

Arbeit mit Unterprogrammen und Dateien

- Das Programm zum Testen der **Simpsonregel** ist so zu erweitern, dass die Ergebnisse als Tabelle in eine Datei geschrieben werden.
- Neben der **Simpsonregel** soll die **Trapezregel** für jeweils die gleiche Anzahl Teilintervalle angewendet werden, um die Ergebnisse zu vergleichen.

Trapezregel:

$$\int_a^b f(x) dx \approx \frac{h}{2} \sum_{k=0}^n c_k f(x_k), \quad \text{mit}$$

$$h = \frac{(b-a)}{n}, \quad x_k = a + kh, \quad c_k = \begin{cases} 1, & k = 0 \text{ bzw. } k = n \\ 2, & k = 1, \dots, n-1 \end{cases}$$

- Zum **Öffnen der Datei** für die Ausgabe der Tabelle ist ein spezielles Unterprogramm zu schreiben, das vom Hauptprogramm durch

```
call Neue_Datei(LUN,IER)
```

gerufen wird und das folgendes realisieren soll:

- Aufforderung zur Eingabe des Dateinamens (Einlesen auf eine Zeichenkettensvariable)
- Warnung, falls die Datei schon existiert (wahlweise trotzdem überschreiben oder mit IER=2 Fehler an Hauptprogramm melden)
- Bestimmung (mittels **INQUIRE**) einer freien Dateinummer **LUN**, der noch keine Datei zugewiesen ist
- Öffnen der Datei und Rückgabe von IER=0, wenn das fehlerfrei war, sonst IER=1

Das Unterprogramm ist damit für andere Anwendungen wiederverwendbar!

- Ausgabe der Tabelle im Hauptprogramm (natürlich nur im Falle **IER=0**) dann mit

```
write(LUN,*) n,Simpson(a,b,n,f),Trapez(a,b,n,f)
```

- Nachdem das Hauptprogramm die Ergebnisse als Tabelle in eine Datei geschrieben hat, soll es noch eine zweite Datei schreiben, in der die notwendigen Steuerkommandos für Gnuplot stehen:

```
set data style linespoints
plot "datei" using 1:2 title "Simpson",\
     "datei" using 1:3 title "Trapez"
```

Dabei ist **datei** der Name der zuvor geschriebenen Datei mit den berechneten Ergebnissen.

- Da das Hauptprogramm nur noch die Dateinummer der zu schreibenden Datei kennt, muss der Dateiname hier nochmals mit `INQUIRE` bestimmt werden, um ihn in die Steuerdatei zu schreiben.

- **HINWEIS:**

Zeichenketten werden in Fortran immer mit der vereinbarten Länge verarbeitet. Man benötigt aber hier nur die Teilzeichenkette, `name(1:L)`, ohne abschließende Leerzeichen. Da dieser Fall immer wieder mal vorkommt, sollte man eine Funktion

```
L=LEN_TRIM (string)
```

benutzen, welche die vereinbarte Länge `LEN(string)`, vermindert um die Anzahl der abschließenden Leerzeichen als Funktionswert liefert.

(`LEN_TRIM` könnte man leicht selbst schreiben, ist aber in gfortran als Standardfunktion vorhanden)

- Am Ende `CLOSE(LUN)` nicht vergessen.