HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN MÜNCHEN

Testat: Programmieraufgabe: Numerische Mathematik

Prof. Dr. Michael Wibmer

Aufgabe ist die numerisch stabile Berechnung der Kreiszahl π. Wir berechnen π nach Archimedes von Sykarus, in dem wir π über den Umfang A_n eines regelmäßigen 2^n-Ecks annähern $(n = 2, 3, 4, \dots)$.
Der Umfang U_k eines regelmäßigen, dem Einheitskreis eingeschriebenen k-Ecks lässt sich durch trigonometrische Zusammenhänge berechnen als $U_k = k\sin(\frac{\pi}{k})$. Man kann zeigen, dass

$$\lim_{k\to\infty} U_k = \lim_{k\to\infty} k\sin\left(\frac{\pi}{k}\right) = \pi. \tag{1}$$

Da π in der Zahlenfolge selbst auftritt, ist dieser Grenzwert natürlich für die Berechnung von π unbrauchbar.

Für $n = 2$ wird ein Quadrat in den Einheitskreis eingeschrieben. Der Umfang berechnet sich zu (Skizze hilft!)

$$A_2 = U_{2^2} = 4\sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = 2\sqrt{2}.$$

Das Verfahren von Archimedes von Sykarus lässt sich als rekursive Iterationsvorschrift

$$A_{n+1} = 2^n\sqrt{2\left(1-\sqrt{1-\left(\frac{A_n}{2^n}\right)^2}\right)}, \quad n = 2, 3, 4, \dots \tag{2}$$

ausdrücken. Eine mathematische Fehlerabschätzung bzgl. A_n lässt sich wie folgt angeben: Der Taylor'sche Satz liefert die Abschätzung $(h > 0)$

$$|\sin(h) - h| \le \frac{h^3}{6}.$$

Setzt man nun $h = \frac{\pi}{k}, \;\; k = 1, 2, 3, \dots$ so folgt aus (1) die Abschätzung

$$|k\sin\left(\frac{\pi}{k}\right) - \pi| \le \frac{\pi^3}{6k^2}. \tag{3}$$

Da A_n der Umfang des regelmäßigen 2^n-Ecks ist, folgt mit (3) eine Abschätzung für A_n (setze $k = 2^n$)

$$|A_n - \pi| \le \frac{\pi^3}{4^n \cdot 6} \tag{4}$$

Für $n \ge 18$ ist der Fehler kleiner 10^{-10}.

Eine etwas andere Form des Algorithmus von Archimedes von Sykarus ist die rekursive Iterationsvorschrift in der Variablen Z_n

$$\begin{aligned} B_n &= \frac{2\left(\frac{Z_n}{2^{n+1}}\right)^2}{1+\sqrt{1-\frac{Z_n}{2^n}^2}} \\ R_n &= 4B_n \\ Z_{n+1} &= 2^n\sqrt{R_n} \end{aligned}$$

mit dem Startwert $Z_2 = 2\sqrt{2}$ und $n = 2, 3, 4, \ldots$.

Die Programmieraufgabe besteht nun darin, die Iterationsvorschrift für A_n und Z_n zu implementieren. Resultat ist ein Plot, in welchem die Fehler

$$A_{err} = |A_n - \pi| \tag{5}$$
$$Z_{err} = |Z_n - \pi| \tag{6}$$
$$M_{err} = \frac{\pi^3}{6 \cdot 4^n} \tag{7}$$

halblogarithmisch über $n = 2, \ldots, 30$ dargestellt sind. Wählen Sie für die y-Achse den Zahlenbereich von 10^0 bis zur Maschinengenauigkeit.

Schreiben Sie eine Matlab-Funktion, welche die drei Fehlergrößen (5) bis (7) berechnet.

Vorgangsweise zur Abgabe:
Sie benennen die Funktion **iteration_archimedes_ID.m** entsprechen der Beschreibung in der Funktionsdokumentation und geben diese Datei in Moodle ab. Diese Datei wird dann von einem Auswerteskript **runit.m** ausgeführt. Das Programm **runit.m** plottet die drei Ausgabevektoren der von Ihnen implementierten Funktion. Das Programm **runit.m** steht Ihnen als Download in Moodle zur Verfügung. Zur Benützung von **runit.m** müssen Sie Ihren Funktionsnamen anpassen.

Es wird überprüft, ob der Plot den korrekten Sachverhalt darstellt. Sie müssen also vor der Abgabe überlegen, ob der Plot inhaltlich korrekt ist!

```
[A_err, Z_err, M_err] = iteration_archimedes_ID(startvalue, n_end)
% Implementiert die beiden Iterationen A_n, Z_n und
% speichert die Fehler A_err, Z_err, M_err in einen Vektor
%
%
% ERSETZEN SIE ID IM FUNKTIONSNAME DURCH IHRE INITALEN (DATEINAME UND
    FUNKTIONSNAME) !
% VOR DER ABGABE TRAGEN SIE IHREN NAMEN UND STUDIENGANG HIER IN DEN
    FUNKTIONSKOMMENTAR EIN.
%
% NACHNAME, VORNAME, STUDIENGANG
%
% Uebergabeparameter:
% startvalue ... Startwert der Iterationen A_n und Z_n, z.B. 2\sqrt{2}
% n_end ... Maximaler Iterationswert, z.B. n=30
%
% Ergebnis:
%
% Drei SPALTEN-Vektoren der Laenge n_end-1 fuer
% die Iterationsschritte n=2,3...,n_end
% A_err ... Fehler-Spaltenvektor A_err
% Z_err ... Fehler-Spaltenvektor Z_err
% M_err ... Fehler-Spaltenvektor M_err
```

```
%
% Der erste Eintrag der Arrays beinhalten den Fehler bzgl. n=2, also dem
    Startwert.
% A(1) ... Fehler bzgl. A_2
% A(2) ... Fehler bzgl. A_3
% A(i) ... Fehler bzgl. A_n mit n=i+1
% i = 1 ... n_end-1
%
% Analog fuer Z.
%
%
%
```