

RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE  
INSTITUT FÜR GEOMETRIE UND PRAKTISCHE MATHEMATIK  
**Mathematisches Praktikum (MaPra) — Sommersemester 2011**

Prof. Dr. Wolfgang Dahmen — M.Sc. Mathieu Bachmann, Dipl.-Math. Jens Berger, M.Sc. Liang Zhang

### Aufgabe 3

**Bearbeitungszeit:** bis Montag, den 9. Mai 2011

**Testattermin:** Donnerstag, der 12. Mai 2011

**Mathematischer Hintergrund:** Quadratur

**Elemente von C++:** Schleifen, arrays, Unterprogramme, Ein-/Ausgabe, Einbinden einer Grafikbibliothek

**14 Punkte**

---

#### Aufgabenstellung und Verfahren

Es ist ein Programm zu schreiben, das zu einer gegebenen Funktion  $f \in C^{2m+2}[a, b]$  den Wert des Integrals  $\int_a^b f(x) dx$  bis auf einen geschätzten Fehler von einem vorgegeben  $\varepsilon$  berechnet. Als Verfahren wird die summierte Trapezregel mit optionaler Extrapolation gewählt.

Bei der summierten Trapezregel wird das Intervall  $[a, b]$  in  $N$  Teile unterteilt und mit  $h = \frac{b-a}{N}$  gilt

$$h \left( \frac{1}{2}f(a) + \sum_{i=1}^{N-1} f(a + i h) + \frac{1}{2}f(b) \right) = \int_a^b f(x) dx + \sum_{i=1}^m c_i h^{2i} + d_{m+1}(h) h^{2m+2}$$

mit von  $N$  unabhängigen Werten für die  $c_i$ . Man spricht deswegen von einer *Fehlerentwicklung in  $h^2$* . Für geeignete Folgen von  $h$ -Werten ( $h_0, h_1, \dots, h_i, \dots$ ) werden dann Näherungen für das Integral berechnet. Wir wollen diese Näherungen mit  $T_{i,0}$  bezeichnen. Durch Linearkombinationen der  $T_{i,0}$  lassen sich Näherungen der Ordnung  $2^{2k+2}$  erreichen. Wie man leicht nachrechnet, muß man dazu wie folgt verfahren

$$T_{i,k} = T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left(\frac{h_{i-k}}{h_i}\right)^2 - 1} \quad i = 0, 1, 2, \dots \quad \text{und} \quad k = 1, 2, \dots, i$$

Dieser Vorgang heißt Extrapolation. Als Schätzung des Fehlers (nach dem  $i$ -ten Schritt mit Extrapolation) kann man die Differenz von  $T_{i,i}$  und  $T_{i,i-1}$  nehmen

Geeignete Schrittfolgen sind

nach Romberg:

$$h_0 := b - a, \quad \text{und} \quad h_i := \frac{h_{i-1}}{2} \quad \text{für} \quad i = 1, 2, \dots$$

nach Bulirsch:

$$h_0 := b - a, \quad h_1 := \frac{h_0}{2}, \quad h_2 := \frac{h_0}{3}, \quad \text{und} \quad h_i := \frac{h_{i-2}}{2} \quad \text{für} \quad i = 3, 4, \dots$$

Sie sollen nun ein Programm schreiben, das mit der summierten Trapezregel zur Romberg–Schrittweitenfolge (der Fehler kann dann zu  $\frac{T_{i,0}-T_{i-1,0}}{3}$  geschätzt werden) und zu beiden obigen Schrittweitenfolgen mit Extrapolation einen Näherungswert für bestimmte Integrale berechnet. Dabei soll die Funktion  $f$  an jeder Stelle höchstens einmal ausgewertet werden (es gibt Rekursionen  $T_{i,0} = \sum_{j=0}^{i-1} a_j T_{j,0} + \sum_j f(x_j)$ , wobei die  $x_j$  Stellen sind, an denen  $f$  noch nicht ausgewertet wurde) und die Extrapolation nach jeder Berechnung eines neuen  $T_{i,0}$  durchgeführt und eine Fehlerabschätzung gemacht werden. **Weiterhin dürfen Sie in Ihrem Programm keine zwei- oder höherdimensionale Felder verwenden. Sie sollen aber Unterprogramme (Funktionen) verwenden!**

Bemerkung: In der Praxis wird bei großen Integrationsbereichen, *oszillierenden* Funktionen etc. das Intervall  $[a, b]$  zunächst unterteilt und dann integriert. Ferner wird dieser Unterteilungsvorgang *lokal* durchgeführt, falls nach einer bestimmten Anzahl von Extrapolationen (5–6) die gewünschte Genauigkeit nicht erreicht wird.

## Schnittstellen

Alle benötigten Daten und Funktionen werden Ihrem Programm durch `unit4.h` und `unit4.o` zur Verfügung gestellt, ebenso Unterprogramme zur grafischen Aufbereitung und Kontrolle der errechneten Werte. Im einzelnen enthält die Header-Datei `unit4.h` folgende Schnittstellen:

```
typedef double Vektor    [MaxTiefe+1];
typedef double (*Funktion) ( double x);
```

Die Konstante `MaxTiefe` ist ausreichend groß gewählt, so daß im Programm keine weiteren Feldtypen benötigt. `Funktion` benötigen wir zur Übergabe der Funktion, die integriert werden soll.

```
extern unsigned int AnzahlBeispiele;
```

gibt an, wieviele Beispiel zur Verfügung stehen.

```
enum VerfahrenTyp { Trapezregel, Romberg, Bulirsch };
```

gibt an, welches Verfahren Sie rechnen wollen; Trapezregel mit Romberg–Schrittweitenfolge ohne Extrapolation, bzw. mit Extrapolation zur Romberg– bzw. Bulirsch–Schrittweitenfolge.

```
void Start ( int        beispiel,
              bool       sehr_genau,
              bool       grafik,
              VerfahrenTyp Verfahren,
              Funktion    &f,
              double      &eps,
              double      &a,
              double      &b );
```

muß zu Anfang aufgerufen werden und Sie geben an, welche Beispielfunktion Sie integrieren wollen, ob hohe Genauigkeit erzielt werden soll, mit Grafik gerechnet werden soll und welches **Verfahren** Sie behandeln wollen. Sie bekommen die entsprechende Beispielfunktion **f**, die Genauigkeit bei der — wenn erreicht — der Integrationsprozess stoppen soll sowie die Intervallgrenzen **a** und **b** zurück.

```
void Ausgabe (int  Stufe,  Vektor  Extrapol )
```

wird nach jedem Schritt aufgerufen. **Extrapol** enthält  $T_{Stufe,0}, \dots, T_{Stufe,Stufe}$ . **Stufe** gibt an, wieviele Extrapolationsschritte Sie ausgeführt haben. Wird mit **Verfahren=Trapezregel** gerechnet, so steht in  $Extrapol_0$  der zuletzt berechnete Integralwert, und Sie sollten **Stufe=0** setzen.

```
void Schluss ( double Integralwert,
               double epsilon,
               Vektor extrapol,
               int    Stufe );
```

wird zum Schluß aufgerufen; Sie übergeben Ihre beste Näherung, die geschätzte Genauigkeit, den Extrapolationsvektor und die Stufe (s.o.). Wird mit **Verfahren=Trapezregel** gerechnet, so gilt ebenfalls das zu **Ausgabe** Gesagte.

## Hinweise zum Compilieren und Linken

Zum Erzeugen Ihres ausführbaren Programms benötigen Sie sämtliche Dateien aus dem zur Verfügung gestellten Paket **a3.zip**. Damit die grafische Ausgabe funktioniert, müssen Sie beim Linken zusätzlich die Optionen **-L/usr/lib64 -lX11 -lXi -lXmu -lGL -lGLU -lglut -lpthread** angeben. Diese Optionen bewirken, dass die (im System vorhandenen) Bibliotheken **libpthread**, **libGL**, **libGLU**, **libglut**, **libXmu**, **libX11** und **libXi** eingebunden werden. Die Option **-L/usr/lib64** teilt dazu dem Linker mit, dass er die Bibliotheken auch im Verzeichnis **/usr/lib64** suchen soll. Darüberhinaus soll auch noch die von uns bereitgestellte Bibliothek **libIGL** eingebunden werden. Dazu werden dem Compiler der Suchpfad **-L.** und die Option **-lIGL** mitgegeben (das bedeutet, dass er auch im aktuellen Verzeichnis suchen und die Bibliothek **libIGL** einbinden soll). Zur Vereinfachung des Compile- und Link-Vorganges existiert ein sogenanntes *Makefile*, welches diesen automatisiert. Wenn Sie Ihr Programm **meina3.cpp** nennen, brauchen Sie nur **make** als Kommando einzugeben. Ansonsten können Sie die Textdatei **Makefile** geeignet anpassen.