Numerische Integration

Manfred Hanke (Gruppe 5: Hanke / Nieva)

17. Jan 2007

1 Zielsetzung

Folgende Integral sollen numerisch gelöst werden:

$$I_a = \int_0^1 \sqrt{x}\,dx, \qquad I_b = \int_{10}^{11} \sqrt{x}\,dx, \qquad I_c = \int_0^{\pi/2} x^2(x^2-2)\sin x\,dx$$

2 Methoden

2.1 Trapezregel

$$\int_{x_0}^{x_N} y(x)\,dx \approx h\left(\frac{y_0}{2} + y_2 + \ldots + y_{N-1} + \frac{y_N}{2}\right) \qquad \text{mit } h = \frac{x_N - x_0}{N},\ y_i := y(x_i)$$

Es läßt sich zeigen, daß die Trapezregel das Integral in der Ordnung $\mathrm{O}(h^2)$ approximiert.

2.2 Simpson-Regel

$$\int_{x_0}^{x_{2N}} y(x)\,dx \approx h\left(\frac{y_0}{3} + \frac{4y_1}{3} + \frac{2y_2}{3} + \frac{4y_3}{3} + \ldots + \frac{2y_{2N-2}}{3} + \frac{4y_{2N-1}}{3} + \frac{y_{2N}}{3}\right)$$

$$\text{mit } h = \frac{x_{2N} - x_0}{2N},\ y_i := y(x_i)$$

Es läßt sich zeigen, daß die Simpson-Regel das Integral in der Ordnung $\mathrm{O}(h^4)$ approximiert.

2.3 Monte-Carlo-Verfahren

Beim Monte-Carlo-Verfahren werden N zufällige Punkte $(x_r,\ y_r)$ im vorgegebenen Bereich $B = [x_1,\ x_2] \times [y_1,\ y_2]$ generiert und die Anzahl N_{below} der Punkte bestimmt, für die $y_r \le y(x_r)$ gilt. Die Fläche unterhalb des Graphens $y(x)$ im Bereich B beträgt dann $A = \frac{N_{\text{below}}}{N} \cdot (x_2 - x_1) \cdot (y_2 - y_1)$ und das Integral ist

$$\int_{x_1}^{x_2} y(x)\,dx = y_1 \cdot (x_2 - x_1) + A = \left(y_1 + \frac{N_{\text{below}}}{N}\right) \cdot (x_2 - x_1)\ ,$$

falls $y_1 \le y(x) \le y_2$ für alle $x_1 \le x \le x_2$ (bzw. $x_1 \ge x \ge x_2$) gilt. (x_1 und x_2 können beliebig geordnet sein, das Integral wird sowohl nach der Formel wie nach Definition ggf. negativ. Für y_1 und y_2 muß hingegen $y_1 < y_2$ gelten.)

3 Durchführung

3.1 Trapezregel

Die Integration nach der Trapezregel wird in der `FUNCTION Trapez(x0, xN, y, epsInt)` umgesetzt. `y` gibt dabei die `REAL`-wertige Integrandenfunktion an, `x0` und `xN` sind die Grenzen und `epsInt` ist die gewünschte Genauigkeit. Die Integration wird so lange verfeinert, bis das Verhältnis aufeinanderfolgender Näherungswerte sich nicht um mehr als `epsInt` von 1 unterscheidet. (Dabei bleibt zu hoffen, daßdie Näherungswerte dann auch hinreichend nahe am wahren Wert liegen, der ja i.a. unbekannt ist.)
Das Verfahren beginnt mit $N = 4$ Schritten. Nach einigen Initialisierungen wird in einer Schleife über $1 \le i < N$ die Summe $\mathtt{S} = \frac{y(x_0)+y(x_N)}{2} + \sum_{i=1}^{N-1} y\big(x_0 + i\,\frac{x_N - x_0}{N}\big)$ berechnet, aus der sich dann direkt der Integralwert ergibt.
Dieser wird mit dem vorherigen Ergebnis (das am Anfang willkrlich auf 0 gesetzt wurde) verglichen. Bei zu großer Abweichung $|\frac{\text{alterWert}}{\text{neuer Wert}} - 1| > \epsilon$ wird N verdoppelt und mit der Berechnung von vorne begonnen. Ansonsten ist das Ergebnis der Rückgabewert der Funktion.

3.2 Simpson-Regel

Die Integration nach der Simpson-Regel erfolgt analog zur Trapezregel in der `FUNCTION Simps(x0, x2N, y, epsInt)`. Mit `S` wird dieses Mal folgende Summe berechnet:
$\mathtt{S} = y(x_0) + 4y\big(x_0 + \frac{x_{2N}-x_0}{2N}\big) + y(x_{2N}) + \sum_{i=1}^{N-1} 2y\big(x_0 + 2i\,\frac{x_{2N}-x_0}{2N}\big) + 4y\Big(x_0 + (2i+1)\,\frac{x_{2N}-x_0}{2N}\Big)$
Die Prozedur beginnt mit $2N = 4$ und verdoppelt N (wie bei der Trapezregel) solange, bis sich der Integral-Näherungswert stabilisiert.

3.3 Monte-Carlo-Verfahren

Die Monte-Carlo-Methode wird in der `FUNCTION MCarlo(x1, x2, y, y1, y2, epsInt)` wie in 2.3 beschrieben umgesetzt. Da es hier bei `y1` und `y2` (im Gegensatz zu `x1` und `x2`; sowohl hier als auch bei den anderen Funktionen) auf die Anordnung $y_1 < y_2$ ankommt, wird das Minimum und Maximum von `y1` und `y2` eigens bestimmt und `ymin` und `yax` zugewiesen. In der Schleife werden mit dem Zufallsgenerator `ran2` (nach Skalierung) Punkte im angegebenen Bereich erzeugt. Die Zählvariablen `i` und `ibelow` sind als `REAL` definiert, um zunächst einen größeren Bereich zur Verfügung zu haben, *was langfristig allerdings auch problematisch ist.* Man beachte, daß es bei diesem Verfahren nicht notwendig ist, bei einer Vergrößerung des Wertes N (zur Erhöhung der Genauigkeit) die Werte zurückzusetzen!

3.4 Hauptprogramm

Während als Funktion $y(x) = \sqrt{x}$ für die Integrale I_a und I_b die `INTRINSIC FUNCTION SQRT` (nach Deklaration als solcher) verwendet werden kann, muß $y(x) = x^2(x^2-2)\sin x$ (Integrand von I_c) erst als `EXTERNAL FUNCTION int_c(x)` – in der nur die Wertzuweisung erfolgt – definiert werden.
Im eigentlichen Hauptprogramm werden jeweils die 3 Methoden für die 3 gesuchten Integrale aufgerufen, wobei bei `MCarlo` auch geeignete einschließende y-Werte angegeben werden müssen. Es ist natürlich $0 \le \sqrt{x} \le 1$ für $0 \le x \le 1$ und z.B. $3 < \sqrt{x} < 3.4$ für $10 \le x \le 11$. Weiterhin kann man sich davon überzeugen (siehe Abb. 1), daß $-0.9 < x^2(x^2-2)\sin x < 1.2$ für $0 \le x \le \frac{\pi}{2}$ gilt.

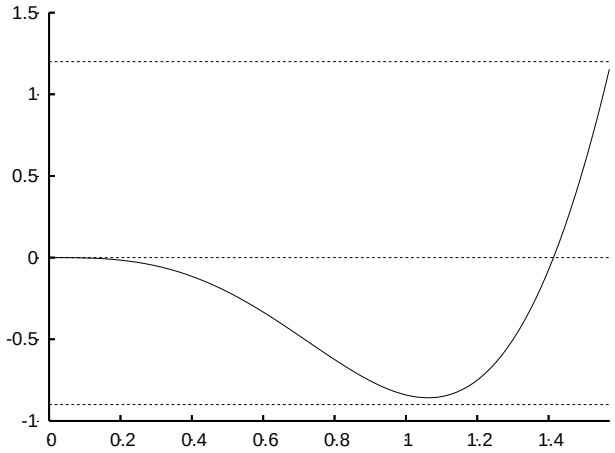

Abbildung 1: `int_c`$(x) = x^2(x^2-2)\sin x$

4 Fortran-Programm

```
      REAL FUNCTION Trapez(x0, xN, y, epsInt)
      REAl x0, xN, y, epsInt
      REAL dx, x, S, Trap0, Trap1
      INTEGER N, i
      N = 4
      Trap0 = 0.
100   dx = (xN-x0)/N
      S = (y(x0)+y(xN))*0.5
      i = 1
200   S = S + y(x0 + i*dx)
      i = i + 1
      IF(i .LT. N) GOTO 200
      Trap1 = dx * S
c     WRITE(*,*) N, Trap1, Trap1-Trap0
      IF(ABS(Trap0/Trap1-1) .GT. epsInt) THEN
        Trap0 = Trap1
        N = N*2
        GOTO 100
      ENDIF
      Trapez = Trap1
      END
c
c
      REAL FUNCTION Simps(x0, x2N, y, epsInt)
      REAl x0, x2N, y, epsInt
      REAL dx, x, S, Simps0, Simps1
      N = 2
      Simps0 = 0.
100   dx = (x2N-x0)/(2*N)
      S = y(x0) + 4*y(x0+dx) + y(x2N)
      i = 1
200   x = x0 + 2*i*dx
      S = S + 2*y(x) + 4*y(x+dx)
      i = i + 1
      IF(i .LT. N) GOTO 200
      Simps1 = dx * S/3
c     WRITE(*,*) 2*N, Simps1, Simps1-Simps0
      IF(ABS(Simps0/Simps1-1) .GT. epsInt) THEN
        Simps0 = Simps1
        N = N*2
        GOTO 100
      ENDIF
      Simps = Simps1
      END
c
c
      INCLUDE "ran2.for"
c
      REAL FUNCTION MCarlo(x1, x2, y, y1, y2, epsInt)
      IMPLICIT NONE
      REAl x1, x2, y, y1, y2, epsInt
      INTEGER idum
      REAL ymin, ymax, deltax, deltay
      REAL i, ibelow, N, ran2, xr, yr, MC0, MC1
      idum = -1
      IF(y2 .GE. y1) THEN
        ymin = y1
        ymax = y2
```

```
      ELSE
         ymin = y2
         ymax = y1
      ENDIF
      deltax = x2-x1
      deltay = ymax-ymin
      i = 0.
      ibelow = 0.
      MC0 = 0.
      N = 100.

100   i = i+1
      xr = x1 + ran2(idum)*deltax
      yr = ymin + ran2(idum)*deltay
      IF(yr .LE. y(xr))  ibelow = ibelow+1
      IF(i .LT. N) GOTO 100
      MC1 = (ymin + ibelow/i*deltay) * deltax
c     WRITE(*,*) i, ibelow, MC1
      IF(ABS(MC0/MC1-1) .GE. epsInt) THEN
         MC0 = MC1
         N = 2*N
         GOTO 100
      ENDIF
      MCarlo = MC1
      END
c
c
      REAL FUNCTION int_c(x)
      REAL x
      int_c = x*x*(x*x-2)*sin(x)
      END
c
      PROGRAM numInt
      IMPLICIT NONE
      INTRINSIC SQRT
      EXTERNAL int_c
      REAL I, Trapez, Simps, MCarlo, ran2, r
c
      I = Trapez(0., 1., SQRT, 1e-4)
      WRITE(*,'("Ia=",F7.4)') I
      I = Simps(0., 1., SQRT, 1e-4)
      WRITE(*,'("Ia=",F7.4)') I
      I = MCarlo(0., 1., SQRT, 0., 1., 1e-3)
      WRITE(*,'("Ia=",F6.3)') I
      WRITE(*,*)
      I = Trapez(10., 11., SQRT, 1e-4)
      WRITE(*,'("Ib=",F7.4)') I
      I = Simps(10., 11., SQRT, 1e-4)
      WRITE(*,'("Ib=",F7.4)') I
      I = MCarlo(10., 11., SQRT, 3., 3.4, 1e-3)
      WRITE(*,'("Ib=",F6.3)') I
      WRITE(*,*)
      I = Trapez(0., 0.5*3.14159265, int_c, 1e-4)
      WRITE(*,'("Ic=",F7.4)') I
      I = Simps(0., 0.5*3.14159265, int_c, 1e-4)
      WRITE(*,'("Ic=",F7.4)') I
      I = MCarlo(0., 0.5*3.14159265, int_c, -0.9, 1.2, 1e-4)
      WRITE(*,'("Ic=",F6.3)') I
      END
```

5 Ergebnisse und Auswertung

Mit den Stammfunktionen

$$\int \sqrt{x}\,dx = \frac{2}{3}x^{\frac{3}{2}}, \qquad \int x^2(x^2-2)\sin x\,dx = (-x^4+14x^2-28)\cos x + (4x^3-28x)\sin x$$

ergeben sich analytisch folgende Werte für die gesuchten Integrale:

$$I_a = \frac{2}{3} \approx 0.66666667, \qquad I_b = \frac{22\sqrt{11}-20\sqrt{10}}{3} \approx 3.24006406,$$

$$I_c = \frac{\pi^3}{2} - 14\pi + 28 \approx -0.47915882$$

Das Programm liefert dafür gute Näherungswerte:

```
                +-                    +-
Ia= 0.6666      |     Ib= 3.2401      |     Ic=-0.4792
Ia= 0.6666      |     Ib= 3.2401      |     Ic=-0.4792
Ia= 0.664       |     Ib= 3.241       |     Ic=-0.478
                -+                    -+
```

Die Simpson-Regel konvergiert – wie wegen $O(h^4)$ erwartet – schneller als die Trapezregel. Das Monte-Carlo-Verfahren ist in diesen Fällen am langsamsten und ungenauesten. Die relative Konvergenz bei $I_b = \int_{10}^{11} 3\,dx + \int_{10}^{11} \sqrt{x} - 3\,dx$ ist besser als bei $I_a = \int_0^1 \sqrt{x}\,dx$, weil der stochastische Anteil kleiner ist.

```
-----
| Mh
```