendiskusion ergab - bis auf einen Sattelpunkt - keine Extremalstellen im Inneren von $(0,1)^2$. Daher gilt

$$\sum_{i,j=1}^{n} \min_{x \in \partial I_{i,j}} h(x) \leq \gamma \leq \sum_{i,j=1}^{n} \max_{x \in \partial I_{i,j}} h(x).$$

Wird das Einheitsrechteck nicht zerlegt so erhalten wir hiermit die Abschätzungen

$$0.333076626706253 = \exp\left(\min_{x \in [0,1]} h(x)\right) \leq \sigma \leq \exp\left(\max_{x \in [0,1]} h(x)\right) = 1.414213562373095.$$

Dem aufmerksamen Leser wird aufgefallen sein, dass wir bis jetzt nicht auf die Bestimmung der Toleranzgrenze ε zur Approximation der $c_{k,l}$ eingegangen sind. Dies ist auch das gravierende Problem an dieser Herangehensweise. Es ist nicht gelungen dieses in Abhängigkeit von den Iterationsschritten M zu bestimmen, sodass sich eine sinnvolle Fehlerabschätzung ergab. Diverse Testläufe zeigten außerdem, dass sich bei einer Wahl von $M = 5000$ und und mehreren ε schlussendlich beliebige Maßapproximationen erzeugen lassen. Daher sind die Resultete die mit dieser Methode erzielt wurden schlussendlich zu verwerfen.

3.16 Konklusio

Die Monte Carlo Simulation ergab klar ein Verhalten der Form $|x_n| \approx c^n$ für ein $c \geq 1$. Womit wir behaupten können, dass der gesuchte Grenzwert $\sigma \geq 1$ ist. Darüberhinaus

erhalten wir mit Lemma 3.10 sehr gute obere Schranken. Dabei ist jedoch zu beachten, dass

$$\gamma = \lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\ln\left(\|M_n \cdots M_1\|\right)\right] \tag{28}$$

gilt, womit wir auch wissen, dass die oberen Schranken O_m gegen σ konvergieren. Daher ist es möglich

$$\sigma = \mathbf{1.0}\ldots$$

zu behaupten.

4 Bsp 4

Ausarbeitung: Lisa Mayer

4.1 Substitution

Die Formel hat unendlich viele Nullstellen in der Nähe von 0. Wenn wir mit $y = \frac{1}{x}$ substituieren, können wir die durch $\frac{1}{x}$ verursachte Variation entfernen. Wir erhalten aber dafür ein unendliches Integral. Wegen der Periodizität des Sinus lässt sich die Funktion aber aufsummieren und es ergeben sich wieder endliche Integralgrenzen:

$$\int_1^\infty \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)\frac{1}{x^2}dx$$

Weil die Sinusfunktion periodisch ist, ist das dasselbe wie

$$\int_1^{\pi/2} \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)\frac{1}{x^2}dx + \int_{\pi/2}^{5\pi/2} \sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)\sum_{n=0}^{\infty}\frac{1}{(x+2n\pi)^2}dx$$

Die zweite Polygammafunktion hat die Reihendarstellung

$$\Psi(1,x) := \sum_{k=0}^{\infty}\frac{1}{(x+k)^2}$$

daher lässt sich das rechte Integral folgendermassen in geschlossener Form darstellen:

$$\int_{\frac{\pi}{2}}^{\pi}\left(\frac{\Psi(1,\frac{x}{2\pi})}{4\pi^2} - \frac{\Psi(1,\frac{x}{2\pi}+\frac{1}{2})}{4\pi^2} - \frac{\Psi(1,1-\frac{x}{2\pi})}{4\pi^2} + \frac{\Psi(1,\frac{3}{2}-\frac{x}{(2\pi)})}{4\pi^2}\right)\sin\left(\frac{1}{\sin\left(\frac{1}{\sin(x)}\right)}\right)dx$$

Das linke Integral hat keine Nullstellen, kann also einfach so integriert werden.

4.2 Aufteilung an den Häufungspunkten

Wir können jetzt bestimmen, wo Häufungspunkte von Nullstellen bzw Extremstellen auftreten. Ist $f := \sin(\frac{1}{sin(x)})$, dann findet sich ein Häufungspunkt von Extremalpunkten von $\sin\frac{1}{f(x)}$ an x_0 genau dann, wenn $\lim_{x\to x_0} f(x) = 0$. Man kann jetzt das rechte Integral an den Nullstellen von $\sin(\frac{1}{\sin(x)})$ aufteilen, also an allen Punkten mit $\frac{1}{\sin(x)} = k\pi$. Innerhalb dieser Intervalle oszilliert die Funktion nur noch an den Rändern stark. In diesen Bereichen kann man also die Extremwerte betrachten und über die Intervalle zwischen den Extremstellen integrieren.
Die Intervallgrenzen befinden sich also an $\pi - \arcsin(\frac{1}{k\pi})$, mit $1 \leq k$. Nachdem sich der $\arcsin(x)$ für sehr kleine Werte wie x verhält, würde sich der Fehler bei Abschneiden

ab dem n-ten Intervall wie $\frac{1}{n}$ verhalten. Weiters ist die Reihe der Integrale im wesentlichen alternierend, weil $\sin(-x) = -\sin(x)$ und genau an den Nullstellen von f geteilt wird und der Sinusteil die Funktion für ausreichend kleine Intervalle dominieren wird. Die Extremalpunkte lassen sich ebenfalls explizit berechnen und haben die Form $\pi - \arcsin(\frac{1}{\pm k\pi + arcsin(1/(m\pi))})$. Der Versuch wäre also, zuerst die Integrale zwischen den Häufungspunkten mit Hilfe von Konvergenzbeschleunigung durch das Aitkensche Δ^2-Verfahren zu berechnen und dann das Selbe noch mit dem Gesamtintegral zu machen.
Es ist naheliegend, dass höhere Genauigkeit eine gute Idee sein könnte, wenn unter anderem zweifach aufsummiert wird. Daher haben wir Maple verwendet.
Die kleinsten Teilintegrale haben wir mit Romberg-Integration mit einem Fehler von etwa 10^{-17} berechnet. Wenn man jetzt über die Teilintervalle aufsummiert, kann man den Fehler dabei durch die Differenz bei Summation der letzten beiden Glieder abschätzen. Wir erhalten also die Differenz zwischen dem 40. und dem 41. Glied:

1	2	3	4	5	6	7	8	9
8e-9	6e-9	1e-9	4e-10	2e-10	9e-11	6e-11	4e-11	3e-11

Wir können also den Fehler, wenn wir beispielsweise die ersten 15 Terme summieren einfach durch die Summen der Fehler abschätzen und erhalten etwa 10^{-7}. Der Unterschied zwischen den letzten beiden Summanden beträgt $2 \cdot 10^{-6}$ und das Ergebnis ist 0.496204. Es stimmen hier schon die ersten 4 Stellen mit der Monte-Carlo-Methode überein.
Es wäre jetzt noch möglich, Konvergenzbeschleunigung bei beiden Reihen einzusetzen. Wenn man sowohl bei den Teilintervallen mehrmals das Δ^2-Verfahren einsetzt und dann auch noch für die Ergebnisfolge, erhält man für die Ergebnisfolge:

	Folge	beschleunigte Folge
1	0.4984212965	0.4984212965
2	0.4958197494	0.4961941573
3	0.4963269889	0.4962041857
4	0.4961470315	0.4962012515
5	0.4962307861	0.4962023418
6	0.4961851635	0.4962018653
7	0.4962127162	0.4962020994
8	0.4961948156	0.4962019738
9	0.4962070964	0.4962020460
10	0.4961983079	0.4962020021
11	0.4962048126	0.4962020300
12	0.4961998640	0.4962020117

Man erhält also etwa 7 Stellen, von denen die ersten 4 mit dem Monte-Carlo Ergebnis übereinstimmen. Im Seminar hat sich herausgestellt, dass die weiteren Stellen falsch sind, dieser Ansatz ist also eher gescheitert.

4.3 Monte-Carlo-Integration

Ausarbeitung: Axel Böhm

Angenommen wir wollen folgendes Integral berechnen:

$$I := \int_a^b f(x)\,\mathrm{d}x$$

dann gilt:

$$\int_a^b f(x)\,\mathrm{d}x = (b-a)\int_a^b f(x)\frac{1}{b-a}\,\mathrm{d}x = (b-a)\mathbb{E}(f(X))$$

für eine auf $[a,b)$ uniform verteilte Zufallsvariable X.

Seien nun $X_1, X_2, \ldots, X_n$ unabhängig und uniform verteilt auf $[a,b)$ dann wissen wir, dass das empirische Mittel

$$\overline{Y}_n := \frac{1}{n}\sum_{i=1}^{n} f(X_i)$$

ein unverzerrter Schätzer für $\mathbb{E}(f(X_1))$ ist, der ebenso konsistent ist falls das zweite Moment von $f(X_1)$ existiert.

Des weiteren gilt nach dem ZGVS sogar:

$$\frac{\sqrt{n}(\overline{Y}_n - I)}{S_n} \xrightarrow{d} Y \sim N(0,1)$$

wobei $S_n = \frac{1}{n-1}\sum_{j=1}^{n}(f(X_j) - \overline{Y}_n)^2$.
Hiermit lässt sich ein approximatives $1-\alpha$ Konfidenzintervall bilden

$$\left(\overline{y}_n - z_{1-\frac{\alpha}{2}}\frac{s_n}{\sqrt{n}}, \overline{y}_n + z_{1-\frac{\alpha}{2}}\frac{s_n}{\sqrt{n}}\right)$$

wobei die Kleinbuchbuchstaben die Realisationen der Zufallsvariablen bezeichnen sollen.

4.3.1 Ergebnisse der Monte-Carlo-Integration

Da der Integrand des betrachteten Integrals

$$\int_0^1 \sin\left(\frac{1}{\sin(\frac{1}{\sin(\frac{1}{x})})}\right)\mathrm{d}x$$

sich auf gewissen Teilen des Intervals $[0,1]$ sehr brav verhält, führen wir dort Quadratur durch und nur auf den Bereichen, auf denen die Funktion sehr stark oszilliert, Monte-Carlo-Integration. Diese herangehensweise ergibt den folgenden Wert:

$$0.496213854 \tag{29}$$

Wobei ein 99%-Konfidenzintervall die Richtigkeit der ersten vier Nachkommastellen *garantiert* (unter der Voraussetzung, dass die Quadratur ebensoviele richtige Stellen ergibt) und sogar die fünfte Stelle sich bei mehrmaligen Berechnungen nicht ändert.

5 Beispiel 5

Ausarbeitung: Lisa Mayer

Sei u die klassische Lösung von

$$u_t - \Delta u = e^u \; in \; \Omega = (0,1)^2, \; t > 0$$
$$u = 0 \; auf \; \delta\Omega$$
$$u(0) = 1 \; in \; \Omega.$$

Bestimmen Sie $t^* := \min\{t > 0 : ||u||_\infty = +\infty\}$.

5.1 Theorie und Änderung der Aufgabenstellung

Für unsere Differentialgleichung und $\Omega_T = (0,T] \times \Omega$

$$u_t - \Delta u = f(u,x,t) \; in \; \Omega_T$$
$$u(0,x) = u_0 \; \forall x \in \overline{\Omega_T}$$
$$u(t,x) = 0 \; \forall x \in \delta\Omega_T$$

heisst eine Funktion $v(x,t) \in C(\overline{\Omega}) \cap C^{1,2}(\Omega)$, die

$$v_t - \Delta v \geq f(v,x,t) \; x \in \Omega_T$$
$$v(0,x) \geq u_0 \; \forall x \in \overline{\Omega}_T$$
$$v(t,x) \geq 0 \; \forall x \in \delta\Omega_T$$

erfüllt, eine obere Lösung der Differentialgleichung und für jede Lösung u gilt $u(x,t) \leq v(x,t)$. Insbesondere hat die Differentialgleichung also nur dann einen Blow-up Point, wenn eine Lösung einer modifizierten Differentialgleichung einen hat und der Blow-up-Point dieser bildet eine untere Schranke für den Blow-up-point der ursprünglichen Gleichung.

Es gibt auch das folgende Resultat über die Existenz eines Blow-up Punktes: Siehe Pao, nonlinear partial and elliptical differential equations: Ist $f(u) = \sigma e^{\gamma u}$, dann gibt es eine Konstante $\delta > 0$, mit

$$\frac{\delta}{\gamma} \leq \sigma^* \leq \frac{\lambda_0}{\gamma e}$$

wobei σ^* das Supremum der Werte für σ ist, für die eine positive Lösung existiert (dafür müssen natürlich auch die Anfangswerte ≥ 0 sein) und λ_0 der kleinste Eigenwert des Laplace-Operators ist.

Ist Ω ein Rechteck der Länge a und Höhe b, dann ergeben sich die Eigenwerte und Eigenfunktionen des Laplace-Operators, also die Lösungen von $-\Delta v = \lambda v$ durch

$$\lambda_{n,m} := \pi^2 \left(\frac{n^2}{a^2} + \frac{m^2}{b^2} \right)$$

$$\phi_{n,m}(x,y) := \sin\left(\frac{\pi x}{a}\right) \sin\left(\frac{\pi y}{b}\right)$$

das lässt sich durch Ansatz als $\phi(x,y) = \phi_1(x)\phi_2(y)$ zeigen und damit ergibt sich als kleinster Eigenwert $\lambda_0 = 2\pi^2$ und damit $\sigma^* \leq 7,2616$. Wenn man also für die rechte Seite $f(u) = 8e^u$ verwendet, muss die Lösung für jede positive Anfangsfunktion explodieren.

Man kann folgendermassen zeigen, dass so ein Punkt für die ursprüngliche Aufgabenstellung nicht existiert.Man kann die Gleichung im Eindimensionalen betrachten, also $u_t - \frac{\delta^2}{(\delta x)^2}$ auf $\Omega_1 = (0,1)$ mit $u(0) = 1$. Diese Gleichung lässt sich mit dem pdpde-Solver in Matlab behandeln und man sieht, dass die Lösung beschränkt bleibt. Beschränktheit sieht man daran, dass das Maximum der Funktion nicht monoton steigend ist. Wenn man jetzt die eindimensionale Lösung $v(x,t)$ im zweidimensionalen betrachtet, also $u(x,y,t) := v(x,t)$, dann erfüllt sie die Anfangsbedingung und $u_t - \Delta u = e^u$. Für die Randbedingungen gilt $u \geq 0$, daher ist v eine obere Lösung und damit kann der Punkt t^* nicht existieren.

5.2 Erfolglose FEM

Wir haben jetzt also das modifizierte Problem mit

$$u_t - \Delta u = 8e^u$$

und dieses müsste einen Blow-up-point haben.
Um eine Lösung zu bekommen, erscheint eine Finite Elemente-Methode eine sinnvolle Wahl, das ganze in Matlab zu implementieren war zwar weniger sinnvoll, aber vom Aufwand her vertretbar. Am Ende hatten wir also lineare Elemente und ein implizites Euler-Verfahren. Die sich dabei ergebende nichtlineare Suche einer Nullstelle wurde mit dem Newton-Verfahren gelöst. Wir hatten etwa 2000 Elemente. Das Programm war zwar extrem langsam, aber die Lösung sah zumindest oberflächlich vernünftig aus, hatte aber das Problem, dass sie nicht explodierte. Wir haben also entweder einen Fehler in unserem Programm oder die Methode ist nicht so geeignet. Der Fehler war nicht aufzufinden, daher ist dieser Versuch gescheitert.

5.3 Ein ungenaues Ergebnis

Der letzte Ausweg war wieder Matlab und pdepe. Wenn man das Problem auf dem Kreis mit Radius $\frac{1}{2}$ betrachtet, ist die Lösung des Quadrats eine obere Lösung davon, daher ist auch der Blow-Up-Point des Kreises eine obere Schranke für den Blow-up-Point des

Quadrats. Die Funktion auf dem Kreis lässt sich in Polarkoordinaten transformieren und man erhält das Problem

$$u_t - u_{rr} - \frac{1}{r} = e^u \; auf \; (0, 0.5)$$

$$u_r(0,t) = 0, u(0.5,t) = 0; u(x,0) = 1$$

Auf das ins Eindimensionale transformierte Problem lässt sich wieder pdepe angewenden. Die transformierte Differentialgleichung ist allerdings auch auf $r = 0$ nicht definiert, daher habe ich versucht, das Verhalten bei bei Abschneiden und Annäherung an 0 herauszufinden. Angewendet auf verschiedene Verfeinerungen und Abschneiden sieht die durch Ausprobieren erhaltene untere Schranke für einen Blow-up so aus:

x-Elemente	linker Abstand	t
40	0.001	0.509
40	0.0001	0.508
40	0.00001	0.508
100	0.0001	0.503
200	0.0001	0.503
200	0.00001	0.503

Ich würde also behaupten, dass wir 0.503 als obere Schranke setzen können. Wenn man mit dem selben Argument einen Umkreis um das Quadrat zieht und dadurch versucht, eine untere Schranke zu berechnen, kommt man auf einen Wert von etwa 0.052. Wenn man die Funktion im wieder im Eindimensionalen betrachtet, ergibt sich

x-Elemente	t
40	0.058
100	0.057
500	0.057

Damit erhalten wir die folgende nicht sehr genaue Abschätzung:

$$0.057 \leq t^* \leq 0.503$$

Unsere beste Abschätzung ist also leider nur, dass vor dem Komma eine 0 steht. Für die Kreisscheibe gibt es natürlich keine Aussage über die Existenz eines Blow-ups, es ist also einfach Glück, dass es mit den Anfangswerten funktioniert. Es gibt auch im Fall der Kreisscheibe nicht mehr für alle Anfangswerte ≥ 0 einen Blow-up. Wäre es nicht gegangen, hätte ich das selbe noch mit höheren Werten für σ versucht.

Literatur

[1] Philippe Bougerol and Jean Lacroix. *Products of random matrices with applications to Schrödinger operators*, volume 8 of *Progress in Probability and Statistics*. Birkhäuser Boston, Inc., Boston, MA, 1985.

[2] Mark Embree and Lloyd N. Trefethen. Growth and decay of random Fibonacci sequences. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.*, 455(1987):2471–2485, 1999.

[3] Franklin T Luk and Sanzheng Qiao. A fast eigenvalue algorithm for hankel matrices. *Linear Algebra and Its Applications*, 316(1):171–182, 2000.

[4] Nikol'skiĭ. *Treatise on the shift operator*.

[5] Chia-Ven Pao. *Nonlinear parabolic and elliptic equations*. Springer, 1992.

[6] Vladimir V Peller. An excursion into the theory of hankel operators. *Holomorphic spaces (Berkeley, CA, 1995), Math. Sci. Res. Inst. Publ*, 33:65–120, 1998.

[7] Divakar Viswanath. Random Fibonacci sequences and the number 1.13198824 . . .. *Math. Comp.*, 69(231):1131–1155, 2000.