Aufgabenblatt 7

7.1 Untersuchung des Konvergenzverhaltens

Gesucht ist eine nichttriviale Nullstelle, d.h. $\xi \neq 0$, von

$$f(x) = \sqrt{x} - \tan(x). \tag{1}$$

a. **Wiederholen Sie, wie man aus dieser Funktion eine Rekursionsformel für die Nullstellensuche generieren kann. Welche Bedingungen muss die die Rekursionsformel definierende Abbildung ϕ erfüllen?**

b. Schreiben Sie ein kurzes Programm, das die folgenden Rekursionsformeln verwendet:

   (1) $x = \tan^2(x)$;

   (2) $x = \arctan(\sqrt{x})$;

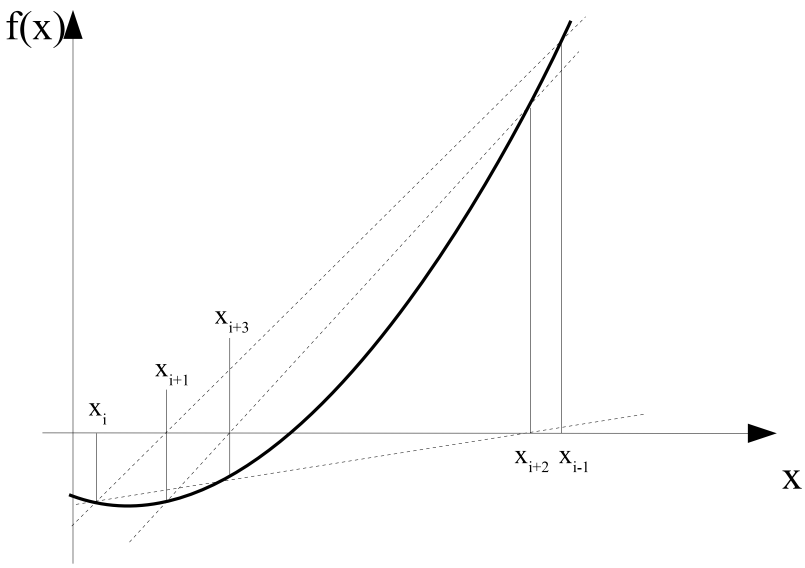
   (3) $x = x - \frac{f(x)}{A}$, $A = f'(x_0)$.

   Dabei sei $x_0 = 0.6$. Begründen Sie das Konvergenzverhalten.

7.2 Sekantenmethode

a. **Wiederholen Sie die Sekantenmethode.**

b. Erklären Sie, was an der Graphik aus dem Skript nicht stimmt. Generieren Sie eine neue Graphik, speichern Sie diese als Encapsulated Postscript ab und schicken Sie mir die Graphik per Email. Die beste Graphik schafft es ins verbesserte Skript.



7.3 Nullstellen der Besselfunktion

Bestimmen Sie die ersten drei Nullstellen der Besselfunktion (erster Art) $J_0(x)$ für $x >= 0$.

a. Die Routine `bessj0.h` approximiert die Besselfunktion $J_0(x)$. Für $x < 8.0$ wird $J_0(x)$ hierzu durch eine rationale Funktion angenähert, während für $x >= 8.0$ eine approximative Form bestehend aus gewissen Polynomen sowie Cosinus- und Sinus-Funktionen verwendet wird (siehe Numerical Recipes in C, Kapitel 6.5). Plotten Sie die Funktion für $x >= 0$ und suchen Sie jeweils ein geeignetes Startintervall $[x_1, x_2]$ für die Nullstellensuche.

b. Schreiben Sie dann unter Verwendung der Funktionen `zbrent.h` und `bessj0.h` ein Programm zur Bestimmung der Nullstellen von $J_0(x)$ mit vorgebbarer Genauigkeit. Überprüfen Sie Ihre Ergebnisse für die Nullstellen mit Mathematica. Beachten Sie dabei, dass die Genauigkeit der von Ihnen berechneten Nullstellen durch die Genauigkeit der in `bessj0.h` auftretenden Konstanten begrenzt ist.

   Die Funktion `zbrent.h` erwartet als Argument u.a. einen Pointer auf eine Funktion und kann damit für die Nullstellensuche bei beliebigen Funktionen benutzt werden. Die Funktion verwendet den Van Wijngaarden-Dekker-Brent-Algorithmus, der sog. *root bracketing*, Bisektion und inverse quadratische Interpolation für die Bestimmung von Nullstellen kombiniert. Diese Methode hat in unproblematischen Fällen den Vorteil hoher Geschwindigkeit und garantiert auch in problematischen Fällen eine Konvergenz der Nullstellensuche (siehe Numerical Recipes in C, Kapitel 9.3).

c. Berechnen Sie die Nullstellen auch mit einer von Ihnen geschriebenen Unterfunktion, die die Regula falsi verwendet. Programmieren Sie die Funktion so, dass sie wie `zbrent.h` verwendet werden kann.