HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN · FH
MÜNCHEN

Praktikum Ingenieurinformatik

Termin 4

Funktionen,
numerische Integration

Praktikum Ingenieurinformatik

Termin 4

1. **Funktionen**
2. **Numerische Integration, Trapezverfahren**

1.1. Funktionen

Eine **Funktion** ist ein Teil eines Programms der aus (ggf. mehreren) anderen Programmteilen heraus aufgerufen werden kann. Es ist möglich, Werte („**Parameter**“) an Funktionen zu übergeben sowie einen **Rückgabewert** an die aufrufende Stelle zurückzugeben.

Eine Funktion bearbeitet in der Regel eine einzelne, konkrete Aufgabe. Beispielsweise dient die Funktion „**printf**“ zur Ausgabe auf dem Bildschirm und die Funktion „**scanf**“ zur Eingabe von der Tastatur.

Bei der Programmierung von bzw. mit Funktionen unterscheidet man:

- **Funktionsdeklaration** (Hinweis an den Compiler, dass eine bestimmte Funktion existiert und welche Parameter sie erwartet)
- **Funktionsaufruf** (eine Funktion kann von beliebig vielen Stellen im Programm aus aufgerufen werden, vergl. „printf“…)
- **Funktionsdefinition** (Beschreibung des internen Funktionsablaufs)

1.2. Übungsaufgabe, Aufruf einer Funktion

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

```c
#include <stdio.h>
int addiere(int param1, int param2);

int main(void)
{
    int x = 10, y = 20, erg;
    setvbuf(stdout, 0, _IONBF, 0);
    erg = addiere(x, y);
    printf("Funktion 'main':\n");
    printf("erg: %d\n\n", erg);
    return 0;
}

int addiere(int param1, int param2)
{
    int ret;
    ret = param1 + param2;
    printf("Funktion 'addiere'\n");
    printf("param1: %d, param2: %d\n", param1, param2);
    printf("ret: %d\n\n", ret);
    return ret;
}
```

Aufgabe:

Geben Sie den Quelltext in den Rechner ein und lassen Sie ein lauffähiges Programm erzeugen.

Wo finden Sie die Funktionsdeklaration, Aufruf und Funktionsdefinition der Funktion „addiere“?

Aufgabe:

- Ändern Sie in der Funktion „addiere“ die Variablennamen: statt „param1“ und „param2“ jetzt „x“ und „y“. Gibt es Probleme, weil die Namen der Variablen in den Funktionen „main“ und „addiere“ gleich sind?
- Ändern Sie auch „ret“ in „erg“. Gibt es nun Probleme?
- Lassen Sie das Programm im Debugger ablaufen. Können Sie auch die Befehle innerhalb der Funktion „addiere“ schrittweise ausführen? (Tipp: Mit der F5-Taste in die Funktion hinein gehen…)
- Überprüfen Sie den Programmablauf, wenn Sie den Aufruf der Funktion „addiere“ wie ändern: `erg=addiere(123, 10);`
- Fügen Sie die folgende Anweisung in die Funktion „main“ ein: `printf("%d\n", addiere(30, 40));`

Praktikum Ingenieurinformatik

Termin 4

1. Funktionen
2. Numerische Integration, Trapezverfahren

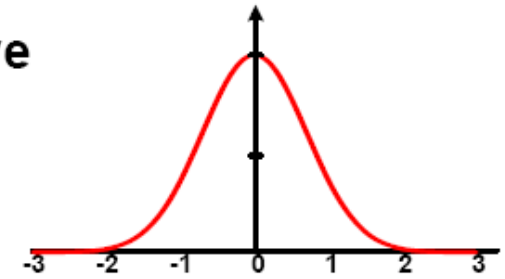
2.1. Numerische Integration, Motivation

Viele Integrale aus der Praxis lassen sich nicht in geschlossener Form mit Hilfe von elementaren Funktionen darstellen.

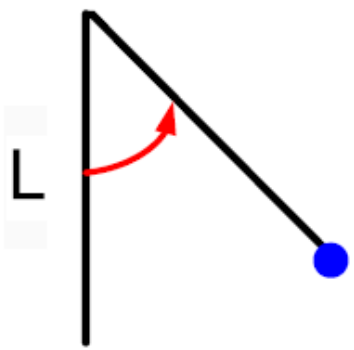
Beispiele :

- **Gauß‘ sche Glockenkurve**

$$\int_a^b e^{-t^2} \cdot dt$$



- **Periodendauer einer Pendelschwingung**



$$T = 4 \cdot \sqrt{\frac{L}{g}} \cdot \int_0^{\pi/2} \frac{d\psi}{\sqrt{1 - k^2 \sin^2 \psi}}$$

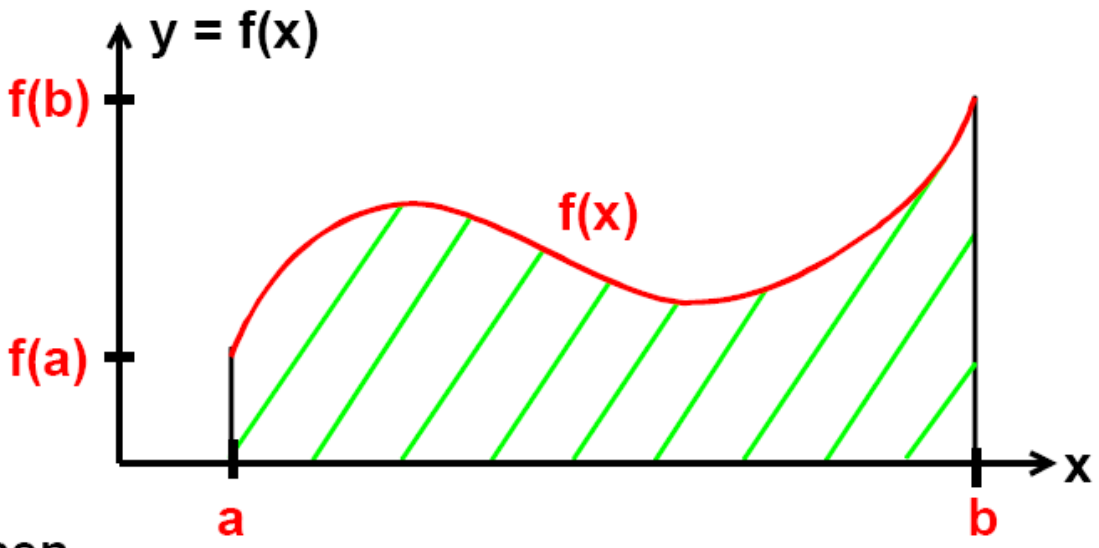
$$\sin\frac{\varphi}{2} = k \cdot \sin\psi \qquad k = \sin\frac{\varphi_0}{2} \qquad \varphi_0 \text{ Anfangsauslenkung}$$

$$k << 1 \quad : \quad T_N = 4 \cdot \sqrt{\frac{L}{g}} \cdot \int_0^{\pi/2} \frac{d\psi}{\sqrt{1}} = 4 \cdot \sqrt{\frac{L}{g}} \cdot \frac{\pi}{2} = 2 \cdot \pi \cdot \sqrt{\frac{L}{g}}$$

2.2. Aufgabenstellung

Problem :

Erstelle ein Programm mit dem man eine beliebige vorgegebene Funktion numerisch integrieren kann.



Gegeben :

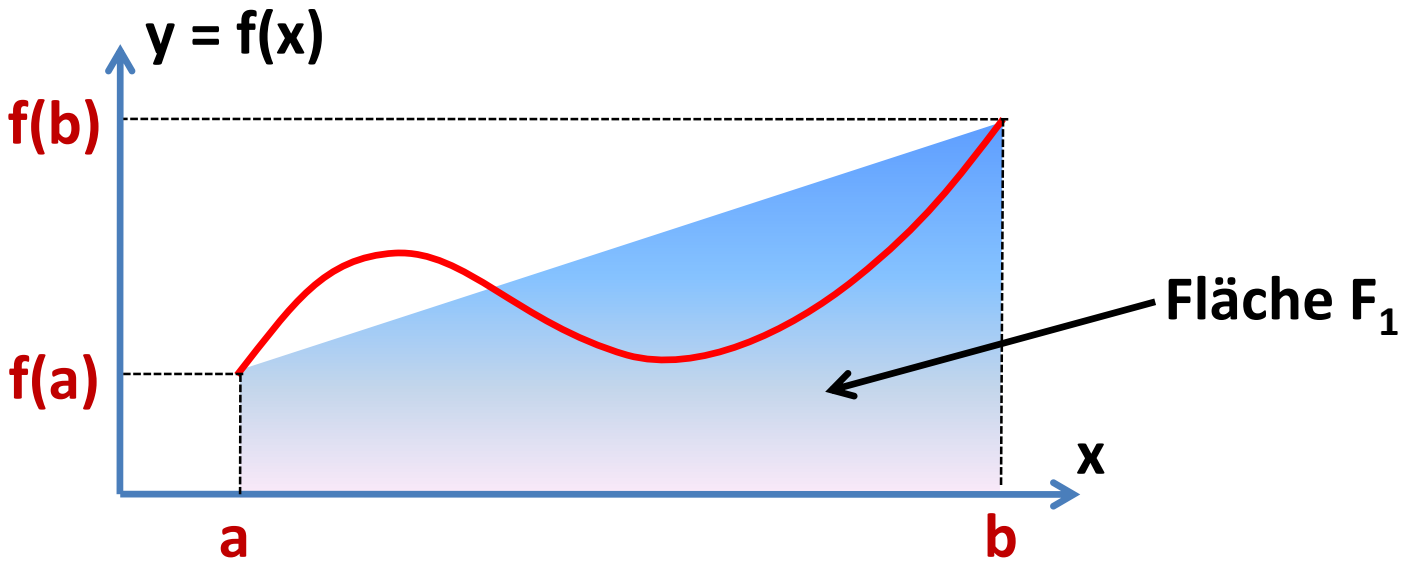
Integrand y = f(x)

Intervallgrenzen a und b einlesen

Anforderungen an das Programm :

- Ändert sich der Integrand, dann soll nur eine Änderung an einer einzigen Stelle im Programm erforderlich sein, um die neue Funktion zu integrieren.
- Die Genauigkeit, mit der das Integral berechnet wird, soll einstellbar sein.
- Es soll überprüft werden, dass das Programm korrekt arbeitet.

Die Fläche F1 ist eine erste Näherung für das gesuchte Integral:

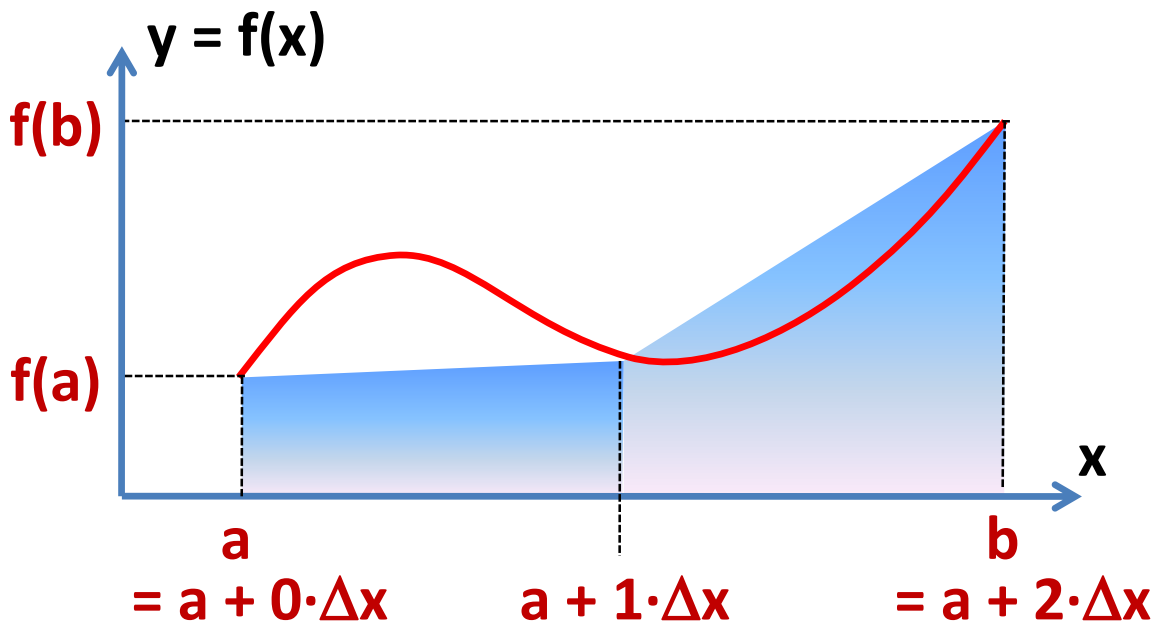


$$F_1 = \frac{f(a)+f(b)}{2} \cdot (b-a)$$

$$F_1 = \frac{f(a+0\cdot\Delta x)+f(a+1\cdot\Delta x)}{2} \cdot \Delta x$$

Breite der Fläche F1: $\Delta x = \frac{b-a}{1}$

Die Fläche F2 ist eine bessere Näherung für das gesuchte Integral:



Breite einer Teilfläche:

$$\Delta x = \frac{b-a}{2}$$

$$F_2 = \frac{f(a+0\cdot\Delta x)+f(a+1\cdot\Delta x)}{2}\cdot\Delta x + \frac{f(a+1\cdot\Delta x)+f(a+2\cdot\Delta x)}{2}\cdot\Delta x$$

Die Fläche Fn ist eine beliebig gute Näherung für das Integral:

Bei n Teilintervallen gilt für die Breite einer Teilfläche:

$$\Delta x = \frac{b-a}{n}$$

Für die Fläche folgt:

$$F_n = \frac{f(a+0\cdot\Delta x)+f(a+1\cdot\Delta x)}{2}\cdot\Delta x + \frac{f(a+1\cdot\Delta x)+f(a+2\cdot\Delta x)}{2}\cdot\Delta x + \ldots + \frac{f(a+(n-1)\cdot\Delta x)+f(a+n\cdot\Delta x)}{2}\cdot\Delta x$$

2.6. Übungsaufgabe, Trapezverfahren (a)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Aufgabe:

Implementieren Sie die folgende Funktion:

```
double trapez(double a, double b, int n);
```

Die Funktion „trapez“ integriert eine Funktion f(x) nach dem Trapezverfahren im Bereich von a bis b. Es werden insgesamt n Teilflächen berechnet und aufsummiert.

Tipp: Während der Berechnung der Gesamtfläche muss der Funktionswert f(x) an verschiedenen Stellen x berechnet werden (siehe vorherige Folie). Dazu ist im C-Programm folgende Funktion definiert:

```
double f(double x);
```

Der x-Wert wird als Parameter übergeben, der Wert y = f(x) wird als Rückgabewert zurückgegeben.

2.7. Übungsaufgabe, Trapezverfahren (b)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

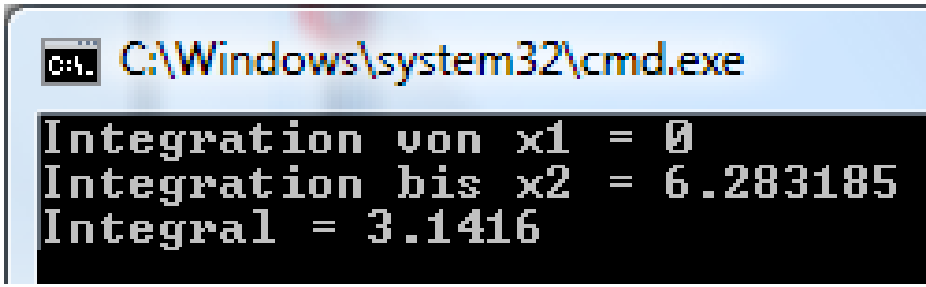
Beispiel für ein Hauptprogramm:

```
#include <stdio.h>
#include <math.h>

double trapez(double a, double b, int n);
double f(double x);

int main(void)
{
    double x1, x2, flaeche;
    setvbuf(stdout, 0, _IONBF, 0);
    printf("Integration von x1 = "); scanf("%lf", &x1);
    printf("Integration bis x2 = "); scanf("%lf", &x2);
    flaeche = trapez(x1, x2, 1000);
    printf("Integral = %.4f\n", flaeche);
    return 0;
}

double f(double x)
{
    return sin(x) * sin(x);
}
```



2.8. Übungsaufgabe, Trapezverfahren (c)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Aufgabe:

Erweitern Sie das numerische Integrationsprogramm, so dass die Anzahl der Teilflächen (n) nicht mehr fest eingestellt ist. Stattdessen soll die Anzahl solange immer weiter verdoppelt werden, bis die Ergebnisse der Funktion trapez „genau genug“ sind.

Erstellen Sie dazu eine zusätzliche Funktion „integral“ (siehe folgende Seite), welche die bereits bestehende Funktion „trapez“ aufruft.

Vergleichen Sie die Ergebnisse für n und (2·n) Teilflächen und brechen Sie das Verfahren ab, wenn der relative Unterschied beider Ergebnisse eine vorgegebene Grenze „EPSILON“ unterschreitet.

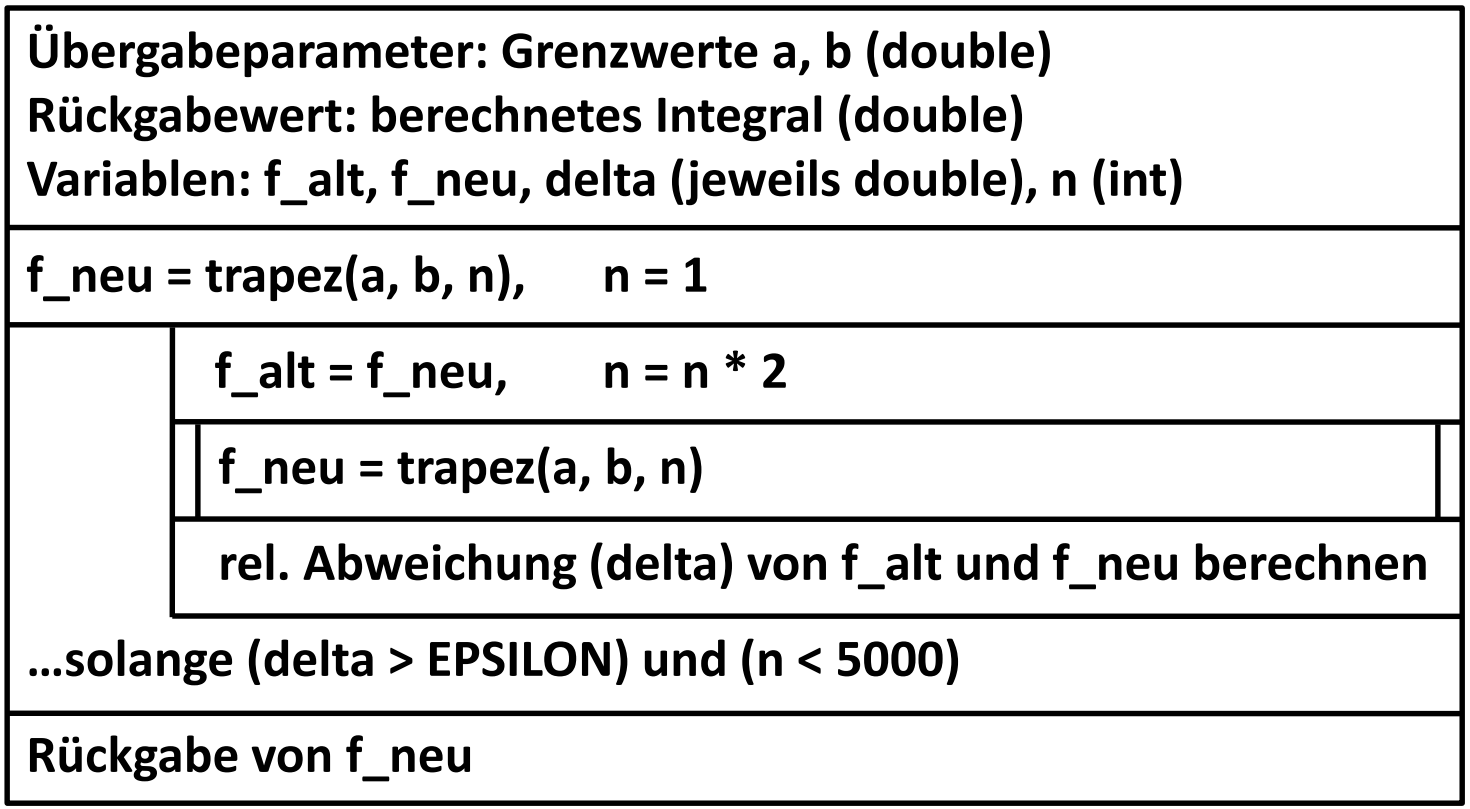
Wenn die Anzahl der Teilflächen (n) größer als 5000 ist, soll das Verfahren ebenfalls beendet werden.

2.9. Übungsaufgabe, Trapezverfahren (d)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Struktogramm der Funktion „integral“:

Diese Funktion ruft solange die Funktion „trapez“ mit einer wachsenden Anzahl von Teilflächen auf, bis das Ergebnis eine vorgegebene Genauigkeit erreicht hat.



2.10. Übungsaufgabe, Trapezverfahren (e)

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Aufgabe:

- Testen Sie das Programm zur numerischen Integration für verschiedene Funktionen f(x) und verschiedene Grenzen a und b.
- Vergleichen Sie die vom Programm ermittelten Ergebnisse mit Werten, die Sie entweder von Hand oder mit Hilfe von MATLAB und/oder Maple berechnet haben.
- Implementieren Sie andere Funktionen f(x) unter Nutzung von Standardfunktionen, die Sie in der Bibliothek „math.h“ finden.