

---

# Praktikum Ingenieurinformatik

**Termin 4**

**Funktionen,  
numerische Integration**

---

# Praktikum Ingenieurinformatik

## Termin 4

- 1. Programmieren von Funktionen**
- 2. Numerische Integration, Trapezverfahren**
- 3. Zusatzaufgabe**

# 1.1. Programmieren von 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. Programmieren von Funktionen

```
#include <stdio.h>
double rechteck(double a, double b);

int main(void)
{
    double s1, s2, erg;
    printf("Seite 1: "); scanf("%lf", &s1);
    printf("Seite 2: "); scanf("%lf", &s2);

    erg = rechteck(s1, s2);
    printf("Umfang: %.2f\n", erg);
    return 0;
}

double rechteck(double a, double b)
{
    double umfang;
    umfang = 2 * a + 2 * b;
    return umfang;
}
```

### Aufgabe:

Geben Sie den Quelltext in den Rechner ein und starten Sie das Programm.

Wo finden Sie die Deklaration (Prototyp), den Aufruf und die Definition der Funktion „rechteck“?

## 1.3. Programmieren von Funktionen

### Aufgabe:

- Ändern Sie in der Funktion „rechteck“ die Variablennamen: Gibt es Probleme, wenn in der Funktion dieselben Variablennamen wie im Hauptprogramm verwendet werden?
- Lassen Sie das Programm im Debugger ablaufen. Können Sie auch die Befehle innerhalb der Funktion „rechteck“ schrittweise ausführen? (Tipp: Versuchen Sie, innerhalb der Funktion „rechteck“ einen Haltepunkt/Breakpoint zu setzen. Sie können aber auch den Debugger anweisen, in eine Unterfunktion hineinzugehen.)
- Kann man die Funktion „rechteck“ auch mit festen Zahlen aufrufen?  
Zum Beispiel so: `erg = rechteck(1.5, 2.5);`  
Oder auch so: `printf("Umfang: %f\n", rechteck(1.5, 0.5));`

---

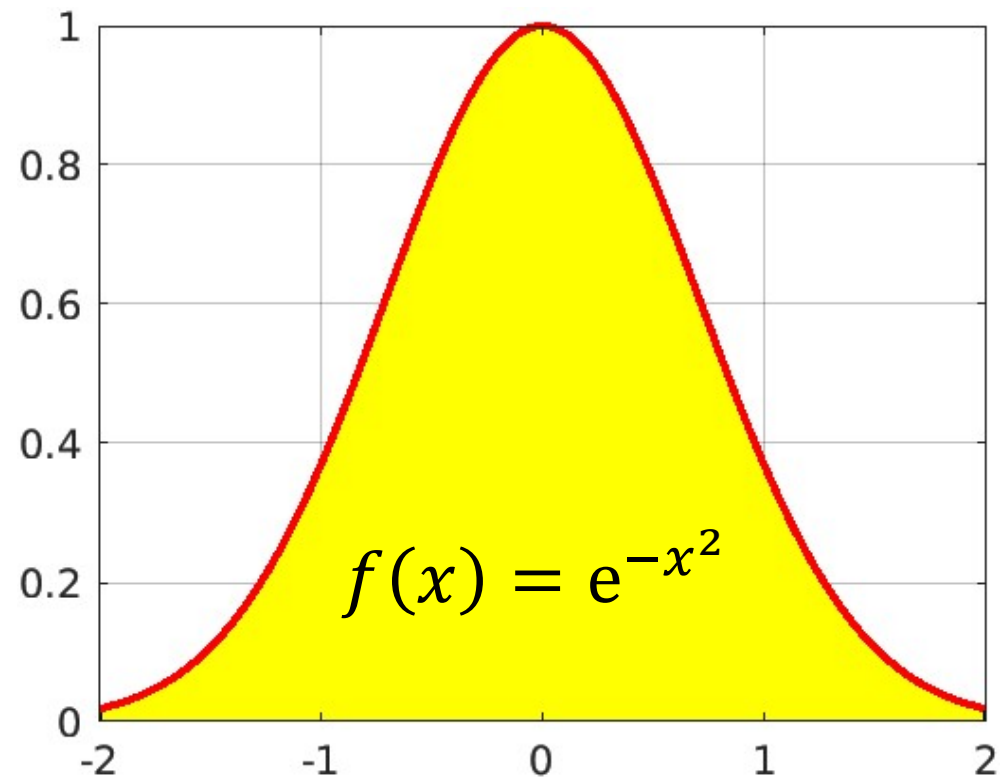
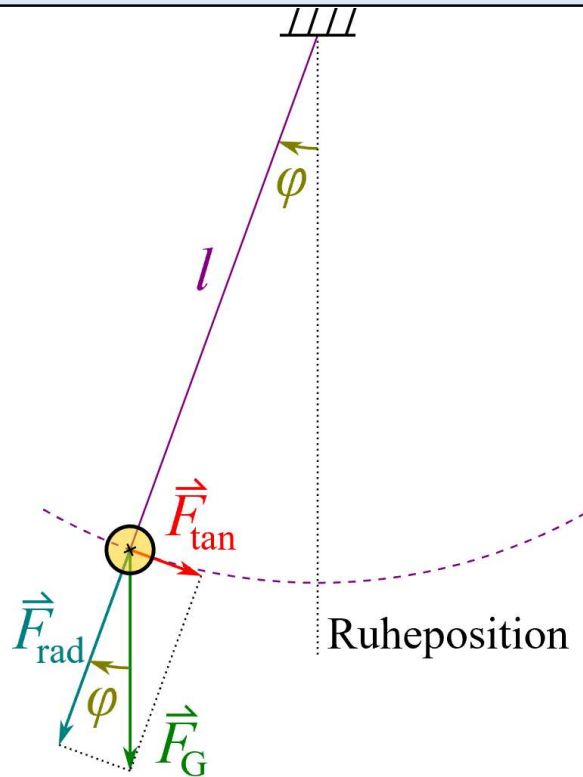
# Praktikum Ingenieurinformatik

## Termin 4

1. Programmieren von Funktionen
2. Numerische Integration, Trapezverfahren
3. Zusatzaufgabe

## 2.1. Numerische Integration, Trapezverfahren

Integrale lassen sich oft nicht in geschlossener Form mithilfe von elementaren Funktionen darstellen. Beispiele sind die Berechnung der Periodendauer eines Pendels oder die Gauß-Funktion. Solche Integrale können nur numerisch berechnet werden.



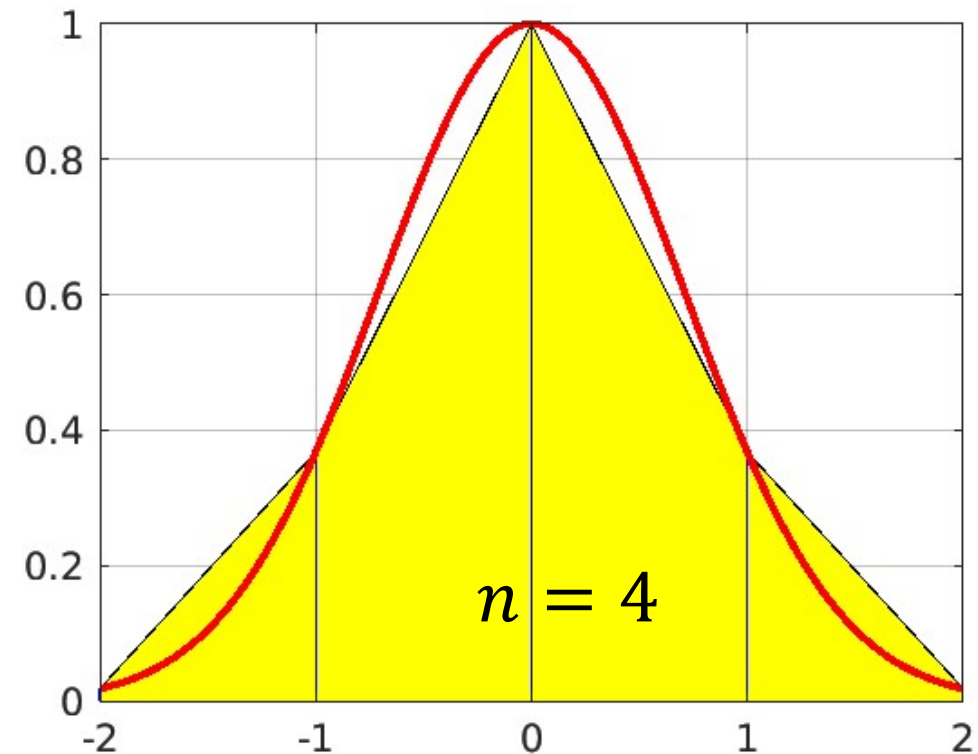
Quelle: Wikimedia,  
Kraefte\_am\_Fadenpendel\_groß.svg

## 2.2. Numerische Integration, Trapezverfahren

### Aufgabe:

Erstellen Sie ein Programm zum numerischen Integrieren von beliebigen Funktionen  $f(x)$  mit dem Trapezverfahren.

- Die Funktion  $f(x)$  ist im C-Programm definiert.
- Die Integrationsgrenzen  $a, b$  werden vom Anwender eingegeben.
- Die gewünschte Trapez-Anzahl  $n$  wird ebenfalls eingegeben.





## 2.3. Numerische Integration, Trapezverfahren

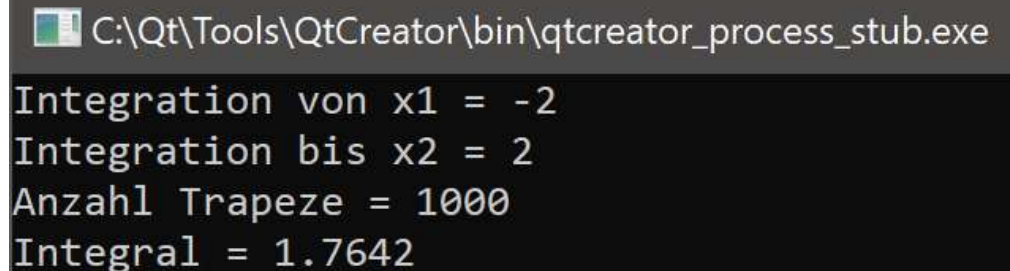
```
#include <stdio.h>
#include <math.h>

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

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

double f(double x)
{
    return exp(-x * x);
}
```

Es fehlt nur noch die Definition der Funktion „trapez“...



```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Integration von x1 = -2
Integration bis x2 = 2
Anzahl Trapeze = 1000
Integral = 1.7642
```

## 2.4. Numerische Integration, Trapezverfahren

### Struktogramm der Funktion „trapez“:

Parameter:	a, b (double, Integrationsgrenzen)
	n (int, Anzahl der Trapeze)
Lokale Variablen:	i (Schleifenzähler)
	flaeche, dx, neu (Berechnung der Trapeze)

Variablen initialisieren:  
flaeche = 0; dx = (b – a) / n

Für alle i von 0 bis einschließlich (n – 1)...

$$\text{neu} = (\text{dx} / 2) * (f(a + i * \text{dx}) + f(a + (i + 1) * \text{dx}))$$

Die gerade berechnete Trapezfläche „neu“ zur Gesamtfläche „flaeche“ hinzuaddieren.

Die Summe aller Trapezflächen (= das Ergebnis der numerischen Integration) steht nun in der Variablen „flaeche“. Dieses Ergebnis wird an das Hauptprogramm zurückgegeben.

### Weitere Aufgaben:

- Wie lautet die Funktion  $f(x)$ , die hier integriert wird?
- Vergleichen Sie die Ergebnisse Ihres Programms mit den Ergebnissen, die von Wolfram Alpha berechnet werden:  
<https://www.wolframalpha.com/examples/mathematics/calculus-and-analysis/>
- Wie viele Trapeze sind erforderlich, damit das Integral in den Grenzen von -2 bis +2 bis auf eine Nachkommastelle genau berechnet wird?
- Wie oft (und woher?) werden die Funktionen „main“, „trapez“ und „f“ während eines Programmablaufs jeweils aufgerufen?
- Ersetzen Sie die Funktion  $f(x)$  durch eine andere Funktion, zum Beispiel  $f(x) = x^2$  und vergleichen Sie wiederum die Ergebnisse Ihres Programms mit denen von Wolfram Alpha.

---

# Praktikum Ingenieurinformatik

## Termin 4

1. Programmieren von Funktionen
2. Numerische Integration, Trapezverfahren
3. Zusatzaufgabe

## 3.1. Zusatzaufgabe

### Zusatzaufgabe:

Die Funktion „trapez“ hat einen großen Nachteil: Beim Aufruf dieser Funktion muss die konkrete Anzahl der zu berechnenden Trapeze angegeben werden.

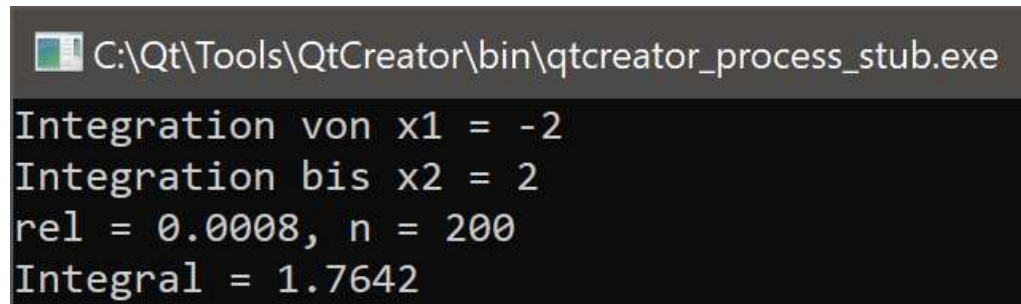
Viel schöner wäre es, wenn die Integrations-Funktion selbst ermitteln könnte, wie viele Trapeze für eine bestimmte Genauigkeit erforderlich sind.

Schreiben Sie daher eine zusätzliche Funktion „integral“, welche mithilfe einer Schleife die bereits bestehende Funktion „trapez“ zunächst mit  $n = 25$  aufruft, dann mit  $n = 50$ , dann mit  $n = 100$  usw. Die Schleife soll beendet werden, wenn die relative Abweichung der letzten beiden Berechnungen kleiner als 0,001% ist oder wenn die Anzahl  $n$  größer als 1.000.000 geworden ist.

## 3.2. Zusatzaufgabe

Beispiel für das neue Hauptprogramm: Beachten Sie den Aufruf der neuen Funktion „integral“ (ohne Angabe der Trapez-Anzahl).  
Tipp: Lassen Sie sich zum Test auch die erreichte relative Abweichung sowie die Anzahl der Trapeze ausgeben!

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



```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Integration von x1 = -2
Integration bis x2 = 2
rel = 0.0008, n = 200
Integral = 1.7642
```

### 3.3. Zusatzaufgabe

#### Weitere Aufgaben:

- Suchen Sie im Internet weitere Informationen über das heute programmierte Trapezverfahren, zum Beispiel bei Wikipedia:
- <https://de.wikipedia.org/wiki/Trapezregel>
- Falls Sie bereits den zweiten Teil des Ingenieurinformatik-Moduls besuchen oder besucht haben („Numerik für Ingenieure“), dann programmieren Sie die heute erstellten Programme zur Übung auch einmal in MATLAB.