1. Aufgabe für OOP

Termin: 11. Okt. 2011, 10:00 Uhr

Das Ziel dieser kleinen Projektarbeit besteht darin, daß Sie eine Projekt selbständig entwerfen und programmieren. Hierbei sollten Sie:

- die Source- und Headerfiles einer gegebenen Bibliothek in Ihr Projekt einfügen und die darin enthaltenen **Funktionen** auch **nutzen**,
- Funktionen für eine weitere, eigene Bibliothek (Source und Headerfile) bzgl. der gegebenen Aufgabenstellung programmieren,
- mit Eingabe- und Ausgabefiles arbeiten,
- Ihre einzelnen Module/Funktionen selbständig testen.
- Ihren Code ausreichend dokumentieren, in den Headerfiles die Parameterliste erläutern und beschreiben, was die jeweilige Funktion macht (Ich benutze den Stil von doxygen; kurze Einführung Download).

Was soll Ihr Code können?

a) Diskrete x-Werte von den gegebenen ASCII-Files einlesen (Modul *file_io* benutzen).

b) Für mehrere gegebene Funktionen f, $\forall x$ die diskreten Funktionswerte $f(x)$, die diskreten Ableitungswerte $f'(x)$ und die diskreten Stammfunktionswerte $F(x) = \int\limits_0^x f(x)dx$ **numerisch** ausrechnen.
Dafür sollen Sie ein Modul (Header-und Sourcefile) für Differentiation und Integration schreiben, wobei die jeweils auszuwertende Funktion als Funktionsparameter übergeben werden muß.

c) Diese Werte sind für die betrachteten x-Werte in drei separaten Files abzuspeichern und mit Matlab visualisieren. Nutzen Sie die Matlab-Fkt `importdata` um jeweils einen Vektor von einem ASCII-File einzulesen.

d) Beim File-I/O sind die Files aus dem Modul *file_io* als zusätzliche Files in Ihr Projekt einzubinden und zu verwenden.

Zu benutzende Funktionen:

$$p(x) = 4x^3 + 3x^2 \quad ; \qquad q(x) = \mathrm{e}^{-x}\sin(40x) \quad ; \qquad s(x) = x\sin(40/x)$$

Files mit Inputwerten für x: *input_1_x.txt* und *input_2_x.txt*.

Sie sollen also a)-d) für 2 verschiedene Inputdaten und 3 verschiedene Funktionen anwenden. Da lohnt sich schon eine weitere Funktion!

Nun zu den benötigten Formeln:

Numerische Integration

Die numerische Integration einer Funktion $f(x)$ erfolgt näherungsweise, z.B., über die Riemannsummen

$$\int_a^b f(x)dx \approx \sum_{j=1}^{n} f(a+jh) \cdot h \tag{1}$$

wobei n die Anzahl der äquidistanten (gleichgroßen) Teilintervalle von $[a,b]$ mit der Teilintervalllänge $h = \frac{b-a}{n}$ ist.
Natürlich hängt die Genauigkeit der numerischen Integration von der gewählten Anzahl der Teilintervalle n ab, d.h., **Sie müssen** n ***solange*** **vergrößern** bis sich Ihre Summe für das neue n von der vorherigen Summation nur noch um einen kleinen Wert unterscheidet (ε). Sei die rechte Seite in (1) mit $F_n(x)$ bezeichnet, so gehen Sie nun in Richtung des Grenzwertes $\lim\limits_{n\to\infty} F_n(x)$, allerdings nur so weit wie nötig.
Vorschlag für die Parameterliste der Integrationsfunktion: f, a, b, ε.
Testen Sie Ihre Funktion mit leicht nachzurechnenden Daten!

Numerische Differentiation

Die numerische Differentiation einer Funktion f im Punkt x kann, z.B., über die zentrale Differenz

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \tag{2}$$

numerisch dem wahren Wert angenähert werden. Die Schrittweite h muß hierbei vorgegeben werden. Für unsere Aufgabe ist die Wahl $h = 0.1$ ein guter Start.
Wiederum soll die numerische Differentiation genau genug werden, d.h., **Sie müssen** $h > 0$ ***solange*** **verkleinern** bis sich die beiden letzten Werte um weniger als ein ε unterscheiden. Sei die rechte Seite in (2) mit $f'_h(x)$ bezeichnet, so entspricht dieses Vorgehen einer numerischen Annäherung in Richtung des Grenzwertes $\lim\limits_{h\to 0} f'_h(x)$, allerdings nur so weit wie nötig.
Vorschlag für die Parameterliste der Integrationsfunktion: f, x, ε.
Testen Sie Ihre Funktion mit leicht nachzurechnenden Daten!
