Numerische Integration, Rechteckmethode

1. Bestimmen Sie für die Funktion $f(x) = e^x$ im Intervall [0,1]:

a.) Ober- und Untersummen für eine Zerlegung in 10 Teilintervalle.

$$h = \frac{b-a}{n} = \frac{1-0}{10} = \frac{1}{10}$$
$$Z = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$$
$$m_k = \inf\left(e^x : x_k \le x \le x_{k+1}\right) = e^{x_k} \qquad \text{(da monoton steigend)}$$
$$M_k = \sup\left(e^x : x_k \le x \le x_{k+1}\right) = e^{x_{k+1}}$$
$$U(f,Z) = \sum_{i=0}^{9} m_i(x_{i+1} - x_i) = \sum_{i=0}^{9} m_i h = 0.1\left(e^0 + e^{0.1} + e^{0.2} + e^{0.3} + e^{0.4} + e^{0.5} + e^{0.6} + e^{0.7} + e^{0.8} + e^{0.9}\right) \approx 1.63378$$
$$O(f,Z) = \sum_{i=0}^{9} M_i(x_{i+1} - x_i) = \sum_{i=0}^{9} M_i h = 0.1\left(e^0 + e^{0.1} + e^{0.2} + e^{0.3} + e^{0.4} + e^{0.5} + e^{0.6} + e^{0.7} + e^{0.8} + e^{0.9} + e^1\right) \approx 1.805628$$

b.) Den absoluten und relativen Fehler dieser Summen bezüglich des als exakt angenommenen Integralwertes e-1=1.718281828.

$U(f,Z)$:
$$\varepsilon = \tilde{y} - y = 1.63378 - e + 1 = -0.084502$$
$$\delta = \frac{\varepsilon}{y} = \frac{1.63378 - e + 1}{e - 1} = -0.0491781$$

$O(f,Z)$:
$$\varepsilon = \tilde{y} - y = 1.805628 - e + 1 = 0.0873462$$
$$\delta = \frac{\varepsilon}{y} = \frac{1.805628 - e + 1}{e - 1} = 0.0508334$$

c.) Wie gross müsste die Anzahl Teilintervalle n sein, damit der absolute Fehler kleiner als $3 \cdot 10^{-3}$ wird?
Hinweis: Verwenden Sie die Fehlerabschätzungsformel.

Aus dem vorher bestimmten absoluten Fehler für *n=10* wird die Konstante c für die Fehlerabschätzung bestimmt. Anschliessend wird *n* für $\varepsilon \le 3 \cdot 10^{-3}$ berechnet:

$$c = \frac{n|\varepsilon|}{\max|f'|\,(b-a)^2} = \frac{10 \cdot 0.0873462}{e^1 \cdot 1} = 0.3233$$
$$n \ge \frac{c(b-a)^2 \max|f'|}{|\varepsilon|} = \frac{0.3233 \cdot 1 \cdot e^1}{3 \cdot 10^{-3}} = 291.154 \quad \to n \ge 292$$

2. Führen Sie eine Berechnung des obigen Integralwertes mit 10, 100 und 1000 Teilintervallen unter EXCEL aus. Bestimmen Sie die jeweiligen Fehler der Zerlegung durch Vergleich mit dem exakten Integralwert.

   Lösung: Sollte kein Problem sein! (Download: ftp.krucker.ch)

Programmieraufgabe:

3. Entwerfen Sie aufgrund der Erkenntnisse in 1.) und den Ausführungen im Skript eine programmierte Berechnung für die Integration nach dem Rechteckverfahren.

   Lösung: (Download: ftp.krucker.ch)
   Eine etwas umfangreichere Implementierung des Verfahrens wird nachfolgend gezeigt. Sie erlaubt die Eingabe des Integrationsintervalles [a,b] und der zu benutzenden Zerlegung. Allerdings werden keine Plausibilitätskontrollen für die eingegebenen Grössen durchgeführt.

```
/* IAM Uebung #13: Numerische Integration nach der Rechteckmethode
   Das Programm berechnet naeherungsweise das bestimmte Integral e^x ueber den Bereich von [a,b] mit
   verschiedenen Zerlegungen.

   Die Rechnung wird durch Bilden der 'linken Rechtecksumme Sum(f(xi)*h)' und 'rechten Rechtecksumme
   Sum(f(x[i+1])*h)'. ,Links' bedeudet, dass fuer die Rechteckhoehe der linke Funktionswert relevant ist.
   'Rechts' besagt, dass der jeweils rechte Funktionswert fuer die Hoehe des Teilrechteckes relevant ist.

   Autor: Gerhard Krucker
   Datum: 13.5.1997
   Sprache: MS Visual-C V4.1 (WinNT Console Application) */

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

/* Aufzuintegrierende Funktion f(x)
   Hier: e^x, gemaess der Aufgabenstellung.
*/
double f(double x)
{ return exp(x);
}

/* Berechnen des bestimmten Integrales mit der Rechteckmethode ueber 'linke Rechtecksumme'.
   Dabei werden einzelne Rechtecke zu einer Gesamtfläche aufsummiert.
   Man ersieht leicht, dass diese Methode in der Regel nur für konstante Funktionen f(x)
   exakt rechnet.

  Parameter: a: untere Integrationsgrenze (Voraussetzung: b > a)
             b: obere Integrationsgrenze
             n: Zerlegung (Anzahl Teilflaechen fuer die Integration)
*/
double rectIntegralL(double a, double b,int n)
{ double h;      /* Aequidistante Maschenweite (Breite einer Teilflaeche) */
  double sumL;   /* Linke Rechtecksumme */
  double xi;     /* Aktuelles Argument im Schritt i *) */
  int i;

  sumL = 0.0;
  h = (b - a) / n;

  for (i = 0; i < n; i++)  /* Einzelne Teilrechtecke bilden und aufsummieren */
    { xi = a + i * h;
          sumL += f(xi) * h;
    }
  return sumL;
}
```

```
/* Berechnen des bestimmten Integrales mit der Rechteckmethode ueber 'rechte Rechtecksumme'.
   Dabei werden einzelne Rechtecke zu einer Gesamtfläche aufsummiert.
   Man ersieht leicht, dass diese Methode in der Regel nur für konstante Funktionen f(x)
   exakt rechnet.

  Parameter: a: untere Integrationsgrenze (Voraussetzung: b > a)
             b: obere Integrationsgrenze
             n: Zerlegung (Anzahl Teilflaechen fuer die Integration)
*/
double rectIntegralR(double a, double b,int n)
{ double h;                      /* Aequidistante Maschenweite (Breite einer Teilflaeche) */
  double sumR;           /* Rechte Rechtecksumme */
  double xiplus1;   /* Aktuelles Argument im Schritt i *) */
  int i;

  sumR = 0.0;
  h = (b - a) / n;

  for (i = 0; i < n; i++)  /* Einzelne Teilrechtecke bilden und aufsummieren */
    { xiplus1 = a + (i+1) * h;
         sumR += f(xiplus1) * h;
    }
  return sumR;
}


main()
{ double a,b;    /* Anfang und Ende des Integrationsintervalls */
  double iL, iR; /* Resulate der Integrationen */
  int n;         /* Zerlegung */
  int count=1;   /* Anzahl gueltig eingelesener Werte von der Tastatur */

  printf("Numerische Integration der Funktion e^x nach Rechteckmethode im Intervall [a,b].");

  while (count != 0)
  {  printf("\nEingabe der Intervallgrenzen und Zerlegung in a, b, n: \n(Eingabe: x x x = beenden)\n");
     count=scanf("%lf %lf %d",&a,&b, &n);

     if (count == 3)
     { iL=rectIntegralL(a,b,n);
       iR=rectIntegralR(a,b,n);
       printf("Integralnaeherung nach Rechteckmethode mit %d Teilflaechen:\n"
              "Linke rechtecksumme = %6.4f\nRechte Rechtecksumme = %6.4f\n",n,iL,iR);
     }
    else
     printf("Fehler bei der Eingabe: a, b sind Gleitkommazahlen, durch Leerschlag zu trennen.\n"
                   "                   n ist eine positive Ganzzahl >= 1\n x x x als Eingabe beendet das Programm\n");
  }
   return 0;
}
```