

Universität Bielefeld Fakultät für Physik	Computerphysik SS 2016	Prof. Dr. Jürgen Schnack jschnack@uni-bielefeld.de
--	---------------------------	---

Aufgabenblatt 6

6.1 Taylor-Reihe versus Padé-Approximation (Hausaufgabe + Email, Einsendeschluss Mitternacht vor der Übung)

Für die Lösung dieser Aufgabe können Sie Mathematica verwenden.

- Wiederholen Sie zu Hause, wie Padé-Approximationen über eine gegebene Reihenentwicklung einer Funktion berechnet werden können. Schauen Sie sich insbesondere an, wie man Padé-Approximationen in beliebiger Ordnung über Determinanten bestimmen kann.
- Berechnen Sie für die Taylor-Reihe (von $\ln(1+x)$)

$$f(x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} \quad (1)$$

die zugehörigen Padé-Approximationen $[M, N](x)$ mit $M, N = 0, 1, 2$. Sie können die benötigten Determinanten z.B. mit Mathematica ermitteln.

- Stellen Sie die Padé-Approximationen zusammen mit der Taylor-Entwicklung in verschiedenen Ordnungen und der Funktion $f(x)$ für $x \in [-1, 5]$ graphisch dar.
- Vergleichen und diskutieren Sie die Ergebnisse.
- Zusatzaufgabe I:** Schreiben Sie ein Mathematica-Notebook, das für eine vorgegebene Reihenentwicklung einer Funktion Padé-Approximationen in beliebiger Ordnung berechnen kann.
- Zusatzaufgabe II:** Sie können das von Ihnen entwickelte Programm mit einem veröffentlichten Fortran-Programm aus dem Jahre 1972 vergleichen [1]. ;-)

6.2 ϵ -Algorithmus (Hausaufgabe + Bearbeiten in den Übungen)

Die für diese Aufgabe zur Verfügung gestellten Programme finden Sie in stud.ip.

- Wiederholen Sie, wie der ϵ -Algorithmus funktioniert. Schauen Sie sich vor allem noch einmal die Rekursionsvorschrift an.
- Für die Verwendung des ϵ -Algorithmus steht das Hauptprogramm `epsilon.c` zur Verfügung. Dieses liest die zu extrapolierenden Datenpunkte mit Hilfe der bekannten Funktion `readin.h` in die Feldkomponenten `Epsilon[p=0][k]` ein.

Schreiben Sie die noch fehlende Unterfunktion `epsilon.h` für den ϵ -Algorithmus und verwenden Sie dabei den folgenden Prototyp:

```
void epsilon(int N, double Epsilon[NMAX][NMAX], double *bestApproximation);
```

`N` gibt hier die Anzahl der zur Verfügung stehenden Datenpunkte an, die im Feld `Epsilon[p=0][k]` mit $k = 0, 1, \dots, N - 1$ stehen. `NMAX` ist ein im Hauptprogramm definiertes Makro, das die maximale Anzahl von Datenpunkten festlegt. Die beste Approximation, die über den ϵ -Algorithmus berechnet werden kann, wird über den Pointer `bestApproximation` in `*bestApproximation` geschrieben. Beachten Sie beim Schreiben der Funktion, dass im ϵ -Algorithmus auch der Index $p = -1$ auftritt und behandeln Sie diesen Fall richtig. (Mehrdimensionale Felder sind im übrigen ein interessantes Thema, da sie eindimensional abgelegt werden.)

Was ist hier mit „bester“ Approximation gemeint? Welche Einschränkung gibt es beim ϵ -Algorithmus?

- c. Das Problem der Extrapolation der Grundzustandsenergie für Spin-1/2-Ringe mit antiferromagnetischer Nächster-Nachbar-Kopplung im Heisenberg-Modell ist in der Vorlesung angesprochen worden. Untersuchen Sie die Folge der Grundzustandsenergien $E_0(N)/N$ für Ringe der Länge N mit dem Programm `epsilon.c`. Die Wertepaare $(N, E_0(N))$ finden Sie in der Datei `energien.dat`. Erzeugen Sie die benötigten Datenpaare $(N, E_0(N)/N)$, extrapolieren Sie diese mit dem ϵ -Algorithmus und vergleichen Sie das Ergebnis mit dem bekannten Grenzwert für $N \rightarrow \infty$.
- d. Für die Anregungsenergien (Singlett-Triplett-Energielücke) von Spin-1-Ringen mit antiferromagnetischer Nächster-Nachbar-Kopplung gibt es bisher keinen analytisch bekannten Grenzwert. Es existiert nur die Vermutung, dass der Grenzwert echt größer Null ist (Haldane-Vermutung [2, 3, 4]). Generieren Sie mit dem Programm `epsilon.c` eine Abschätzung für das „Haldane gap“, d.h. den Grenzfall $\lim_{N \rightarrow \infty} \Delta E(N)$. Die Anregungsenergien $(N, \Delta E(N))$ stehen ebenfalls in `energien.dat`.
- e. Was geschieht, wenn man dem ϵ -Algorithmus Elemente einer nicht konvergenten Folge übergibt? Denken Sie sich doch einfach mal eine aus, z.B. n^2 , und überprüfen Sie, was Ihr Programm tut.

Literatur

- [1] I. Longman, Intern. J. Computer Math. **3**, 53 (1972).
- [2] F. D. M. Haldane, Phys. Lett. A **93**, 464 (1983).
- [3] F. D. M. Haldane, Phys. Rev. Lett. **50**, 1153 (1983).
- [4] J. Schnack, Phys. Rev. B **62**, 14855 (2000).