

Numerische Mathematik– 9. Übung

Aufgabe 1 (Aitken's Δ^2 -Verfahren in \mathbb{R})

Sei $x_0, x_1, \dots \in \mathbb{R}$ eine linear konvergente Folge mit Grenzwert x_* , so dass $x_k - x_* = \gamma(x_{k-1} - x_*)$ mit $|\gamma| < 1$.

a) Zeigen Sie:

$$\gamma = \frac{x_{k+2} - x_{k+1}}{x_{k+1} - x_k}$$

und

$$x_* = \frac{x_{k+2}x_k - x_{k+1}^2}{x_{k+2} - 2x_{k+1} + x_k}.$$

b) Definieren Sie $\Delta x_k := x_{k+1} - x_k$ und $\Delta^2 x_k = \Delta(\Delta(x_k))$. Zeigen Sie, dass gilt:

$$x_* = x_k - \frac{(\Delta x_k)^2}{\Delta^2 x_k}.$$

c) Sei nun $(y_i)_{i \in \mathbb{N}}$ eine Folge in \mathbb{R} mit Grenzwert y_* , $y_i \neq y_*$ und

$$y_{k+1} - y_* = (\kappa + \delta_k)(y_k - y_*),$$

für $|\kappa| < 1$, sowie $(\delta_i)_{i \in \mathbb{N}}$ eine Nullfolge. Es sei weiter

$$\bar{y}_k := y_k - \frac{(\Delta y_k)^2}{\Delta^2 y_k}.$$

Zeigen Sie

$$\lim_{k \rightarrow \infty} \frac{\bar{y}_k - y_*}{y_k - y_*} = 0$$

d) Die beschleunigt konvergente Folge ist i.a. wieder linear konvergent, so dass man das Verfahren erneut anwenden kann usw. Warum ist das in endlicher Arithmetik nur begrenzt sinnvoll?

Aufgabe 2 (Bildatenkompression mit SVD)

Schreiben Sie ein MATLAB-Programm, welches eine Matrix mit Bilddateninformation einliest, eine beste Approximation vom Rang k an diese Matrix berechnet, sowie den Approximationsfehler (in der 2-Norm), das Original und das komprimierte Bild darstellt (**Hinweis:** Verwenden Sie `subplot`), sowie den für das Original und das komprimierte Bild benötigten Arbeitsspeicher ausgibt. Testen Sie das Programm für die in MATLAB verfügbaren Bilder (`clown`, `gatlin`, `durer`, `mandrill`, `earth`) und verschiedene Werte von k .

Ermitteln Sie (empirisch) für jedes Bild die beste Kompression, so dass das approximierte Bild noch optisch verlustfrei dargestellt werden kann.

Hinweis: Die MATLAB-Bilder erhält man mit dem `load`-Befehl. Z.B. erhält man mit

```
>> load clown
```

eine Bilddatenmatrix `X` (eine 200×320 -Matrix in diesem Fall, also hat das Clown-Bild 200×320 Bildpunkte) sowie eine Matrix `map`, in der die Information über das verwendete Farbmodell gespeichert ist. Um nun das Clown-Bild anzuzeigen, kann man die Befehle

```
>> colormap(map)
```

```
>> image(X)
```

ausführen.