Lernziele

In diesem Praktikum sollen Sie üben und lernen:

- Numerische Berechnung von Nullstellen
- Graphische Veranschaulichung von numerischen Verfahren

Am Anfang wollen wir einige Fragen über das Lösen von nichtlinearen Gleichungen stellen.

Machen Sie bitte zuerst die folgenden Offline-Übungen bevor Sie sich einloggen!

Offline Aktivitäten

Nichtlineare Gleichungen

1. Welche Verfahren zur Lösung von nichtlinearen Gleichungssystemen kennen Sie?

   Ihre Antwort:

2. Welche Vor- und Nachteile haben diese Verfahren (auch Konvergenzgeschwindigkeit)?

   Ihre Antwort:

Matlab-Befehle

3. Was machen die Befehle `pause` und `pause(t)`?

   Ihre Antwort:

4. Wofür verwendet man die Befehle `tic` und `toc` sowie `cputime`?

   Ihre Antwort:

5. Was gibt `nargin` an?

   Ihre Antwort:

Praktikumsaufgabe 1 - Regula Falsi

Sei $I \subset \mathbb{R}$ ein Intervall. Implementieren Sie eine Funktion `regula`, die zu einer stetigen Funktion $f : I \to \mathbb{R}$ und zwei Startwerten $a, b \in I$ mit $f(a)f(b) < 0$ die Werte ξ, a bzw. ξ, b mit

$$\xi = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

ausgibt, sodass f eine Nullstelle zwischen ξ und a bzw. ξ und b hat. Dabei muss nicht notwendigerweise $a < b$ gelten. Verwenden Sie als Eingabeparameter ein Funktions-Handle `func` für f und einen Vektor `init` der Länge 2, der die Werte a und b enthält. Der Ausgabeparameter `out` soll ein Vektor der Länge 2 sein, dessen erster Wert ξ ist und der zweite Wert a bzw. b ist. Die Funktion beginnt also mit:

```
function out=regula(func,init)
```

Praktikumsaufgabe 2 - Sekantenverfahren

Implementieren Sie analog zu Praktikumsaufgabe 1 eine Funktion `sekante`, die zu einer stetigen Funktion $f : I \to \mathbb{R}$ und zwei Startwerten $a, b \in I$ die Werte ξ, a mit

$$\xi = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

ausgibt. Verwenden Sie als Eingabeparameter ein Funktions-Handle `func` für f und einen Vektor `init` der Länge 2, dessen erster Wert a und zweiter Wert b ist. Der Ausgabeparameter `out` soll ein Vektor der Länge 2 sein, dessen erster Wert ξ ist und der zweite Wert a ist. Die Funktion beginnt also mit:

```
function out=sekante(func,init)
```

Praktikumsaufgabe 3 - Newtonverfahren

Implementieren Sie eine Funktion `newton`, die zu einer stetigen Funktion $f : I \to \mathbb{R}$ und zu einem Startwert $x \in I$ den Wert y mit

$$y = x - \frac{f(x)}{f'(x)}$$

ausgibt. Verwenden Sie als Eingabeparameter ein Funktions-Handle `func` für f, einen skalaren Wert `init` für x, sowie einen Funktions-Handle `abl` für f'. Der Ausgabeparameter `out` soll y ausgeben. Die Funktion beginnt also mit:

```
function out=newton(func,init,abl)
```

Praktikumsaufgabe 4 - Grafische Veranschaulichung

Die Funktion `myplot` soll nun die Verfahren zum Lösen des Nullstellenproblems graphisch veranschaulichen und die berechneten Zwischenwerte ausgeben.
Verwenden Sie dabei folgende Vorlage die Sie auf der Vorlesungshomepage finden:

```
function T=myplot(left,right,func,methode,init,n,abl)

z=init;
x=left:(right-left)/1000:right;
y=func(x);
T=z(1);

for i=1:n
    if nargin==7
        z_new=methode(func,z,abl);
    else
        z_new=methode(func,z);
    end
    % hier wird f, x-Achse und Sekante (regula, sekante) bzw.
    % Tangente (newton) geplottet
    hold off
    z=z_new;
    T=[T;z(1)];
    pause(1)
end
```

Die Parameter `left` und `right` beschreiben den Bildbereich, in dem geplottet wird. `func` ist ein Funktions-Handle, von dem die Nullstelle berechnet werden soll und das Funktions-Handle `abl` sei die Ableitung von `func`. Das Verfahren wird als Funktions-Handle `methode` übergeben (hier: `regula`, `sekante` oder `newton`). Der Parameter `init` ist je nach Verfahren ein skalarer Wert oder ein Vektor der Länge 2. Wieviele Iterationsschritte durchgeführt werden sollen, wird durch `n` angegeben. Ersetzen Sie den Kommentar in der `for`-Schleife durch den Plot von `func`, x-Achse und der Geraden, die das Verfahren veranschaulicht. Dabei benötigen Sie für die y-Werte der Geraden nur die Werte `z(1)`, `z_new(1)` und `func(z(1))` und ist unabhängig von der Wahl von `methode`. Z. B. soll bei der Verwendung von `methode=@newton` die Tangente an die Funktion `func` an der Stelle `z` eingezeichnet werden. Testen Sie Ihr Programm mit:

```
myplot(0.1,2,@(x)log(x),@regula,[0.1,2],10)
myplot(0.1,2,@(x)log(x),@sekante,[0.1,2],10)
myplot(0.1,2,@(x)log(x),@newton,0.1,10,@(x)1./x)
```

Testen Sie mit Ihrem Programm das Newton-Verfahren für die Funktion $f(x) = \arctan(x)$ mit Startwerten $x_0 \in \{1, 1.2, 1.3, 1.38, 1.4, 1.4\}$. Beschreiben Sie was sie sehen im Hinblick auf die *l*okale Konvergenz des Newton Verfahrens.

Praktikumsaufgabe 5 - Konvergenzgeschwindigkeit

Schreiben Sie ein Skript `error_method`, in dem die Konvergenzgeschwindigkeiten von Regula Falsi, Sekantenverfahren und Newton-Verfahren untersucht werden. Verwenden Sie als Funktion, von der numerisch die Nullstelle ermittelt werden soll, $f(x) = x^3$ sowie als Startwerte für Regula Falsi und das Sekantenverfahren $x_1 = -0,5$ und $x_2 = 1$ und als Startwert für das Newton-Verfahren $x_1 = 1$. Stellen Sie die Fehler der ersten 20 Iterationswerte der drei Verfahren grafisch auf einer halblogarithmischen Skala (`semilogy`) dar. Was lässt sich über die Konvergenzgeschwindigkeit aussagen? Kennen Sie ein Verfahren, dessen Konvergenzgeschwindigkeit in so einem Fall höher wäre?

Hinweis. Sie dürfen hierfür die erstellten Funktionen aus den Praktikumsaufgaben 1,2,3 und 4 verwenden.