7 Numerische Integration

Die Aufgabe der Integralrechnung ist die Bestimmung des Integralwertes I für eine im Intervall $[a,b]$ integrierbare Funktion f:

$$I = \int_a^b f(x)dx \qquad f:[a,b] \to \mathrm{R} \tag{7.1}$$

Für eine stetige Funktion f deren Stammfunktion (engl. antiderivative) F mit $F'=f$ bekannt ist, gilt nach dem Hauptsatz der Integralrechnung:

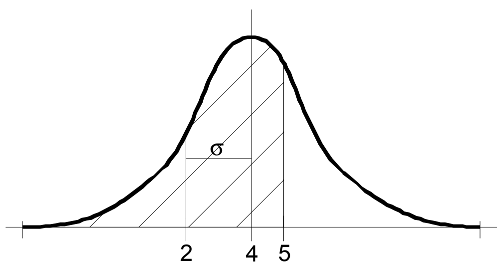
$$\int_a^b f(x)\,dx = F(b) - F(a) \tag{7.2}$$

Hauptsatz der Integralrechnung

In der Praxis ist die Stammfunktion nur schwer oder überhaupt nicht zu bestimmen, so dass keine geschlossene Lösung für das Integral bekannt ist (was nicht heisst, dass sie nicht existiert).

Beispiel:
Die Verteilungsfunktion für die Normalverteilung:

$$P(X \le t) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{t} e^{\frac{-(X-\mu)^2}{2\sigma^2}} dt$$



Mit einer numerischen Integration können wir nun ohne Tabellen direkt das Integral berechnen und erhalten so beispielsweise für $\mu = 4, \sigma = 2$ eine Fläche für $P(t \le 5) = 0.691462\ldots$

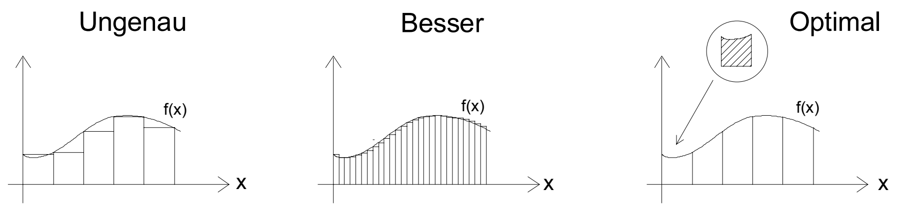
Es liegt auf der Hand, **dass ausschliesslich bestimmte Integrale** mit numerischen Methoden berechnet werden können.

Zur Rechnung sind verschiedene Verfahren bekannt. Sie arbeiten alle nach dem Prinzip, dass das Integral durch eine **Summe endlicher Teilflächen** angenähert wird. Die Verfahren unterscheiden sich im Wesentlichen dadurch wie die Approximationen der einzelnen aufzusummierenden Teilflächen erfolgt (Rechteck, Trapez, etc.).

7.1 Grundlegende Prinzipien zur numerischen Infinitesimalrechnung

Das wesentliches Prinzip bei der numerischen ist das Annähern des infinitesimal kleinen Differenzialquotienten durch einen endlich grossen Differenzenquotienten. Somit sind also diese Berechnungen grundsätzlich immer Näherungen.

Bei der numerischen Integration erfolgt nach einer ähnlichen Idee eine Näherung des bestimmten Integrals durch eine endliche Summation von Teilflächen. Durch geschickte Wahl der Form und/ oder Anzahl der Teilflächen kann der verbleibende Fehler beliebig klein gemacht werden:



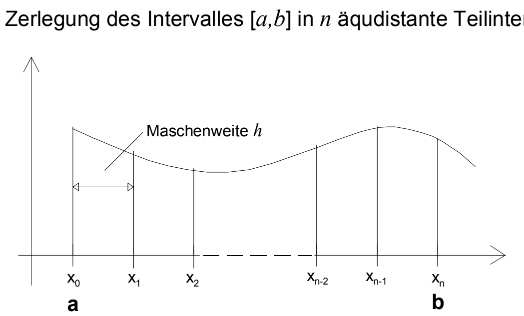
7.1.1 Intervallzerlegung

Jedes numerische Integrationsverfahren arbeitet mit einer Zerlegung des zu integrierenden Intervalls [a,b].
Wir gehen davon aus, dass $f:[a,b]\to \mathrm{R}]$ im ganzen Intervall integrierbar ist und zerlegen das Intervall in n äquidistante Teilstücke mit der Weite h:

$$h=\frac{b-a}{n}$$ **Äquidistante Zerlegung mit Maschenweite *h*** (7.3)

woraus sich n+1 Stützstellen (Teilpunkte) ergeben: $x_i = a + i \cdot h \qquad i = 0,\ldots,n$ (7.4)

Wenden wir diese Zerlegung für eine beliebige in $[a,b]$ integrierbare Funktion f an, so erhalten wir beispielsweise ein Bild:



Beispiel:
Bestimmen Sie die äquidistante Zerlegung für n=6 und zeigen Sie alle Teilintervalle!

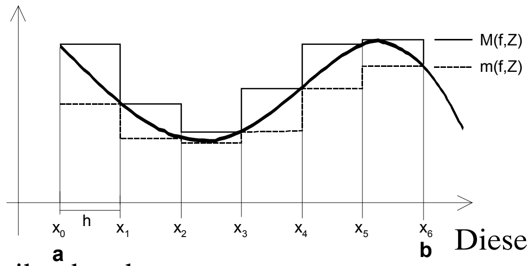
7.1.2 Rechtecksummen

Obwohl die Ideen des Riemann-Integrals nicht direkt auf numerische Integrationsmethoden übertragen werden können, beinhalten sie die wesentlichen Gedanken zur Formulierung von numerischen Verfahren zur Intergation. Häufig werden die Rieman-Summen benutzt um grundsätzlich die Integrierbarkeit einer Funktion zu beweisen.

Aus der geometrischen Interpretation des bestimmten Integrals wissen wir, dass es die Fläche des abgeschlossenen Intervalls zwischen dem Graphen der Funktion *f(x)* und der *x*-Achse zeigt.
Diese Fläche kann auf zwei Arten beschrieben werden: Obere Rechtecksummen von *f* und untere Rechtecksummen von *f*. Diese verkörpern beide näherungsweise die Fläche unter dem Graphen.

Wir definieren zur Rechnung einige Bezeichnungen.
Sei $Z = x_0, \ldots, {}_{xn}$ eine Zerlegung (sog. Partition) des Intervalles $[a,b]$. Dann ist

$$\begin{aligned} m_k &= \inf\{f(x): x_{k-1} \le x \le x_k\} \\ M_k &= \sup\{f(x): x_{k-1} \le x \le x_k\} \end{aligned} \qquad (7.5)$$



Diese Mengen verkörpern im Prinzip die Höhen der einzelnen Teilrechtecke.

Die entsprechenden Unter- und Obersummen von *f* heissen dann:

$$\begin{aligned} U(f,Z) &:= \sum_{i=0}^{n-1} m_i (x_{i+1} - x_i) && \text{Untersumme von } f \text{ für die Zerlegung } Z \\ O(f,Z) &:= \sum_{i=0}^{n-1} M_i (x_{i+1} - x_i) && \text{Obersumme von } f \text{ für die Zerlegung } Z \end{aligned} \qquad (7.6)$$

Beispiel:
Wie lauten die numerischen Werte für Unter- und Obersummen für die Funktion $f(x) = x^2$ im Intervall [0,1] mit einer Zerlegung mit n=4:

$$h = \frac{b-a}{n} = \frac{1-0}{4} = \frac{1}{4}$$

$$Z = \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$$

$$m_k = \inf\left\{x^2 : x_k \le x \le x_{k+1}\right\} = x_k^2$$

$$M_k = \sup\left\{x^2 : x_k \le x \le x_{k+1}\right\} = x_{k+1}^2$$

$$U(f,Z) = \sum_{i=0}^{3} m_i\left(x_{i+1} - x_i\right) = \sum_{i=0}^{3} x_k^2\left(x_{i+1} - x_i\right) = (0 + \frac{1}{16} + \frac{1}{4} + \frac{9}{16})\frac{1}{4} = \frac{7}{32}$$

$$O(f,Z) = \sum_{i=0}^{3} M_i\left(x_{i+1} - x_i\right) = \sum_{i=0}^{3} x_{k+1}^2\left(x_{i+1} - x_i\right) = (\frac{1}{16} + \frac{1}{4} + \frac{9}{16} + 1)\frac{1}{4} = \frac{15}{32}$$

Dieses Beispiel berechnet eine Näherung für das Integral $\int_0^1 x^2 dx$. Der exakte Wert liegt jedoch zwischen den beiden Summen, hier etwa in der Mitte der beiden Werte.

Der exakte Wert liegt zwischen den beiden Summen. Er erfüllt die folgende Ungleichung für jede beliebige Zerlegung Z:

$$U(f,Z) \le \int_a^b f(x)dx \le O(f,Z) \tag{7.7}$$

Ist die Funktion integrierbar und wählen wir die Zerlegung Z genügend fein, so sehen wir, dass beide Summen zueinander konvergieren.

Eine Funktion f heisst integrierbar (sog. Riemann-integrierbar) über $[a.b]$, wenn für eine Zerlegung Z gilt:

$$\sup U(f.Z) = \inf O(f,Z). \tag{7.8}$$

Dann heisst die Zahl "Integral von f über $[a.b]$" und wird geschrieben als :

$$\int_a^b f(x)dx = \sup U(f,Z) = \inf O(f,Z) \tag{7.9}$$

Riemann-Integrierbarkeitskriterium

Beispiel für die Herleitung eines bestimmten Integrales mit Unter- und Obersummen: $I = \int_a^b x\, dx$

$$h = \frac{b-a}{n}$$

$$m_k = \inf\{x : x_k \le x \le x_{k+1}\} = x_k$$

$$M_k = \sup\{x : x_k \le x \le x_{k+1}\} = x_{k+1}$$

$$U(f,Z) = \sum_{i=0}^{n-1} m_i(x_{i+1} - x_i) = \sum_{i=0}^{n-1} x_i(x_{i+1} - x_i) = \sum_{i=0}^{n-1} x_i\, h = \sum_{i=0}^{n-1}(a + i\,h)\,h = \frac{a(b-a)}{n}\sum_{i=0}^{n-1} 1 + \frac{(b-a)^2}{n^2}\sum_{i=0}^{n-1} i =$$

$$= \frac{a(b-a)}{n}n + \frac{(b-a)^2}{n^2}\frac{n(n-1)}{2} = a(b-a) + \frac{(b-a)^2(n-1)}{2n} = \underline{\frac{(b-a)(a-b+an+bn)}{2n}}$$

$$O(f,Z) = \sum_{i=0}^{n-1} M_i(x_{i+1} - x_i) = \sum_{i=0}^{n-1} x_{k+1}(x_{i++} - x_i) = \sum_{i=0}^{n-1} x_{i+1}h = \sum_{i=0}^{n-1}(a + (i+1)h)h = \frac{a(b-a)}{n}\sum_{i=0}^{n-1} 1 + \frac{(b-a)^2}{n^2}\sum_{i=0}^{n-1}(i+1) =$$

$$= \frac{a(b-a)}{n}n + \frac{(b-a)^2}{n^2}\frac{n(n+1)}{2} = a(b-a) + \frac{(b-a)^2(n+1)}{2n} = \underline{\frac{(b-a)(b-a+an+bn)}{2n}}$$

$$\lim_{n\to\infty} U(f,Z) = \lim_{n\to\infty}\frac{(b-a)(a-b+an+bn)}{2n} = \frac{b^2-a^2}{2}$$

$$\lim_{n\to\infty} O(f,Z) = \lim_{n\to\infty}\frac{(b-a)(b-a+an+bn)}{2n} = \frac{b^2-a^2}{2}$$

$$\lim_{n\to\infty} U(f,Z) = \lim_{n\to\infty} O(f,Z) \Longrightarrow \int_b^b x\, dx = \underline{\underline{\frac{b^2-a^2}{2}}}$$

Wir sehen, dass es bei der Existenz des Integrals genügt eine Summe zu berechnen. Diesen Umstand werden wir auch in der Praxis nutzen, so dass immer nur eine Summe berechnet wird.

7.1.3 Integrierbarkeit von Funktionen

Aus dem Riemann-Integrierbarkeitskriterium können wesentliche Aussagen für die Integrierbarkeit einer (reellen) Funktion gewonnen werden:

1. Jede stückweise stetige Funktion die höchstens endlich viele Unstetigkeitstellen hat, ist Riemann-integrierbar.
2. Jede stetige, in einem abgeschlossenen Intervall definierte Funktion ist Riemann-integrierbar.
3. Jede im Intervall $[a,b]$ zu integrierende Funktion muss beschränkt sein.

Es existieren viele Funktionen die nach obiger Definition nicht Riemann-integrierbar sind. Das einfachste Beispiel ist die Dirichlet-Funktion:

$$d(x) := \begin{cases} 0 & x \text{ ist rational} \\ 1 & x \text{ ist irrational} \end{cases} \qquad \textbf{Dirichlet-Funktion} \tag{7.10}$$

Wir erhalten hier für jede Zerlegung Z im Intervall $[a,b]$ die Summen $U(d,Z)=0$ und $O(d,Z)=b-a$. Somit wird

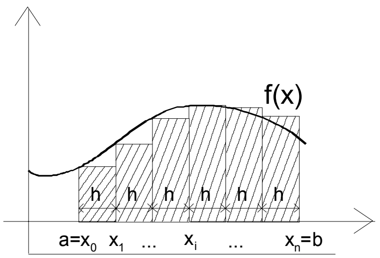
$$0 = \sup_Z U(d,Z) < \inf_Z O(d,Z) = b-a$$

und wir sehen, dass das Integrierbarkeitskriterium für keine Zerlegung Z erfüllt ist.

7.2 Rechteckverfahren

Das Rechteckverfahren ist das einfachste numerische Integrationverfahren. Es beruht darauf, dass die einzelnen zu summierenden Teilflächen durch eine endliche Anzahl Rechtecke dargestellt werden.

Da nach Voraussetzung die Funktion integrierbar ist, spielt es bei einer genügend feinen Zerlegung Z (= Anzahl Teilflächen) keine Rolle, ob der linke oder rechte Funktionswert für die Flächenberechnung benutzt wird. Im Falle der Existenz des Integrals, die ja vorausgesetzt wird, konvergieren beide Summen zum gleichen Grenzwert.



Näherung des Integralwertes durch aufsummieren von Teilrechteckflächen beim Rechteckverfahren.

Wir können das Integral mit einer genügend feinen Zerlegung durch eine Summe der Rechteckflächen nähern und erhalten eine Rechtecksumme der Ordnung n: $R(n) := h\sum_{i=0}^{n-1} f(x_i)$

Wir können mit diesem Ansatz das Integral beliebig genau berechnen, sofern wir n genügend gross wählen, da nach (7.8) gilt: (7.11)

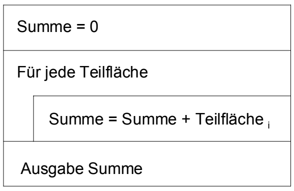
$$\lim_{n\to\infty} R(n) = \lim_{n\to\infty} U(f,Z) \overset{\text{f ist R-integrierbar}}{=} \int_a^b f(x)dx$$

7.2.1 Entwurf und Implementierung des Rechteckverfahrens

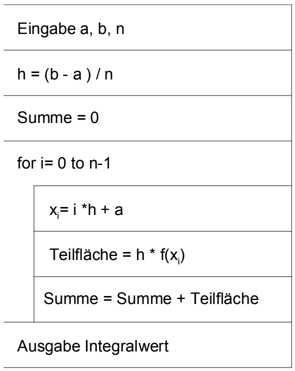
Wir entwerfen ein einfaches Programm zur Berechnung des bestimmten Integrals einer Funktion $f(x) := e^x$ nach dem Rechteckverfahren. Die zu integrierende Funktion wird als eigenständig definierte

C-Funktion mit dem Namen `f` implementiert:

Grobentwurf:



Verfeinerung:



Die Verfeinerung des Entwurfes lässt sich direkt implementieren. Wir benötigen jetzt nur noch die Definition der Funktion selbst, sowie die zweckmässige Wahl der Datentypen im Programm.

Da *f(x)* eine Abbildung von $[a,b] \rightarrow \mathrm{R}$ ist, müssen wir mit Gleitkommazahlen rechnen. Wir wählen als Grunddatentyp für die Summen und Argumente den Typ `double`. Die Schleifen und Zählvariablen sind ganzzahlig und wir verwenden dafür `int`'s.

Die Funktion wird als C-Funktion definiert die ein `double` Argument erhält und ein `double` Resultat zurückgibt. (Dies ist für diesen Fall sicher die vernünftigste Definition).

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

Eine einfache Implementierung in C ist nachfolgend gezeigt.

```c
/* Programm Rechteck Integral                                        (File: RECT_INT.C )
   Numerische Integration einer Funktion f nach dem Rechteckverfahren.

   Arbeitsweise:
   Aufsummieren von n Teilflaechen, die durch (linke) Rechteckelemente repraesentiert
   werden. Die resultierende Summe stellt eine Naeherung des Integrals dar.
   Voraussetzung:
   Es wird vorausgesetzt, dass die Funktion im zu integrierenden Intervall
   integrierbar ist.

   Autor: Gerhard Krucker
   Datum: 3.5.1995
   Sprache: MS Visual-C V1.5 (QuickWin Applikation)
 */

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

/* Definition der Funktion f, die ueber das Intervall [a,b] integriert werden
   soll. Sie liefert als Resultat den Funktionswert zurueck.
   Hier wird fuer das Testbeispiel die Funktion e^x implementiert.
*/
double f (double x)
{
  return exp(x);
}

int main()
{  int n;             /* Anzahl Teilintervalle (Teilflaechen, Zerlegung) */
   double a,b;        /* Integrationsgrenzen */
   double int_summe;  /* Summe der einzelnen Teilflaechen */
   double h;          /* Intervallbreite */
   int i;             /* Schleifenvariable */

   printf("Integration nach der Rechteckmethode\n");
   printf("Eingabe der Integrationsgrenzen a  b: ");
   scanf("%lf %lf",&a,&b);
   printf("Eingabe der Anzahl Teilintervalle n: ");
   scanf("%d",&n);

   int_summe=0;
   h = (b - a) / n;

   for (i = 0; i <= n-1; i++)
     { double xi;        /* Argument im Schritt i */
       double teilfl;    /* Teilflaeche */

       xi = i * h + a;
       teilfl = f(xi) * h;
       int_summe += teilfl;
      }

    printf("Integral: %10.8g\n",int_summe);

    return 0;
 }
```

7.2.2 Testbeispiel

Wie bei anderen numerischen Methoden testen wir das Verfahren mit einer einfachen Testaufgabe, die wir ohne grossen Aufwand von Hand rechnen können. Wir verwenden hierfür das Testintegral:

$$\int_0^1 e^x dx = e^x\Big|_0^1 = e - 1 \approx 1.718281828\ldots$$

Wir können damit die Güte des Verfahrens für unterschiedliche n anhand des exakten Vergleichswertes beurteilen.

Wird das Programm genau nach dem Entwurf implementiert, so erhalten wir folgende Werte:

n	*R(n)*
10	1.6338
100	1.7097
1000	1.71742
10000	1.7182

Da unser Programm mit den Untersummen rechnet, erhalten wir für die Funktion e^x generell zu kleine Resultate, da die Untersumme zum Integralwert nach oben konvergiert.

Wir sehen, dass diese Methode selbst für eine so einfache Funktion (glatt, monoton steigend, etc.) sehr langsam den Integralwert approximiert. Man braucht hier sehr grosse n um eine halbwegs brauchbare Näherung zu erreichen. Selbst für n=1000 erhalten wir nur eine Genauigkeit von 3 signifikanten Stellen.

7.2.3 Fehlerabschätzung des Rechteckverfahrens

Eine Fehlerabschätzung lässt sich für eine stetig differenzierbare Funktion f nach [Kausen, Numerische Mathematik, S.121], wie folgt vornehmen:

$$\left|\int f - R(n)\right| \le c\frac{(b-a)^2}{n}\max|f'| \qquad (c \in [0,1)) \tag{7.12}$$

Wobei das Maximum von $|f'|$ auf dem Integrationsintervall $[a,b]$ zu nehmen ist. Wesentlich ist an dieser Aussage, dass der Fehler nur proportional zu n abnimmt, oder anders ausgedrückt: Um die Genauigkeit um eine Dezimalstelle zu verbessern muss n verzehnfacht werden.

Eine Betrachtung, wie eine solche Abschätzung zustandekommt wird im Kapitel "Trapezregel" gezeigt.

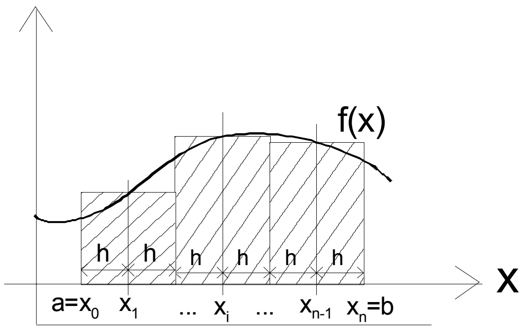
7.2.4 Zusammenfassende Ergebnisse zum Rechteckverfahren

Die Näherung der aufzusummierenden Teilflächen erfolgt durch Rechtecke. In der Regel birgt diese Form der Näherung einen recht grossen Fehler.

Dadurch ist das Rechteckverfahren für die Integralberechnung kaum praktisch zu gebrauchen, da die notwendige Anzahl Teilintervalle für eine brauchbare Genauigkeit viel zu gross ist. Dennoch zeigt dieses Verfahren anschaulich die Grundidee, nach der fast alle numerischen Integrationsverfahren arbeiten:
Berechnung des Integrals als Summe der Teilflächen über (meist) äquidistante Stützstellen.

7.3 MacLaurin Verfahren

Das Rechteckverfahren kann mit minimalem Aufwand erheblich verbessert werden, indem man nicht die linke oder rechte Summe aufsummiert, sondern Rechtecke, die in die Mitte des jeweiligen Teilintervalles gelegt werden.



Dazu zerlegt man das Intervall $[a,b]$ in n Teilintervalle wobei n eine gerade Zahl sein muss.

Zu jedem ungeraden Index $\{x_1, x_3, \ldots x_{n-1}\}$ bilden wir das Rechteck, das aus dem Intervall $[x_{i-1}, x_{i+1}]$ und der Höhe $f(x_i)$ mit der Fläche

$$M_i = 2h\,f(x_i) \tag{7.13}$$

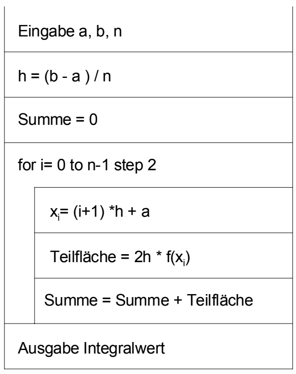
Die MacLaurin-Summen der Ordnung n erhalten wir:

$$M(n) := 2h \sum_{i=1,3,5..,n-1} f(x_i) \tag{7.14}$$

Wie bei den Rechtecksummen, verkörpert die MacLaurin-Summe bei einer unendlich feinen Zerlegung ($n \to \infty$) exakten Integralwert.

7.3.1 Entwurf und Implementierung des MacLaurin-Verfahrens

Wir entwerfen den Algorithmus analog dem Rechteckverfahren, so dass hier nur noch die verfeinerte Version gezeigt wird. Alle dort gemachten Betrachtungen und Gedanken zu den Daten und Ablauf gelten aber auch hier.



Konkret könnte man das MacLaurin Verfahren in C implementieren:

```
/* Programm MacLaurin Integral                                   (File: MACL_INT.C )
   Numerische Integration einer Funktion f nach dem MacLaurin Verfahren.

   Arbeitsweise:
   Aufsummieren von n/2 Teilflaechen die durch die Intervalle [xi-1,xi+1] * h repraesentiert
   werden zu einer Gesamtsumme die eine Naeherung des Integrals darstellt.
   Voraussetzung:
   Es wird vorausgesetzt, dass die Funktion im zu integrierenden Intervall
   integrierbar ist. n muss ein Gerade Zahl sein.

   Autor: Gerhard Krucker
   Datum: 3.5.1995
   Sprache: MS Visual-C V1.5 (QuickWin Applikation)
 */

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

/* Definition der Funktion f, die ueber das Intervall [a,b] integriert werden
   soll. Sie liefert als Resultat den Funktionswert zurueck.
   Hier wird fuer das Testbeispiel die Funktion e^x implementiert.
*/
double f (double x)
{
  return exp(x);
}

int main()
{  int n;                     /* Anzahl Teilintervalle (Teilflaechen, Zerlegung) */
   double a,b;        /* Integrationsgrenzen */
   double int_summe; /* Summe der einzelnen Teilflaechen */
   double h;         /* Intervallbreite */
   int i;            /* Schleifenvariable */

   printf("Integration nach dem MacLaurin-Verfahren\n");
   printf("Eingabe der Integrationsgrenzen a  b: ");
   scanf("%lf %lf",&a,&b);
   printf("Eingabe der Anzahl Teilintervalle n: ");
   scanf("%d",&n);

   int_summe=0;
   h = (b - a) / n;

   for (i = 0; i <= n-1; i += 2)
     { double xi;        /* Argument im Schritt i */
       double teilfl;    /* Teilflaeche */

       xi = (i+1) * h + a;  /* xi in die Mitte des Intervalls [xi-1,xi+1] setzen */
       teilfl = 2 * h *f(xi);    /* MacLaurin Summenglied berechnen */
       int_summe += teilfl;
      }
    printf("Integral: %10.8g\n",int_summe);

    return 0;
 }
```

Wir vergleichen nun die Summen beim Rechteckverfahren dem MacLaurin-Verfahren für $f(x) = e^x$ bei verschiedenen n und erhalten die Übersicht:

n	$R(n)$	$M(n)$
10	1.6338	1.7154214
100	1.7097	1.7182532
1000	1.71742	1.7182815
10000	1.7182	1.7182818

7.3.2 Fehlerabschätzung des MacLaurin-Verfahrens

Es lässt sich zeigen, dass beim MacLaurin-Verfahren der Fehler in den Summen in der Grössenordnung

$$\left|\int f - M(n)\right| \le \frac{(b-a)^3}{6n^2}\max|f''| = \frac{b-a}{6}h^2\max|f''| \tag{7.15}$$

Wobei hier das Maximum der zweiten Ableitung auf dem Intervall $[a,b]$ zu nehmen ist. Wir haben hier also eine wesentlich bessere Konvergenz als beim Rechteckverfahren, obwohl wir nur mit der Hälfte der Rechtecke rechnen.

Dennoch brauchen wir für eine Genauigkeit von 6 Nachkommastellen 1000 Stützstellen zu berechnen. Dieses Verfahren ist also sehr rechenintensiv und konvergiert zu langsam für die Praxis.

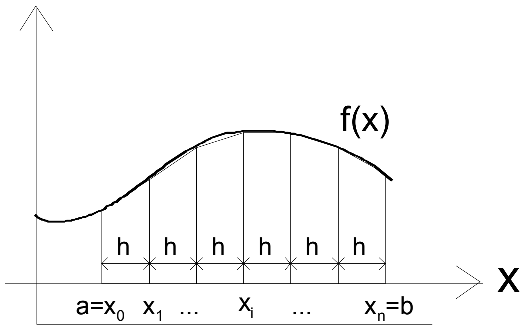
Bessere Verfahren arbeiten auf der Grundlage, dass die Teilflächen nicht durch Rechtecke, sondern durch Sehnenpolygonzüge oder Parabeln angenähert werden. Dadurch lassen sich die Teilflächen wesentlich besser annähern als durch eine Treppenfunktion.

7.4 Trapezregel

Die Beschreibung der Teilflächen erfolgt hier mit Trapezflächen. Dies bringt eine wesentliche Verbesserung des numerischen Integralwertes gegenüber der einfachen Rechteckmethode. Allerdings erreichen wir keine wesentliche Verbesserung gegenüber dem MacLaurin-Verfahren, da hier auch der Fehler nur mit n^2 abnimmt.
Trotzdem ist die Trapezmethode eine der grundlegenden numerischen Integrationsmethoden, die relativ einfach zu implementieren ist.

Die einzelnen Teilflächen werden nach der allgemeinen Flächenformel für das Trapez berechnet und wir erhalten für eine Zerlegung des Intervalls [a,b] in n äquidistante Teilintervalle der Länge $h=(b-a)/n$ die Teilfläche, dargestellt durch das Trapez i:

Fläche des Trapez i:



$$T_i = \frac{h}{2}\left(f(x_i) + f(x_{i+1})\right) \quad i = (0,\dots,n-1) \tag{7.16}$$

Wir summieren nun alle diese Teilflächen auf und erhalten die Trapezsumme der Ordnung n:

$$T(n) = \sum_{i=0}^{n-1} T_i = \sum_{i=0}^{n-1}\frac{h}{2}\left(f(x_i)+f(x_{i+1})\right) = \frac{h}{2}\sum_{i=0}^{n-1}\left(f(x_i)+f(x_{i+1})\right) = \frac{h}{2}\sum_{i=0}^{n-1}\left(f(a+h)+f(a+(i+1)h)\right)$$

$$\Rightarrow T(n) = \frac{h}{2}\left(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right) \tag{7.17}$$

Bemerkung: Die 2 vor dem Summenzeichen entsteht dadurch, dass die inneren Stützstellen $i=1,\dots,n-1$ durch das Aneinanderstossen doppelt gezählt werden.

7.4.1 Implementierung der Trapezregel

Die Berechnung einer Trapezsumme der Ordnung n lässt sich leicht in ein Programm umsetzen:

```
/* Berechnen des Integrales nach der Trapezregel.           (File: TRAPEZ.C)

   Autor: Gerhard Krucker
   Datum: 15.5.1995
   Sprache: MS Visual-C V1.5
*/

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

/* Zu integrierende Funktion: e^x */
double f(double x)
{
   return exp(x);
}

main()
{ double a,b,h,x,n;
  double s;          /* Innere Summen */
  double I;          /* Integralwert */
  int i;
  int n;             /* Anzahl Teilintervalle */

  printf("Berechnen des Integrales nach der Trapezregel für f(x):\n");
  printf("Eingabe der unteren Integrationsgrenze a= ");
  scanf("%lf",&a);
  printf("Eingabe der oberen Integrationsgrenze b= ");
  scanf("%lf",&b);
  printf("Anzahl Teilintervalle n= ");
  scanf("%d",&n);

  h = (b - a) / n;    /* x-Schrittweite berechnen */
  s = 0;
  for (i=1; i <= n-1; i++)    /* Innere Summen berechnen */
    { x = a + i * h;
      s = s + f(x);
    }
  I = (h / 2 )* (f(a) + 2*s + f(b));

  printf("Integralwert = %10.8g\n",I);

  return 0;
}
```

Wir vergleichen das Resultat für unser Testbeispiel $f(x) = e^x = 1.1718281828\ldots$ mit den vorherigen Verfahren:

n	*R(n)*	*M(n)*	*T(n)*
10	1.6338	1.7154214	1 .7182961
100	1.7097	1.7182532	1.7182961
1000	1.71742	1.7182815	1.718282
10000	1.7182	1.7182818	1.7182818

Wir sehen, dass das Trapezverfahren etwa in derselben Genauigkeitsordnung wie das MacLaurin-Verfahren liegt.

7.4.2 Fehlerabschätzung der Trapezregel

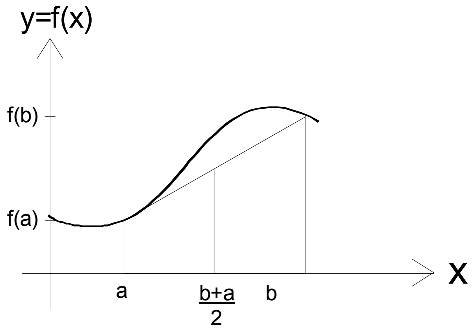
Für die Trapezregel gilt die Fehlerabschätzung:

Wenn f'' stetig ist im Intervall $[a,b]$ und die Trapezregel mit einer äquidistanten Zerlegung h angewandt wird um das Integral $\int_a^b f(x)dx$ zu berechnen, dann gilt für irgendein $\xi \in (a,b)$:

$$\left| T - \int_a^b f(x)dx \right| \le \frac{n}{12} h^3 \max_{[a,b]} |f''| = \frac{(b-a)^3}{12n^2} \max_{[a,b]} |f''| \tag{7.18}$$

7.4.2.1 Begründung der Fehlerabschätzungsformel

Wir betrachten hierzu ein Modell, dass mit einer Trapezfläche eine Näherung vornimmt und schliessen nachher auf den Fehler mit n Teilflächen:



Den Fehler definieren wir als Betrag der Differenz:

$$\left| T - \int_a^b f(x)dx \right| \tag{7.19}$$

Als Hilfsmittel definieren wir die Funktion F als bestimmtes Integral:

$$F(t) = \int_a^t f(x)dx = \tag{7.20}$$

Voraussetzung: f ist n mal differenzierbar ($n > 3$). Wir schreiben nun die Integralfunktion F als Taylor-Reihe mit Entwicklungspunkt a:

$$F(a+h) = F(a) + hF'(a) + \frac{h^2}{2!}F''(a) + \frac{h^3}{3!}F'''(a) + \cdots \tag{7.21}$$

Da F die Stammfunktion von f ist folgt: $F'=f$, $F''=f'$, $F''=f''$ usw. Ferner ist $F(a)=0$.
Wir setzen das nun in die Taylor-Reihe ein und erhalten:

$$\int_a^{a+h} f(x)dx = h f(a) + \frac{h^2}{2!}f'(a) + \frac{h^3}{3!}f''(a) + \cdots$$

Die Taylor-Reihe für f mit Entwicklungspunkt a ist:

$$f(a+h) = f(a) + h f'(a) + \frac{h^2}{2!}f''(a) + \frac{h^3}{3!}f'''(a) + \cdots \tag{7.22}$$

Wir addieren auf beiden Seiten der Gleichung *f(a)* und multiplizieren mit $\frac{h}{2}$:

$$\frac{h}{2}\left[f(a)+f(a+h)\right]=h\,f(a)+\frac{h^2}{2}f'(a)+\frac{h^3}{4}f''(a)+\cdots$$

Wir subtrahieren beide Taylor-Reihen und erhalten:

$$\int_a^{a+h} f(x)dx-\frac{h}{2}\left[f(a)+f(a+h)\right]=-\frac{h^3}{12}f''(a)+\cdots \tag{7.23}$$

7.4.2.2 Anwendung der Fehlerformel

Wir können durch Anwenden der Abschätzungsformel die minimale Anzahl Teilflächen bestimmen, die zum Erreichen einer bestimmten Genauigkeit erforderlich sind.

Beispiel:
Mit der Trapezregel soll das Integral $\int_0^1 e^{-x^2}dx$ mit einem maximalen Fehler von $0.5\cdot 10^{-4}$ berechnet werden. Bestimmen Sie die notwendige Anzahl Teilflächen *n*!

Lösung:
Aus der Fehlerformel (7.18) erhalten wir:

$$\frac{1}{2}10^{-4}\le\frac{n}{12}h^3\max_{[a,b]}|f''|\quad\overset{h=\frac{b-a}{n}}{=}\quad\frac{(b-a)^3}{12n^2}\max_{[a,b]}|f''|$$

Wir bestimmen:

$$f(x)=e^{-x^2}\qquad f'(x)=-2x\,e^{-x^2}\qquad f''(x)=\left(4x^2-2\right)e^{-x^2}$$

Eine Rechnung zeigt, dass $|f''(x)|\le 2$ auf dem ganzen Intervall [0,1]. Daher wird die notwendige Anzahl Teilflächen:

$$n=\sqrt{\frac{(b-a)^3}{12\varepsilon\max_{[a,b]}|f''|}}=\sqrt{\frac{1}{12\cdot 0.5\cdot 10^{-4}\cdot 2}}=57.73..\rightarrow 58 \text{ Teilflächen}$$

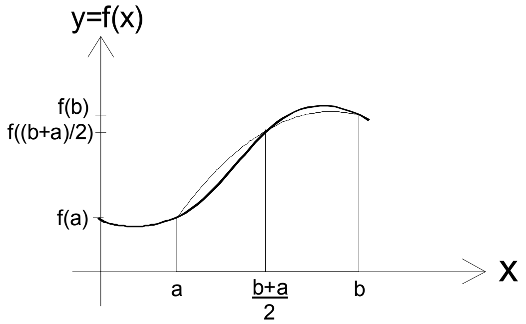
7.5 Simpson Regel

Das Simpson-Verfahren ermöglicht Integralberechnungen mit einer wesentlich besseren Genauigkeit als die vorher gezeigten Verfahren (diverse Rechteckregeln, Trapezregel).

Die Simpsonregel beruht darauf, dass die Stützpunkte durch Parabelbögen verbunden werden, anstatt durch Geraden (Rechteckregel, Trapezregel). Dies bringt eine wesentlich bessere Näherung, sodass viel weniger Rechenschritte für ein genaues Resultat notwendig sind.

Diese Integrationsregel beruht auf der sog. keplerschen Fassregel, die vom bekannten Astronomen Johannes Kepler (1571-1630) vorgestellt wurde, um die Fläche eines solchen fassförmigen Flächenelementes näherungsweise zu bestimmen. Der sonst nicht weiter bekannte englische Mathematiker Thomas Simpson (1710-1761) benutzte diese Formel um den Integralwert mit n Teilflächen zu berechnen.

Die Näherungsfläche wird als Parabelbogen, der durch die Punkte f(a), f((a+b)/2) und f(b) geht, gelegt:



7.5.1 Keplersche Fassregel

Sie liefert die Fläche *S*, die zwischen dem Parabelbogen und der *x*-Achse liegt:

Keplersche Fassregel

$$m = \frac{b+a}{2} \tag{7.24}$$

$$S = \frac{b-a}{6}\left(f(a) + 4f(m) + f(b)\right) \tag{7.25}$$

Die Begründung der Formel ist einfach, wenn man die entsprechende Funktion für die Parabel kennt. Diese Polynomfunktion 2. Grades ist ein Interpolationspolynom, das durch die Funktionswerte an den Stellen *a, (b-a)/2* und *b* geht. Das Polynom kann durch geeignete Methoden (Lagrange, Newton, etc.) bestimmt werden.

Wir bestimmen dazu das Lagrange-Interpolationspolynom 2. Grades, das durch die Punkte

a	*(a+b)/2*	*b*
f(a)	*f((a+b)/2)*	*f(b)*

verläuft.
Die Lagrange-Polynome L_0, L_1, L_2 werden gemäss Kapitel 3 :

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(2x-a-b)(x-b)}{(b-a)^2} \qquad L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{4(a-x)(x-b)}{(b-a)^2}$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(2x-a-b)(x-a)}{(b-a)^2}$$

Das Interpolationspolynom wird demnach:

$$p_2(x) = f(a)\cdot L_0 + \cdot L_1 f(m) + f(b)\cdot L_2$$
$$= f(a)\frac{(2x-a-b)(x-b)}{(b-a)^2} + f\left(\frac{a+b}{2}\right)\frac{4(a-x)(x-b)}{(b-a)^2} + f(b)\frac{(2x-a-b)(x-a)}{(b-a)^2}$$

Zur weiteren Berechnung benutzen wir Mathematica, da die Umformungen recht aufwendig werden. Wir integrieren das Interpolationspolynom von [a,b] und erhalten sofort formal die keplersche Fassregel:

```
(* Beweis Keplersche Fassregel mit Lagrange-Polynomen
   Gerhard Krucker
   6.6. 1997
 *)

x0 = a;
x1 = (a + b) / 2;
x2 = b;
y0 = fa;
y1 = fab;
y2 = fb;
L0 = (x - x1) / (x0 - x1) * ( x - x2) / ( x0 - x2) // Simplify
L1 = (x - x0) / (x1 - x0) * ( x - x2) / ( x1 - x2) // Simplify
L2 = (x - x0) / (x2 - x0) * ( x - x1) / ( x2 - x1) // Simplify
```

$$\frac{(a+b-2x)\ (b-x)}{(a-b)^2}$$

$$\frac{4\ (a-x)\ (-b+x)}{(a-b)^2}$$

$$\frac{(a+b-2x)\ (a-x)}{(a-b)^2}$$

```
p2x = fa L0 + fab L1 + fb L2 // Simplify
```

$$\frac{1}{(a-b)^2}\left(b^2\, fa + a^2\, fb - b\ (3\, fa - 4\, fab + fb)\ x + 2\ (fa - 2\, fab + fb)\ x^2 + a\ (b\ (fa - 4\, fab + fb) - (fa - 4\, fab + 3\, fb)\ x)\right)$$

```
Integrate[p2x, {x, a, b}] // Simplify
```

$$-\frac{1}{6}\ (a-b)\ (fa + 4\, fab + fb)$$

Ein alternativer Weg kann wiederum über einer Reihenentwicklung beschritten werden. Dieser Weg hat zudem den Vorteil, dass der verbleibende Fehler als Restterm erscheint.

Bemerkenswert ist, dass sogar kubische Parabeln exakt integriert werden. Dies ergibt sich auch aus der Fehlerabschätzung für die Fassregel:

$$\left|\int_a^b f(x)dx - S\right| \le \frac{(b-a)^5}{2880}\max\left|f^{(4)}\right| \tag{7.26}$$

Diese Abschätzung gilt für Funktionen, die mindestens 4 mal stetig differenzierbar sind. Das Maximum ist auf dem Integrationsintervall $[a,b]$ zu nehmen.

7.5.2 Integration mit der Simpson-Regel

Erst die mehrfache Anwendung der Fassregel ergibt die eigentliche Simpson-Regel. Dies erfolgt indem in bekannter Weise das Integrationsintervall in *n* Teilintervalle zerlegt wird und die einzelnen Teilflächen aufsummiert werden.

Dazu wird das Integrationsintervall $[a,b]$ in eine gerade Anzahl *n* äquidistante Teilintervalle $h = \frac{b-a}{n}$ zerlegt.

Wir erhalten *n+1* Teilpunkte auf der *x*-Achse und berechnen die Stützstellen: $x_i = a + ih$.
Nun wenden wir die Grundformel (Fassregel) auf das Intervall $[x_i, x_{i+2}]$ an. Wir erhalten *n/2* Summen für die Teilflächen der Form

Simpson-Formel

$$S_i = \frac{2h}{6}\left(f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\right) \qquad i = 0,2,4,\ldots,n-2 \tag{7.27}$$

Den Integralwert erhalten wir als Addition der *n/2* Teilsummen zu einer Simpson-Summe *S(n)* der Ordnung *n*:

$$S(n) = \frac{h}{3}\sum_{i=0}^{n} c_i f(x_i) \quad c_i = \begin{cases} 1 & i = 0 \text{ oder } i = n \\ 4 & i \text{ ungerade} \\ 2 & i \text{ gerade, aber } \neq 0,n \end{cases} \tag{7.28}$$

Die c_i werden Simpson-Gewichte genannt. Die Gewichte ergeben sich daraus, dass bei den geraden Indizes jeweils zwei Doppelintervalle aneinander stossen. Diese Funktionswerte müssen dann doppelt gerechnet werden, ausser bei *n* und 0. Der Faktor 4 für die ungeraden Indizes ergibt sich aus der 4 für den mittleren Wert in der Grundformel.

Beispiel: $\int_0^2 x^2 dx$ mit einer Zerlegung in n=10 Teilintervalle.

$$h = \frac{x_n - x_0}{n} = \frac{2-0}{10} = 0.2$$

$$f(x_0) = 0^2 = 0$$

$$su = \sum_{i=1,3,..}^{n-1} f(x_i) = 0.2^2 + 0.6^2 + 1^2 + 1.4^2 + 1.8^2 = 6.6$$

$$sg = \sum_{i=2,4,..}^{n-2} f(x_i) = 0.4^2 + 0.8^2 + 1.2^2 + 1.6^2 = 4.8$$

$$f(x_n) = 2^2 = 4$$

$$S(10) = \frac{h}{3}\left(f(x_0) + 4su + 2sg + f(x_n)\right) = \frac{0.2}{3}(0 + 4\cdot 6.6 + 2\cdot 4.8 + 4) = 2.6667$$

7.5.3 Implementierung der Simpson-Regel

Wir entwerfen in einer ersten Version ein Programm, dass für eine gegebene gerade Anzahl Teilintervalle die Simpson-Summe *S(n)* als Integralnäherung berechnet.

Entwurf:
Wir wenden zur Berechnung der Teilsummen die keplersche Fassregel an, wobei wir unterschiedliche Variablen für die Funktionswerte mit geraden und ungeraden Indizes verwenden.

$$m = \frac{b+a}{2} \quad h = \frac{b-a}{n} \qquad n\text{: Anzahl Teilintervalle (gerade)}$$

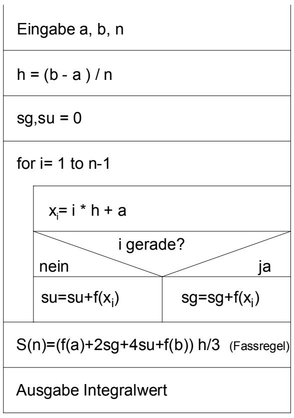
$$S = \frac{b-a}{6}\left(f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\right)$$

sg: Summe der Funktionswerte mit gradem Index

su: Summe der Funktionswerte mit ungradem Index

S: Simpson Summe *S(n)*

Der Programmablauf lässt sich auf einer tiefen Ebene mit einem Nassi-Shneiderman-Diagramm beschreiben:



Die Prüfung im Programm ob ein Index gerade ist , führen wir mit einer modulo-2 Division durch. Wenn der Rest (Modulus) eine 1 ist, so ist der Index ungerade andernfalls gerade.

7.5.3.1 Testbeispiele

Wir vergleichen die Simpson-Regel mit bisher verwendeten anderen numerischen Integrationsmethoden am Testintegral *f(x)=exp(x)* im Intervall [0,1] und stellen eine Tabelle auf:

n	*R(n)*	*M(n)*	*T(n)*	*S(n)*
2	1.32436	1.6487213	1.7539311	1.7188612
10	1.6338	1.7154214	1 .7182961	1.7182828
100	1.7097	1.7182532	1.7182961	1.7182818
1000	1.71742	1.7182815	1.718282	1.7182818

Der exakte Integralwert ist e-1=1.718281828.. . Wir haben mit der Simpson-Regel bereits für ein n=10 eine brauchbare Genauigkeit erreicht. Wird bei der Simpson-Regel die Anzahl Schritte n verdoppelt, so wird der Fehler um den Faktor 16 verkleinert.

Ein Vorschlag zu konkreten Implementierung in C:

```
/* Programm Simpson Integral                                        (File: SIMPSON1.C )
   Numerische Integration einer Funktion f nach der Simpson-Regel.

   Arbeitsweise:
   Aufsummieren Berechnen der Summen für die geraden und ungeraden Indizes i.
   Anschliessend verrechnen der Summen in der keplerschen Fassregel.
   Voraussetzung:
   Es wird vorausgesetzt, dass die Funktion im zu integrierenden Intervall
   integrierbar ist. n muss eine positive gerade Zahl sein.

   Autor: Gerhard Krucker
   Datum: 3.5.1995
   Sprache: MS Visual-C V1.5 (QuickWin Applikation)
 */

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

/* Definition der Funktion f, die ueber das Intervall [a,b] integriert werden
   soll. Sie liefert als Resultat den Funktionswert zurueck.
   Hier wird fuer das Testbeispiel die Funktion e^x implementiert.
*/
double f (double x)
{
  return exp(x);
}

int main()
{  int n;              /* Anzahl Teilintervalle (Teilflaechen, Zerlegung) */
   double a,b;         /* Integrationsgrenzen */
   double su,sg;       /* Summen fuer geraden und ungeraden Indizes */
   double h;           /* Intervallbreite */
   int i;              /* Schleifenvariable */
   double sn;          /* Simpsonsumme S(n) (Integralwert) */

   printf("Integration nach der Simpson-Regel\n");
   printf("Eingabe der Integrationsgrenzen a  b: ");
   scanf("%lf %lf",&a,&b);
   printf("Eingabe der (geraden) Anzahl Teilintervalle n: ");
   scanf("%d",&n);

   h = (b - a) / n;
   su=sg=0;

   for (i = 1; i <= n-1; i++)
     { double xi;        /* Argument im Schritt i */

       xi = i * h + a;
       if ((i % 2) == 0)
          sg += f(xi);
         else
          su += f(xi);
      }
    sn = (f(a) + 2*sg + 4*su + f(b))*h/3;

    printf("Integral: %10.8g\n",sn);

    return 0;
 }
```

7.5.4 Problemfälle für die Simpson-Regel

Obwohl die Simpson-Regel für die Praxis ein gutes Integrationsverfahren ist, sollen die in der Praxis auftretenden Standard-Problemfälle etwas näher gezeigt werden. Diese Problemfälle sind übrigens nicht spezifisch für die Simpson-Regel, sondern gelten (ev. in angepasster Form) auch für alle anderen Verfahren.

Die Problemfälle beruhen alle darauf, dass die Voraussetzungen für eine gute Konvergenz der Verfahren verletzt werden. Dies ist dann gegeben, wenn die jeweiligen Fehlerabschätzungsformeln nicht angewandt werden können.

Generell sind aber Funktionen mit Unstetigkeitsstellen praktisch bei allen Verfahren problematisch, wenn nicht spezielle Vorkehrungen getroffen werden.

7.5.4.1 Unstetigkeitsstellen im Integrationsintervall

Nach Voraussetzung ist eine Funktion Riemann-integrierbar, wenn sie im Integrationsintervall beschränkt ist und höchstens endlich viele Unstetigkeitsstellen hat. Die Simpson-Regel liefert für stetige Funktionen in der Regel eine gute Approximation.

Haben wir jedoch eine (oder mehrere) Unstetigkeitstellen im Integrationsbereich, so kann die Teilfläche mit dieser Unstetigkeitsstelle nur sehr ungenau approximiert werden. Dies äussert sich in einer mehr oder weniger grossen Abweichung vom exakten Integralwert.

Als Beispiel einer unstetigen Funktion betrachten wir:

$$f(x) := \begin{cases} -1 & x \le 0 \\ +1 & x > 0 \end{cases}$$

Wir wollen diese bei 0 unstetige Funktion im Intervall [-1,+1] mit der Simpson-Regel integrieren. Wir wissen, dass der exakte Wert natürlich 0 ergeben muss und betrachten die Abweichungen für verschiedene Anzahl Teilintervalle *n*:

Wir definieren für das Programm auf Seite 7-19 die zu integrierende Funktion `f` neu:

```
double f (double x)
{ if (x <= 0)
     return -1;
   else
     return 1;
}
```

Wir erhalten folgende Simpson-Summen S(n):

n	*S(n)*
2	-1.3333333
10	0.26666667
100	0.013333333

Wir sehen, dass die Simpson-Regel nur "gut" arbeitet, wenn die Funktion mindestens viermal stetig differenzierbar ist. Dies ist bei dieser Funktion natürlich nicht erfüllt und somit kann auch die Fehlerabschätzungsformel nicht angewandt werden.

7.5.4.2 Senkrechte Tangente im Integrationsintervall

Hat eine Funktion eine senkrechte Tangente kann davon ausgegangen werden, dass die Funktion ebenfalls nicht stetig differenzierbar ist und somit die Fehlerabschätzung nicht gilt.

Ein einfaches Beispiel:
$f(x) := \sqrt{x}$ soll im Intervall [0,1] integriert werden. Wir können mit Hilfe der Stammfunktion den exakten Integralwert berechnen und erhalten $I = 2/3$.
Da für $x = 0$ f eine senkrechte Tangente hat, erhalten wir mit der Simpson-Regel die recht ungenauen Integralnäherungen über die Simpson-Summen *S(n):*

n	*S(n)*
2	0.63807119
10	0.66409959
100	0.66658548

7.5.4.3 Verbesserungsmöglichkeiten

In einigen Fällen kann man mit sog. adaptiven Verfahren eine Verbesserung der Genauigkeit erreichen. Diese Verfahren arbeiten nicht mehr mit konstanter Schrittweite, sondern rechnen an den kritischen Stellen mit einer feineren Schrittweite.

7.6 Allgemeine Integrationsverfahren nach Newton-Cotes

Während die Simpson-Regel eine Parabel als Näherung für die zu integrierende Funktion verwendet, liegt es nahe auch Funktionen höheren Grades zu verwenden.

Diese erbringen dann in der Regel eine bessere Genauigkeit des Integralwertes, verbunden mit einem höheren Rechenaufwand.

Alle bisher vorgestellten Methoden lassen sich unter die sog. klassischen *k*-Punkte Methoden zusammenfassen, die auf der Definition aufbauen:

Der Integralwert einer Teilfläche im Intervall $[x_i, x_{i+k-1}]$, gegeben durch die Figur mit den k Stützstellen x_i, äquidistant mit Schrittweite h

$$x_i, x_{i+1,} \cdots, x_{i+k-1} \qquad x_j = x_i + h \cdot j \qquad h = \frac{x_{i+k-1} - x_i}{k-1} \quad j = 0, \ldots, k-1$$

und den zugehörigen Stützwerten $y_i = f(x_i)$ ergibt sich:

$$\int_{x_i}^{x_{i+k-1}} f(x)dx = \frac{x_{i+k-1} - x_i}{t} \sum_{j=0}^{n-1} \gamma_j y_{i+j} + r_k(f) \tag{7.29}$$

Newton-Cotes Quadraturformel

Die charakteristischen Grössen $t, \gamma_j, r_k(f)$ sind zur Rechnung in den gängigen Standardwerken tabelliert aufgeführt:

k	t	γ_0	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	$r_k(f)$
2	1	1	1						$-\frac{1}{12} h^3 f''(\xi)$
3	6	1	4	1					$-\frac{1}{90} h^5 f^{(4)}(\xi)$
4	8	1	3	3	1				$-\frac{3}{80} h^5 f^{(4)}(\xi)$
5	90	7	32	12	32	7			$-\frac{8}{945} h^7 f^{(6)}(\xi)$
6	288	19	75	50	50	75	19		$-\frac{275}{12096} h^7 f^{(6)}(\xi)$
7	840	41	216	27	272	27	216	41	$-\frac{9}{1400} h^9 f^{(8)}(\xi)$

Newton-Cotes Koeffizienten

Wir sehen, dass wir für die verschiedenen *k* die bekannten Verfahren erhalten:

k=1 Rechteckregel
k=2: Trapezregel
k=3: Simpson-Regel

Man beachte, dass alle Newton-Cotes-Formeln mit k=gerade eine schlechtere Genauigkeit haben als das nächst tiefere Verfahren ungerader Ordnung.

7.6.1 Anwendung des Newton-Cotes-Verfahrens

Wir zeigen am Beispiel den Entwurf für k=5, also die Näherung mit Hilfe eines Polynoms 4. Grades, das durch k=5 Stützstellen beschrieben wird. Diese Methode ist als Bode-Verfahren bekannt:

Wir definieren die 5 Stützstellen äquidistant:

$$x_0 = a \quad x_1 = a+h \quad x_2 = a+2h \quad x_3 = a+3h \quad x_4 = a+4h \quad h = \frac{b-a}{4}$$

und erhalten die Grundformel für eine Teilfläche:

$$B = \frac{x_4 - x_0}{90}\left[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)\right]$$

Wir wenden die Grundformel nun auf Viererintervalle $[x_i, x_{i+4}]$ an und wir erhalten $n/4$ Summen der Form:

$$B_i = \frac{4h}{90}\left[7f(x_i) + 32f(x_{i+1}) + 12f(x_{i+2}) + 32f(x_{i+3}) + 7f(x_{i+4})\right] \tag{7.30}$$

Wir summieren n-1 Teilflächen (= n Stützstellen) auf, fassen die Glieder zusammen und bilden die Gewichte c_i:

$$B(n) = \frac{2h}{45}\sum_{i=0}^{n} c_i f(x_i) \tag{7.31}$$

Zur Rechnung bilden wir wie in der Simpson-Regel getrennte Summen für die Indizes i:

s1: ungerade Indizes
s2: gerade, aber nicht durch 4 teilbare Indizes
s4: durch 4 teilbare Indizes

Ein Testlauf für das Integral $I = \int_0^1 e^x dx$ ergibt die Werte für verschiedene n im Vergleich zur Simpson-Regel:

n	*S(n)*	*B(n)*
4	1.7183188	1.7182827
12	1.7182823	1.7182818
100	1.7182818	1.7182818

Wir bemerken die ausserordentlich rasche Konvergenz des Verfahrens.

7.6.2 Fehlerbetrachtung des Newton-Cotes Verfahrens

Unter der Voraussetzung, dass die zu integrierende Funktion f im zu integrierenden Intervall sechsmal stetig differenzierbar ist, kann folgende Fehlerbetrachtung und -abschätzung durchgeführt werden:

Gemäss Tabelle ist das Restglied $r_k(f)$, das den Fehler verkörpert:

$$-\frac{8}{945}h^7 f^{(6)}(\xi) \qquad \xi \in [a,b]. \tag{7.32}$$

Für eine Maximalbetrachtung des Fehlers wird x aus dem Integrationsintervall so gewählt, dass der Ausdruck maximal wird:

$$\left|B-\int f\right| \le \frac{8}{945} h^7 \max\left|f^{(6)}\right| \tag{7.33}$$

Wir setzen für h die Definition ein und betrachten den Fehler für eine Teilfläche, also n=4, und erhalten:

$$\left|B-\int f\right| \le \frac{8}{945}\frac{(b-a)^7}{4^7} \max\left|f^{(6)}\right| = \frac{(b-a)^7}{1935360} \max\left|f^{(6)}\right| \tag{7.34}$$

Dies ist die Grundformel für die Fehlerabschätzung des Bode-Verfahrens. Die allgemeine Form kann leicht aus dieser Formel entwickelt werden.

Für ein gegebenes n (Vielfaches von 4) erhalten wir dann folgendes Programm:

```
/* Programm Bode Integral                                              (File: BODE1.C )
   Numerische Integration nach dem Bode Verfahren als Beispiel zur numerischen
   Integration nach Newton-Cotes.

   Arbeitsweise:
   Aufsummieren Berechnen der Summen für die Indizes i:
   s1: ungerade Indizes
   s2: gerade, aber nicht durch 4 teilbare Indizes
   s4: durch 4 teilbare Indizes

   Anschliessend werden die Teilsummen in der Grundformel verrechnet.

   Voraussetzung:
   Es wird vorausgesetzt, dass die Funktion im zu integrierenden Intervall
   integrierbar ist. n muss eine positive durch 4 teilbare Zahl sein.

   Autor: Gerhard Krucker
   Datum: 30.5.1995
   Sprache: MS Visual-C V1.5 (QuickWin Applikation)
 */

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

/* Definition der Funktion f, die ueber das Intervall [a,b] integriert werden
   soll. Sie liefert als Resultat den Funktionswert zurueck.
   Hier wird die Funktion e^x implementiert.
*/
double f (double x)
{
   return exp(x);
}

int main()
{  int n;           /* Anzahl Teilintervalle (Zerlegung) */
   double a,b;      /* Integrationsgrenzen */
   double s1,s2,s4; /* Summen fuer Indizes */
   double h;        /* Intervallbreite */
   int i;           /* Schleifenvariable */
   double Bn;       /* Bodesumme B(n) (Integralwert) */

   printf("Integration nach der Bode-Regel\n");
   printf("Eingabe der Integrationsgrenzen a  b: ");
   scanf("%lf %lf",&a,&b);
   printf("Eingabe der (durch 4 teilbaren) Anzahl Teilintervalle n: ");
   scanf("%d",&n);

   h = (b - a) / n;
   s1=s2=s4=0;
```

```
 for (i = 1; i <= n-1; i++)
     { double xi;        /* Argument im Schritt i */

       xi = i * h + a;
       if ((i % 2) == 0)
         if ((i % 4) == 0)
             s4 += f(xi);
           else
             s2 += f(xi);
         else
           s1+= f(xi);
       }

    Bn = (7*f(a) + 32*s1 +12*s2+ 14*s4+7*f(b))*2*h/45;

    printf("Integral: %10.8g\n",Bn);

    return 0;
 }
```

7.7 Andere Integrationsverfahren

Die numerische Integration kennt noch ein Reihe weiterer Standardverfahren. Sie bauen meist auf ähnlichen Ideen wie die bisher gezeigten Methoden auf, jedoch werden die Teilflächen anders gebildet und verrechnet.

Der Vollständigkeit halber werden noch einige weitere bekannte Methoden kurz erwähnt:

Romberg-Verfahren

Das Romberg-Verfahren erzeugt einen Satz von Integralnäherungen, die die bisher gezeigten Trapezverfahren und Simpson, etc. als Spezialfälle einschliessen. Romberg entwickelt die Summen sukzessiv nach Newton-Cotes und liefert daher die genauesten Resultate, ist aber das aufwendigste Verfahren.

Gauss-Legendere

Dieses Verfahren arbeitet nicht mit äquidistanten Stützstellen, sondern mit gewichteten Stützstellen. Die Funktion, resp. die Teilflächen, wird durch ein Interpolationspolynom möglichst hohen Grades dargestellt.

Adaptive Verfahren

Die Schrittweite wird aus der Steigung der zu integrierenden Funktion bestimmt. Dadurch können auch Funktionen mit Unstetigkeitsstellen einigermassen genau integriert werden. (Bsp: Adaptive Simpson-Regel). Gut arbeitende Verfahren werden schnell recht aufwendig in der Codierung.

7.8 Aufgaben

Riemann-Integral, Rechteckregel

1. Die Funktion $\int_0^1 \frac{dx}{x^2+1}$ wird mit der Zerlegung $Z = \left\{0, \frac{1}{2}, 1\right\}$ betrachtet.
   a.) Bestimmen Sie die (Riemann-) Untersummen für diese Zerlegung.
   b.) Bestimmen Sie nach der linken Rechteckregel die "linken Summen". (m_k ist nicht mehr das Infinum, sondern immer $f(x_{k-1})$.
   c.) Wie gross ist der Fehler der Näherung durch die Summen in a.), b.) bezüglich des exakten Integralwertes?

2. Wie lautet das Resultat, wenn wir $\int_1^2 \frac{1}{x} dx$ mit Obersummen und der Zerlegung $Z = \left\{1, \frac{3}{2}, 2\right\}$ berechnen?

3. Die (linke) Rechteckregel für die numerische Integration ist ähnlich den Ober- und Untersummen, nur etwas einfacher
   $$\int_a^b f(x)\,dx = \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i)$$
   Wobei die Zerlegung $Z = \{a = x_0 < x_1 < x_2 < \cdots < x_n = b\}$ ist. Zeigen Sie, dass die Summe für $n \to \infty$ zum Integralwert konvergiert.
   Hinweis: Betrachten Sie das als spezielle Rechtecksumme und beachten Sie die Ungleichung der Ober- und Untersummen für die Integrierbarkeit einer Funktion.

Trapezregel

4. Wie lauten die Teilflächenwerte, wenn die Trapezregel auf die Funktion $f(x) = \frac{1}{x}$ mit den Stützstellen $1, \frac{3}{2}, 2$ angewandt wird?

5. Berechnen Sie mit der Trapezregel und drei Stützpunkten eine Näherung für das Integral $\int_0^1 \frac{dx}{x^2+1}$.
   Vergleichen Sie die Näherung mit dem exakten Wert und bestimmen Sie eine Fehlerabschätzung und bestimmen Sie eine Fehlerschranke als numerischen Wert.

6. Betrachten Sie die Funktion $f(x) = |x|$ im Intervall [-1,1]. Berechnen Sie die Resultate unter Verwendung der nachfolgenden numerischen Integrationsverfahren um $\int_{-1}^1 f(x)dx$ zu nähern mit den Intervallbreiten $h = \{2, 1, \frac{1}{2}, \frac{1}{4}\}$.
   a.) Linke Rechteckregel (linke Summe)
   b) Untersumme
   c.) Obersumme
   d.) Trapezregel

   Vergleichen Sie die Resultate mit dem exakten Wert, geben sie den absoluten Fehler an.

7. Wie gross muss n gewählt werden damit $\int_0^2 e^{-x^2} dx$ mit der Trapezregel mit einem Fehler $\le 10^{-6}$ berechnet werden kann?

8. Welche Formel erhält man unter der Benutzung der Trapezregel für die Funktion $f(x)=x^2$ im Intervall [0,1] und n gleichbreiten Teilflächen?
Vereinfachen Sie Ihr Resultat mit der Tatsache, dass $1^2+2^2+3^2+\cdots+n^2=\frac{1}{6}(2n+1)(n+1)n$. Zeigen Sie, dass für $n\to\infty$ die Summe zum exakten Integralwert $\frac{1}{3}$ konvergiert.

9. Berechnen Sie die Näherungswerte für $\int_1^2 \frac{dx}{x^2}$ mit $h=\frac{1}{2}$ mit Untersummen und der Trapezregel.

10. Wie gross ist der maximale absolute Fehler mit der Trapezregel bei der Berechnung von $\int_2^5 \sin(x)\,dx$ mit $h=0.01$. Hinweis: Wenden Sie die Fehlerabschätzungsformel geeignet an.

11. Wir möchten $\int_1^2 f(x)\,dx$ berechnen, wobei *f(x)* mit der Wertetabelle gegeben ist:

x	1	$5/4$	$3/2$	$7/4$	2
$f(x)$	10	8	7	6	5

Berechnen Sie das Resultat mit der Trapezregel. Könnte die Rechnung auch mit Ober- und Untersummen durchgeführt werden?

Simpson-Regel

12. Die Fehlerabschätzungsformel der keplerschen Fassregel berücksichtigt nur eine Teilfläche. Bestimmen Sie nun eine allgemeine Formel zur Fehlerabschätzung, so dass das notwendige n für einen vorgebenden minimalen Fehler bestimmt werden kann.

13. Berechnen Sie $\int_0^1 \frac{dx}{1+x^2}$ mit der Simpson-Regel und einer Zerlegung an den Punkten x={0, 0.5, 1}. Vergleichen Sie das Resultat mit dem exakten Wert.

14. Betrachten Sie das Integral $\int_0^1 \sin(\frac{\pi x^2}{2})\,dx$. Wir möchten dieses Integral mit den nachfolgenden numerischen Verfahren mit einer Genauigkeit von besser als 10^{-3} lösen. Welche Intervallbreite h ist notwendig für:

a.) Trapezregel
b.) Simpson-Regel

15.Eine Funktion *f(x)* ist mit folgender Wertetabelle definiert:

x	1	$5/4$	$3/2$	$7/4$	2
$f(x)$	10	8	7	6	5

a.) Benutzen Sie die Simpson-Regel und die Zerlegung x=1, 1.5, 2 um eine Näherung für $\int_1^2 f(x)\,dx$ zu berechnen.

b.) Wiederholen Sie die Rechnung mit der Zerlegung x=1, 1.25, 1.5 1.75, 2.

16.Finden Sie einen Näherungswert für $\int_1^2 \frac{1}{x}\,dx$ mit der Simspon-Regel mit h=0.25. Geben Sie eine obere Schranke für den Fehler an.

Newton Cotes Verfahren

17.Wie gross ist der maximale Fehler einer Teilfläche für k=7, ausgedrückt als Ungleichung?

Programmieraufgaben

Trapezregel

18.Entwickeln Sie eine Funktion in C

```
double trapez_aequidistant(double *f,double a, double b, int n)
```

um $\int_a^b f(x)\,dx$ nach der Trapezregel mit n Teilflächen zu berechnen.

19.Testen Sie Ihre Funktion mit den folgenden Integralen:

a.) $\int_0^{\pi} \sin(x)\,dx$

b.) $\int_0^1 e^x dx$

c.) $\int_0^1 \arctan x\,dx$

20.Berechnen Sie die Zahl π mit Hilfe des Integrales der Form $c\int_a^b \frac{dx}{1+x^2}$.

21.Berechnen Sie einen Näherungswert für das Integral $\int_0^{0.8} \frac{\sin x}{x}\,dx$.

22. Berechnen Sie mit der Trapezregel die folgenden uneigentlichen Integrale, indem Sie die obere Grenze durch einen sehr grossen Wert nähern. Führen Sie die Rechnung ein zweites mal durch, unter vorgängiger Anwendung der vorgeschlagenen Variablensubstitution.

a.) $\int_0^\infty e^{-x^2}\,dx$ mit $x = -\ln t$

b.) $\int_1^\infty x^{-1} \sin x\,dx$ mit $x = t^{-1}$

c.) $\int_0^\infty \sin x^2 dx$ mit $x = \tan t$

Hinweis: a.) ist die Verteilungsfunktion der Normalverteilung, b.) ist der Integralsinus, c.) ist ein Fresnel-Integral.

Simpson-Regel

23. Erstellen Sie ein Programm das *mit möglichst wenig Aufwand* die

Rechtecksummen *R(n)*
MacLaurin-Summen *M(n)*
Trapezsummen *T(n)*
Simpson-Summen *S(n)*
Bode-Summen *B(n)*

für ein gegebenes *n* (Vielfaches von 4) gleichzeitig berechnet.

24. Erstellen Sie ein Programm, das die standardisierte Normalverteilung

$$P(t \le X) = \int_0^t \Theta(x)dx = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-\frac{x^2}{2}} dx$$

für t=0..3.99 mit einer Schrittweite von 0.01 berechnet und tabellarisch ausgibt.