BERGISCHE UNIVERSITÄT WUPPERTAL
FAKULTÄT FÜR MATHEMATIK
UND NATURWISSENSCHAFTEN

WISSENSCHAFTLICHES RECHNEN /
SOFTWARETECHNOLOGIE
Dr. W. Hofschuster

Verifikationsnumerik I

WS 2016/17

Übungsblatt 8

Aufgabe 25: *Automatische Differentiation (AD)*

Als Differentiationsarithmetik (in der einfachsten Form) bezeichnet man eine Arithmetik für geordnete Paare der Form

$$U = (u, u') \quad \text{mit} \quad u, u' \in I\!R.$$

In der ersten Komponente von U steht der Funktionswert $u = u(x)$, in der zweiten der Wert der Ableitung $u' = u'(x)$. Verwendet man zur Berechnung der zweiten Komponente jeweils die entsprechende Differentiationsregel, so ergeben sich die folgenden Verknüpfungsvorschriften:

$$\begin{array}{lclcll}
U + V & = & (u, u') + (v, v') & = & (u + v, & u' + v') \\
U - V & = & (u, u') - (v, v') & = & (u - v, & u' - v') \\
U \cdot V & = & (u, u') \cdot (v, v') & = & (u \cdot v, & u \cdot v' + u' \cdot v) \\
U / V & = & (u, u') / (v, v') & = & (u / v, & (u' - u \cdot v' / v) / v), \quad v \neq 0, \\
f(U) & = & f((u, u')) & = & (f(u), & f'(u) \cdot u')
\end{array}$$

Für die Funktion $f(x) = x$ (unabhängige Variable) bzw. die Funktion $f(x) \equiv c$ (Konstante) ergeben sich mit $f'(x) = 1$ bzw. $f'(x) = 0$ die Ausgangspaare

$$X = (x, 1) \quad \text{bzw.} \quad C = (c, 0).$$

Berechnen Sie von Hand mit Hilfe der obigen Regeln für

$$f(x) = e^{-3x} + \frac{x + 3}{(x - 2)(x + 4)}$$

das Tupel $(f(1), f'(1))$.

Aufgabe 26: *Berechnung höherer Ableitungen*

Schreiben Sie ein C–XSC Programm, mit dem Sie den Wert der zweiten Ableitung der Funktion

$$f(x) = \frac{\ln\left(x^2 + \arctan\left(\sqrt{2 + \sin(\,x\,)}\right)\right)}{1 + \sinh(\,x\,)^2}$$

an einer einzulesenden Stelle x_0 einschließen können. Verwenden Sie hierzu sowohl die symbolische als auch die automatische Differentiation und vergleichen Sie den notwendigen Programmieraufwand. Die benötigte zweite Ableitung ist im folgenden als C-Quellcode angegeben (von Maple generiert mit Hilfe der Funktionen `diff`, `simplify`, `readlib(C)` und `C`).

```
      s1 = -1/pow(cosh(x),4.0)/4;
      s5 = 40.0*sinh(x)*cosh(x)*cos(x)*sin(x)*atan(sqrt(2.0+sin(x)))-8.0*x*pow(
cos(x),3.0)*pow(cosh(x),2.0)+56.0*sinh(x)*cosh(x)*cos(x)*atan(sqrt(2.0+sin(x)))
+56.0*sinh(x)*cosh(x)*cos(x)*x*x+14.0*sin(x)*atan(sqrt(2.0+sin(x)))*pow(cosh(x)
,2.0)+56.0*x*cos(x)*pow(cosh(x),2.0)-3.0*x*x*pow(cos(x),2.0)*pow(cosh(x),2.0)
-208.0*sqrt(2.0+sin(x))*atan(sqrt(2.0+sin(x)))*pow(cosh(x),2.0)+624.0*log(x*x+
atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*pow(x,4.0)+624.0*log(x*x+atan(sqrt(2.0
+sin(x))))*sqrt(2.0+sin(x))*pow(atan(sqrt(2.0+sin(x))),2.0)+208.0*sqrt(2.0+sin(
x))*x*x*pow(cosh(x),2.0)+10.0*x*x*pow(cosh(x),2.0)+sqrt(2.0+sin(x))*pow(cos(x),
2.0)*pow(cosh(x),2.0)-3.0*atan(sqrt(2.0+sin(x)))*pow(cos(x),2.0)*pow(cosh(x),
2.0);
      s6 = s5-32.0*cosh(x)*sqrt(2.0+sin(x))*x*atan(sqrt(2.0+sin(x)))*sinh(x)*
sin(x)*pow(cos(x),2.0)+528.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*
sin(x)*pow(x,4.0)-64.0*sqrt(2.0+sin(x))*x*x*pow(cos(x),2.0)*pow(cosh(x),2.0)+
176.0*sqrt(2.0+sin(x))*x*x*sin(x)*pow(cosh(x),2.0)-176.0*sqrt(2.0+sin(x))*sin(x
)*atan(sqrt(2.0+sin(x)))*pow(cosh(x),2.0)+1056.0*log(x*x+atan(sqrt(2.0+sin(x)))
)*sqrt(2.0+sin(x))*x*x*atan(sqrt(2.0+sin(x)))*sin(x)-256.0*sinh(x)*cosh(x)*sqrt
(2.0+sin(x))*x*x*x*pow(cos(x),2.0);
      s4 = s6+1248.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*x*x*atan(
sqrt(2.0+sin(x)))-256.0*sinh(x)*cosh(x)*sqrt(2.0+sin(x))*x*atan(sqrt(2.0+sin(x)
))*pow(cos(x),2.0)-8.0*cosh(x)*pow(cos(x),3.0)*x*x*sinh(x)+x*x*sin(x)*pow(cos(x
),2.0)*pow(cosh(x),2.0)-384.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*
x*x*atan(sqrt(2.0+sin(x)))*pow(cos(x),2.0)+atan(sqrt(2.0+sin(x)))*sin(x)*pow(
cos(x),2.0)*pow(cosh(x),2.0)+8.0*sqrt(2.0+sin(x))*atan(sqrt(2.0+sin(x)))*sin(x)
*pow(cos(x),2.0)*pow(cosh(x),2.0)+40.0*x*sin(x)*cos(x)*pow(cosh(x),2.0);
      s6 = -8.0*sinh(x)*cosh(x)*pow(cos(x),3.0)*atan(sqrt(2.0+sin(x)))+10.0*
atan(sqrt(2.0+sin(x)))*pow(cosh(x),2.0)-192.0*log(x*x+atan(sqrt(2.0+sin(x))))*
sqrt(2.0+sin(x))*pow(x,4.0)*pow(cos(x),2.0)+64.0*sqrt(2.0+sin(x))*atan(sqrt(2.0
+sin(x)))*pow(cos(x),2.0)*pow(cosh(x),2.0)-24.0*log(x*x+atan(sqrt(2.0+sin(x))))
*sqrt(2.0+sin(x))*pow(x,4.0)*sin(x)*pow(cos(x),2.0)-8.0*sqrt(2.0+sin(x))*x*x*
sin(x)*pow(cos(x),2.0)*pow(cosh(x),2.0)+528.0*log(x*x+atan(sqrt(2.0+sin(x))))*
sqrt(2.0+sin(x))*sin(x)*pow(atan(sqrt(2.0+sin(x))),2.0);
      s5 = s6-24.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*pow(atan(
sqrt(2.0+sin(x))),2.0)*sin(x)*pow(cos(x),2.0)-192.0*log(x*x+atan(sqrt(2.0+sin(x
))))*sqrt(2.0+sin(x))*pow(atan(sqrt(2.0+sin(x))),2.0)*pow(cos(x),2.0)-48.0*log(
x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*x*x*atan(sqrt(2.0+sin(x)))*sin(x)*
pow(cos(x),2.0)-32.0*cosh(x)*sqrt(2.0+sin(x))*x*x*x*sinh(x)*sin(x)*pow(cos(x),
2.0)+40.0*sinh(x)*cosh(x)*cos(x)*sin(x)*x*x+s4+256.0*log(x*x+atan(sqrt(2.0+sin(
x))))*sqrt(2.0+sin(x))*x*x*atan(sqrt(2.0+sin(x)))*pow(cos(x),2.0)*pow(cosh(x),
2.0)+128.0*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+sin(x))*
pow(x,4.0)*pow(cos(x),2.0);
      s6 = s5+16.0*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+
sin(x))*pow(x,4.0)*sin(x)*pow(cos(x),2.0)+14.0*sin(x)*x*x*pow(cosh(x),2.0)+
704.0*sinh(x)*cosh(x)*sqrt(2.0+sin(x))*x*x*x*sin(x)+32.0*log(x*x+atan(sqrt(2.0+
sin(x))))*sqrt(2.0+sin(x))*x*x*atan(sqrt(2.0+sin(x)))*sin(x)*pow(cos(x),2.0)*
pow(cosh(x),2.0)+704.0*sinh(x)*cosh(x)*sqrt(2.0+sin(x))*x*sin(x)*atan(sqrt(2.0+
sin(x)))+128.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*pow(atan(sqrt(
2.0+sin(x))),2.0)*pow(cos(x),2.0)*pow(cosh(x),2.0)-352.0*log(x*x+atan(sqrt(2.0+
sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+sin(x))*sin(x)*pow(x,4.0);
      s7 = s6-704.0*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+
sin(x))*sin(x)*x*x*atan(sqrt(2.0+sin(x)))-352.0*log(x*x+atan(sqrt(2.0+sin(x))))
*pow(cosh(x),2.0)*sqrt(2.0+sin(x))*pow(atan(sqrt(2.0+sin(x))),2.0)*sin(x)-416.0
*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+sin(x))*pow(atan(
sqrt(2.0+sin(x))),2.0);
```

```
      s3 = s7-832.0*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+
sin(x))*x*x*atan(sqrt(2.0+sin(x)))+832.0*sinh(x)*cosh(x)*sqrt(2.0+sin(x))*x*x*x
-416.0*log(x*x+atan(sqrt(2.0+sin(x))))*pow(cosh(x),2.0)*sqrt(2.0+sin(x))*pow(x,
4.0)+16.0*log(x*x+atan(sqrt(2.0+sin(x))))*sqrt(2.0+sin(x))*pow(atan(sqrt(2.0+
sin(x))),2.0)*sin(x)*pow(cos(x),2.0)*pow(cosh(x),2.0)+832.0*sinh(x)*cosh(x)*
sqrt(2.0+sin(x))*x*atan(sqrt(2.0+sin(x)));
      s4 = 1/(sqrt(pow(2.0+sin(x),3.0)))/(12.0*sin(x)*atan(sqrt(2.0+sin(x)))*
x*x+10.0*pow(x,4.0)+6.0*sin(x)*pow(x,4.0)+10.0*pow(atan(sqrt(2.0+sin(x))),2.0)+
20.0*atan(sqrt(2.0+sin(x)))*x*x-2.0*x*x*atan(sqrt(2.0+sin(x)))*pow(cos(x),2.0)-
pow(x,4.0)*pow(cos(x),2.0)-pow(atan(sqrt(2.0+sin(x))),2.0)*pow(cos(x),2.0)+6.0*
sin(x)*pow(atan(sqrt(2.0+sin(x))),2.0));
      s2 = s3*s4;
      t0 = s1*s2;
```

Was müssen Sie tun, wenn Sie die gegebene Funktion modifizieren möchten? Untersuchen Sie z.B. die Funktion

$$g(x) = \frac{\ln\left(x^2 + \arctan\left(\sqrt{2+\sin(\,x\,)}\right)\right)}{(\,x+1\,)^2 + \sinh(\,x\,)^2}.$$

Aufgabe 27: *Taylorkoeffizienten*

Zeigen Sie mindestens eine der folgenden Rekursionsformeln zur Berechnung von Taylorkoeffizienten ($k \geq 1$):

a) $$(u^a)_k = \frac{1}{ku} \cdot \sum_{j=0}^{k-1} (a(k-j)-j)(u)_{k-j}(u^a)_j\ , \quad a \in I\!R \text{ konstant}$$

b) $$(\sin u)_k = \frac{1}{k} \cdot \left(\sum_{j=0}^{k-1} (j+1)(\cos u)_{k-j-1}(u)_{j+1} \right)$$

c) $$(\arctan u)_k = \frac{1}{1+u^2} \cdot \left((u)_k - \frac{1}{k} \sum_{j=1}^{k-1} j(\arctan u)_j (1+u^2)_{k-j} \right).$$

Aufgabe 28: *Implementierung und Anwendung einer Taylorarithmetik*

Mittels einer Intervall-Taylorarithmetik können Einschließungen für Taylorkoeffizienten

$$(f)_k := \frac{f^{(k)}(t_0)}{k!}$$

an der Stelle t_0 berechnet werden. Eine Implementierung in C-XSC finden Sie in den auf der Webseite zur Vorlesung bereitgestellten Dateien `itaylor.hpp` und `itaylor.cpp`. Ein kleines Beispiel ist in der Datei `itaylor_ex1.cpp` enthalten.

a) Untersuchen Sie die vorhandene Implementierung. Welche Methoden und Attribute gehören zur Klasse `itaylor`? Warum wird bei der Implementierung der Operatoren `*` und `/` das in C-XSC vorhandene exakte Skalarprodukt (Datentyp `idotprecision`, Funktion `accumulate()`) eingesetzt?
Was ist bei dieser Implementierung gut gelöst? Gibt es Verbesserungsmöglichkeiten?

b) Compilieren Sie diese Taylorarithmetik und testen Sie das Beispielprogramm `itaylor_ex1.cpp` mit verschiedenen Eingaben.

c) Berechnen Sie Einschließungen für die Taylorkoeffizienten des Polynoms

$$f(x) = 3x^6 + 2x^5 - x^4 + 4x^3 - 2x^2 + 5x + 1$$

nacheinander an den Stellen $t_0 = 0$, $t_0 = 1$ und $t_0 = 5$.

d) Schreiben Sie ein Programm, mit dem Sie eine Einschließung des Wertes der 10. Ableitung der Funktion

$$f(x) = \frac{\ln\left(x^2 + \arctan\left(\sqrt{2 + \sin(\,x\,)}\right)\right)}{1 + \sinh(\,x\,)^2}$$

an einer einzugebenen Stelle x_0 berechnen können.

Aufgabe 29: *Einschließung bestimmter Integrale*

a) Schreiben Sie ein C–XSC Programm, mit dem Einschließungen für bestimmte Integrale unter Verwendung der Simpsonregel berechnet werden können. Verwenden Sie dazu

$$\begin{aligned}\int_a^b f(x)dx \quad &\in \quad \frac{h}{3}\sum_{i=0}^{n} s_i \cdot f(x_i) \\ &\quad -\frac{h^5}{90}(f^{(4)}([x_0, x_2]) + f^{(4)}([x_2, x_4]) + \ldots + f^{(4)}([x_{n-2}, x_n]))\end{aligned}$$

mit: n gerade, $h := (b-a)/n$, $x_i := a + i \cdot h$, $i = 0(1)n$, $s_0 = s_n = 1$, $s_i = 4$, falls i ungerade und $s_i = 2$ sonst. Die auftretenden Ableitungen lassen sich mit Hilfe der Intervall-Taylorarithmetik berechnen.

Verdoppeln Sie n so lange, bis eine gewünschte Einschlußgenauigkeit erreicht wird. Falls dies für $n \geq 4096$ noch nicht erreicht sein sollte, so soll eine Fehlermeldung ausgegeben werden.

b) Implementieren Sie zum Vergleich die sogenannte „Weddle-Regel" (siehe J. Stoer, Numerische Mathematik 1)

$$\begin{aligned}\int_a^b f(x)dx \quad &= \quad \frac{h}{140}\Big(41f(x_0) + 216f(x_1) + 27f(x_2) + 272f(x_3) \\ &\quad +27f(x_4) + 216f(x_5) + 41f(x_6)\Big) \\ &\quad -\frac{9h^9}{1400}f^{(8)}(\xi)\end{aligned}$$

mit $\xi \in [a,b]$, $h := \frac{b-a}{6}$ und $x_i := a + i \cdot h$, $i = 0(1)n$. Leiten Sie hierbei zunächst eine Summenformel für eine äquidistante Unterteilung des Gesamtintegrationsbereichs (siehe Simpsonregel) her.

c) Berechnen Sie mit Hilfe von beiden Regeln Einschließungen der Werte der Integrale

$$\int_0^2 e^{-3x} \sin 2x \, dx \qquad \text{und} \qquad \int_1^2 \sin 2x \, \cos 3x \, dx.$$

d) Bestimmen Sie zum Vergleich Einschließungen dieser Integralwerte mit Hilfe der Stammfunktionen.