```
> restart:
```

Aufgabe 7

Numerische Differentiation

Es soll die erste Ableitung der Funktion $f := x \to \sin(x)\ln(x)$ an der Stelle $x_0 = 0.5$ mit der zentralen Differenzenformel

```
> Dfz:=(f(x[0]+h)-f(x[0]-h))/(2*h);
```

$$Dfz := \frac{f(x_0+h) - f(x_0-h)}{2h} \quad (1)$$

und der Differenzenformel

```
> Df3 := 1/6*1/h*(-9*f(x[0]+2*h)+2*f(x[0]+3*h)-11*f(x[0])+18*f(x[0]
  +h));
```

$$Df3 := \frac{-9f(x_0+2h) + 2f(x_0+3h) - 11f(x_0) + 18f(x_0+h)}{6h} \quad (2)$$

bei 40 signifikanten Dezimalstellen berechnet werden.

a) Berechnen Sie die logarithmierten absoluten Fehler von `Dfz` und `Df3` bei den Schrittweiten $h_i = 10^{-i}, i = 1..10$. Nutzen Sie seq für die Generierung der Punktepaare $[i, \log10(|D(f)(x_0) - Dfz|)]$ und $[i, \log10(|D(f)(x_0) - Df3|)]$. Berechnen Sie die linearen Ausgleichsgeraden für diese Punktefolgen mit LeastSquares aus dem Paket CurveFitting und ermitteln Sie damit die Fehlerordnungen der Differenzenformeln. Plotten Sie die Geraden in ein Schaubild.

```
> Digits:=40;
```

$$Digits := 40 \quad (3)$$

```
> f := x -> sin(x) * ln(x);
```

$$f := x \mapsto \sin(x)\cdot\ln(x) \quad (4)$$

```
> x_0 := 0.5;
```

$$x\_0 := 0.5 \quad (5)$$

```
> f_strich := unapply(diff(f(x), x), x);            # Tipp:
                                                    # Du
  benötigst die Ableitung nur an der Stelle x[0]
  x[0] := 0.5;                                      # Mit dem D-
  Operator kannst Du die Ableitung
  D(f)(x[0]);                                       # direkt an
  einer bestimmten Stelle auswerten;
```

$$f\_strich := x \mapsto \cos(x) \cdot \ln(x) + \frac{\sin(x)}{x}$$

$$x_0 := 0.5$$

$$0.3505571987255204440647627306602388807476 \quad \textbf{(6)}$$

```
> error_dfz := seq([i, log10(abs(f_strich(x_0) - evalf(subs([f = f
  (x), x[0] = x_0, h=10**(-i)], Dfz))))], i=1..10);
```

$error\_dfz := [1, -2.064274749209769717810313823033664576421], [2, -4.068127538171408198110071128330010742225], [3, -6.068165463315968997585185315986050125753], [4, -8.068165842508378893325967685926370389085], [5, -10.06816584630029708992055981221489722549], [6, -12.06816584633821627129624130205565794481], [7, -14.06816584633859546309907540961563173440], [8, -16.06816584633859923416496719745934293175], [9, -18.06816584633859283618600739250888044741], [10, -20.06816584640772040395321570632197114024]$ **(7)**

```
> error_df3 := seq([i, log10(abs(f_strich(x_0) - evalf(subs([f = f
  (x), x[0] = x_0, h=10**(-i)], Df3))))], i=1..10);
```

$error\_df3 := [1, -2.723708794485635684589324719028324500003], [2, -5.479259237956753280937589557509981271282], [3, -8.450207830675559586486817156908063701045], [4, -11.44724592658422807642670942656827792094], [5, -14.44694915374597468388028673811200869432], [6, -17.44691947062249049385563898096103208119], [7, -20.44691650225171585820727714538681388312], [8, -23.44691620590001795379472248154245635555], [9, -26.44692309234115511280706946794657967345], [10, -29.42468289634675496009950671177141969638]$ **(8)**

```
> h := 10.0^(-i);      # Tipp. Maple ersetzt in den verwendeten
  Ausdrücken
                       # automatisch die Werte für h und x[0] durch
  die entsprechenden Werte
                       # wenn h und x[0] definiert werden.
                       # subs für f, h und x[0] kannst Du einsparen.
```

$$h := 10.0^{-i} \tag{9}$$

```
> error_dfz := seq([i, log10(abs(D(f)(x[0]) - Dfz))], i=1..10);
  # viel kürzer und übersichtlicher.
  error_df3 := seq([i, log10(abs(D(f)(x[0]) - Df3))], i=1..10);
```

$$\begin{aligned}
error_dfz := {} & [1, -2.06427474920976971781031382303366457639 6], [2,\\
& -4.0681275381714081981100711283300107676 28], [3,\\
& -6.0681654633159689975851853159859993156 86], [4,\\
& -8.0681658425083788933259676859009653329 92], [5,\\
& -10.068165846300297089920559786809840910 75], [6,\\
& -12.068165846338216271296292112168291858 38], [7,\\
& -14.068165846338595463124480465932610700 18], [8,\\
& -16.068165846338599259570023514438530057 30], [9,\\
& -18.068165846338592836186007392508880447 41], [10,\\
& -20.068165846356910291314142784001535553 75]
\end{aligned}$$

$$\begin{aligned}
error_df3 := {} & [1, -2.7237087944856356845893247190283244998 42], [2,\\
& -5.4792592379567532809375895575099825805 96], [3,\\
& -8.4502078306755595864868171568835718202 14], [4,\\
& -11.447245926584228076426709142775839943 57], [5,\\
& -14.446949153745974683881097035385263533 01], [6,\\
& -17.446919470622490497906824138906373626 68], [7,\\
& -20.446916502251675346596131813572458402 81], [8,\\
& -23.446916206305133825456424152638167715 07], [9,\\
& -26.446914989970745212396965619172933275 57], [10,\\
& -29.322290940556435709064813730560417118 88]
\end{aligned} \tag{10}$$

```
> g_error_dfz := unapply(CurveFitting:-LeastSquares([error_dfz],
  x), x);
```

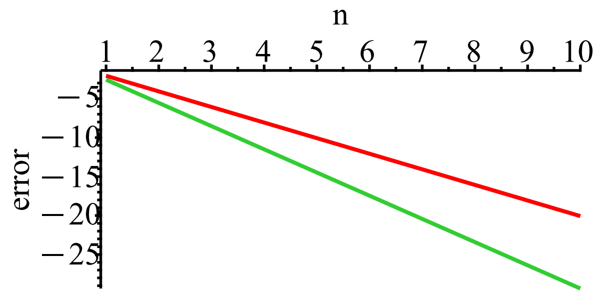
$$\begin{aligned}
g_error_dfz := x \mapsto {} & -0.066596535183797774198921862388400708366 57\\
& - 2.0002138785341592595682513706404785897 62 \cdot x
\end{aligned} \tag{11}$$

```
> g_error_df3 := unapply(CurveFitting:-LeastSquares([error_df3],
  x), x);
```

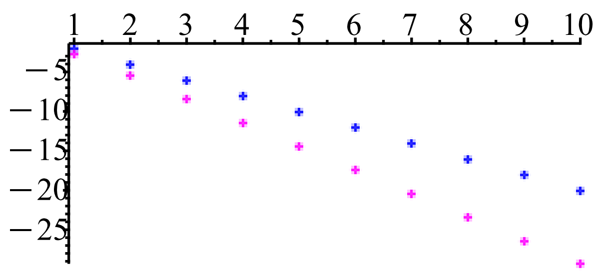
$$\begin{aligned}
g_error_df3 := x \mapsto {} & 0.405712211376599228288299081027114527706 1\\
& - 2.9766263848531022579386488704855359598 81 \cdot x
\end{aligned} \tag{12}$$

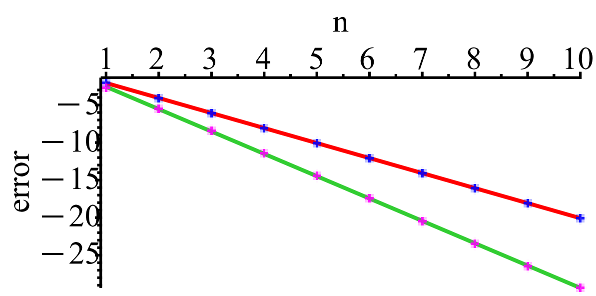
```
> p1 := plot([g_error_dfz(x), g_error_df3(x)], x=1..10, labels=
```

```
["n","error"], labeldirections=[horizontal,vertical]);
```




```
> p2 := plot([ [error_dfz],  [error_df3] ], style=[point,point],
  color=[blue,magenta]);
  with(plots):
  display(p1,p2);
```





Fehlerordnung?

Der lineare Koeffizient bestimmt die Fehlerordnung der Differenzenformel, d.h. Dfz hat die Fehlerordnung 2 und Df3 hat die Fehlerordnung 3.

b) Listen Sie die absoluten Fehler für die Schrittweiten $h_i = 10^{-i}$, $i = 1..20$, auf. Erzeugen Sie eine Tabelle mit Spalten für die Schrittweite h_i und die absoluten Fehler von `Dfz` und `Df3`. Nutzen Sie printf für die formatierte Ausgabe. Interpretieren Sie das Ergebnis.

```
> error_table := seq(
  [evalf(10**(-i), 1),
  abs(f_strich(x_0) - evalf(subs([f = f(x), x[0] = x_0, h=10**(-i)
  ], Dfz))),
  abs(f_strich(x_0) - evalf(subs([f = f(x), x[0] = x_0, h=10**(-i)
  ], Df3)))
  ], i=1..20);
```

$$
\begin{aligned}
&\mathit{error\_table} := [0.1, 0.0086243277134061059397689921864272725631, \qquad \textbf{(13)}\\
&\quad 0.0018892577199145351686920435795539032791], [0.01,\\
&\quad 0.0000854815644454507497956842979095039726,\\
&\quad 3.3169640307472784183282319139938024 \times 10^{-6}], [0.001,\\
&\quad 8.547410001917508664816827037150476 \times 10^{-7}, 3.5464363468235379355732453687524\\
&\quad \times 10^{-9}], [0.0001, 8.5474025389822951069258732762476 \times 10^{-9},\\
&\quad 3.5707058364454489140019275857 \times 10^{-12}], [0.00001,\\
&\quad 8.54740246435309204255881457476 \times 10^{-11}, 3.5731466932801005902025857 \times 10^{-15}],\\
&\quad [1. \times 10^{-6}, 8.547402463606800026821807476 \times 10^{-13}, 3.5733909187620942859191\\
&\quad \times 10^{-18}], [1. \times 10^{-7}, 8.5474024635993371063807476 \times 10^{-15}, 3.5734153427094525857\\
&\quad \times 10^{-21}], [1. \times 10^{-8}, 8.54740246359926238807476 \times 10^{-17}, 3.5734177777859191\\
&\quad \times 10^{-24}], [1. \times 10^{-9}, 8.547402463599388807476 \times 10^{-19}, 3.5734277859191 \times 10^{-27}],\\
&\quad [1. \times 10^{-10}, 8.5474024632388807476 \times 10^{-21}, 4.7611192524 \times 10^{-30}], [1. \times 10^{-11},\\
&\quad 8.54740252388807476 \times 10^{-23}, 2.69055474143 \times 10^{-29}], [1. \times 10^{-12},\\
&\quad 8.547102388807476 \times 10^{-25}, 6.4277859191 \times 10^{-30}], [1. \times 10^{-13}, 8.1602388807476\\
&\quad \times 10^{-27}, 2.6730944525857 \times 10^{-27}], [1. \times 10^{-14}, 6.602388807476 \times 10^{-28},\\
&\quad 1.39935722140809 \times 10^{-26}], [1. \times 10^{-15}, 3.06602388807476 \times 10^{-26}, 6.39935722140809\\
&\quad \times 10^{-26}], [1. \times 10^{-16}, 2.693397611192524 \times 10^{-25}, 6.026730944525857 \times 10^{-25}], [1.\\
&\quad \times 10^{-17}, 2.7306602388807476 \times 10^{-24}, 1.27306602388807476 \times 10^{-23}], [1. \times 10^{-18},\\
&\quad 1.27306602388807476 \times 10^{-23}, 9.60639935722140809 \times 10^{-23}], [1. \times 10^{-19},\\
&\quad 2.372693397611192524 \times 10^{-22}, 2.372693397611192524 \times 10^{-22}], [1. \times 10^{-20},\\
&\quad 2.372693397611192524 \times 10^{-22}, 1.47627306602388807476 \times 10^{-20}]
\end{aligned}
$$

Anmerkung: Ich habe zuerst probiert, die Tabelle aus dieser Sequence zu erstellen, was allerdings nicht geklappt hat, deshalb die Lösung unten. **Siehe unten, Tabelle mit seq.**

```
> printf("\n i \t | \t h \t | \t error Dfz \t | \t error Df3 \t |
  \t error Dfz - error Df3\n"):
  printf("-------------------------------------------------------------
  --"):
  printf("-------------------------------------------------------------
  --\n"):
  f_strich_x0 := f_strich(x_0):
  for i from 1 to 20 do
    h := evalf(10**(-i)):
    error_dfz := abs(f_strich_x0 - evalf(subs([f = f(x), x[0] =
```

```
x_0, h=h], Dfz))):
  error_df3 := abs(f_strich_x0 - evalf(subs([f = f(x), x[0] =
x_0, h=h], Df3))):
  error_diff := error_dfz - error_df3:
  printf(" %d \t | %10g \t | \t %g \t | \t %g \t | \t %g \n", i,
h, error_dfz, error_df3, error_diff):
end do:
```

```
i            |            h            |            error Dfz    |            error Df3
|            error Dfz - error Df3
-----------------------------------------------------------------------------------------
-----------------------------------
1            |         0.1             |            0.00862433              |
0.00188926             |            0.00673507
2            |        0.01             |            8.54816e-05             |
3.31696e-06            |            8.21646e-05
3            |       0.001             |            8.54741e-07             |
3.54644e-09            |            8.51195e-07
4            |      0.0001             |            8.54740e-09             |
3.57071e-12            |            8.54383e-09
5            | 1.00000e-05             |            8.54740e-11             |
3.57315e-15            |            8.54705e-11
6            | 1.00000e-06             |            8.54740e-13             |
3.57339e-18            |            8.54737e-13
7            | 1.00000e-07             |            8.54740e-15             |
3.57342e-21            |            8.54740e-15
8            | 1.00000e-08             |            8.54740e-17             |
3.57342e-24            |            8.54740e-17
9            | 1.00000e-09             |            8.54740e-19             |
3.57343e-27            |            8.54740e-19
10           | 1.00000e-10             |            8.54740e-21             |
4.76112e-30            |            8.54740e-21
11           | 1.00000e-11             |            8.54740e-23             |
2.69055e-29            |            8.54740e-23
12           | 1.00000e-12             |            8.54710e-25             |
6.42779e-30            |            8.54704e-25
13           | 1.00000e-13             |            8.16024e-27             |
2.67309e-27            |            5.48714e-27
14           | 1.00000e-14             |            6.60239e-28             |
1.39936e-26            |            -1.33333e-26
15           | 1.00000e-15             |            3.06602e-26             |
6.39936e-26            |            -3.33333e-26
16           | 1.00000e-16             |            2.69340e-25             |
6.02673e-25            |            -3.33333e-25
17           | 1.00000e-17             |            2.73066e-24             |
1.27307e-23            |            -1.00000e-23
18           | 1.00000e-18             |            1.27307e-23             |
```

```
9.60640e-23              |            -8.33333e-23
19         | 1.00000e-19            |            2.37269e-22              |
2.37269e-22              |            0
20         | 1.00000e-20            |            2.37269e-22              |
1.47627e-20              |            -1.45255e-20
```

Offensichtlich wird der Fehler mit kleinerem h immer kleiner und nähert sich daher immer weiter an das exakte Ergebnis an. Dies gilt für beide Differenzenformeln. Die Spalte mit der Differenz zeigt allerdings auch, dass nicht eindeutig gesagt werden kann, welche Formel nun genauer/besser ist.

Genauer:
Der Grund für den Anstieg des Fehlers ab einem bestimmten i liegt in den Differenzenformeln.
Die Differenzbildung mit anschliessender Division durch sehr kleine Schrittweiten verursacht Auslöschung signifikanter Dezimalstellen bei Gleitpunktoperationen mit endlicher Mantisse (hier 40). Die Genauigkeit nimmt bei der Differenzenformel Df3 für $i \geq 10$ und bei der Differenzenformel Dfz für $i \geq 14$ kontinuierlich ab.

Hier die Tabelle noch kompakter mit einer seq

```
> i;           # Durch deine Schleife hat i den Wert 21.
  i := 'i';    # Zurücksetzen des Wertes;
```

$$21$$

$$i := i \tag{14}$$

```
> h:=10.0^(-i):
> x[0] := 0.5:
> d1:=abs(D(f)(x[0])-Dfz):
  d2:=abs(D(f)(x[0])-Df3):
> seq(printf("%10.2e %10.2e %10.2e\n",h,d1,d2),i=1..20);
```

```
 1.00e-01   8.62e-03   1.89e-03
 1.00e-02   8.55e-05   3.32e-06
 1.00e-03   8.55e-07   3.55e-09
 1.00e-04   8.55e-09   3.57e-12
 1.00e-05   8.55e-11   3.57e-15
 1.00e-06   8.55e-13   3.57e-18
 1.00e-07   8.55e-15   3.57e-21
 1.00e-08   8.55e-17   3.57e-24
 1.00e-09   8.55e-19   3.57e-27
 1.00e-10   8.55e-21   4.76e-30
 1.00e-11   8.55e-23   2.69e-29
 1.00e-12   8.55e-25   6.43e-30
 1.00e-13   8.16e-27   2.67e-27
 1.00e-14   6.60e-28   1.40e-26
```

```
1.00e-15   3.07e-26   6.40e-26
1.00e-16   2.69e-25   6.03e-25
1.00e-17   2.73e-24   1.27e-23
1.00e-18   1.27e-23   9.61e-23
1.00e-19   2.37e-22   2.37e-22
1.00e-20   2.37e-22   1.48e-20
```