```
> 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);
```

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

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));
```

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

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;
```

$$Digits := 40 \tag{3}$$

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

$$f := x \mapsto \sin(x)\cdot\ln(x) \tag{4}$$

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

$$x\_0 := 0.5 \tag{5}$$

```
> f_strich := unapply(diff(f(x), x), x);
```

$$f\_strich := x \mapsto \cos(x)\cdot\ln(x) + \frac{\sin(x)}{x} \tag{6}$$

```
> 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);
```

$$error\_dfz := [1, -2.064274749209769717810313823033664576421], [2, -4.068127538171408198110071128330010742225], [3, \tag{7}$$

$-6.068165463315968997585185315986050125753$], [4,
$-8.068165842508378893325967685926370389085$], [5,
$-10.06816584630029708992055981221489722549$], [6,
$-12.06816584633821627129624130205565794481$], [7,
$-14.06816584633859546309907540961563173440$], [8,
$-16.06816584633859923416496719745934293175$], [9,
$-18.06816584633859283618600739250888044741$], [10,
$-20.06816584640772040395321570632197114024$]

```
> 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);
```

$error\_df3 := [1, -2.723708794485635684589324719028324500233], [2,$ **(8)**
$-5.479259237956753280937589557509982580596$], [3,
$-8.450207830675559586486817156920309641461$], [4,
$-11.44724592658422807642670954819534601221$], [5,
$-14.44694915374597468388150217794469178894$], [6,
$-17.44691947062249050600920660836468423468$], [7,
$-20.44691650225183739305286664806271887342$], [8,
$-23.44691620711536558206699678329757953112$], [9,
$-26.44693524618019788997072130943944652222$], [10,
$-29.55891483583055155222651796144713462839$]

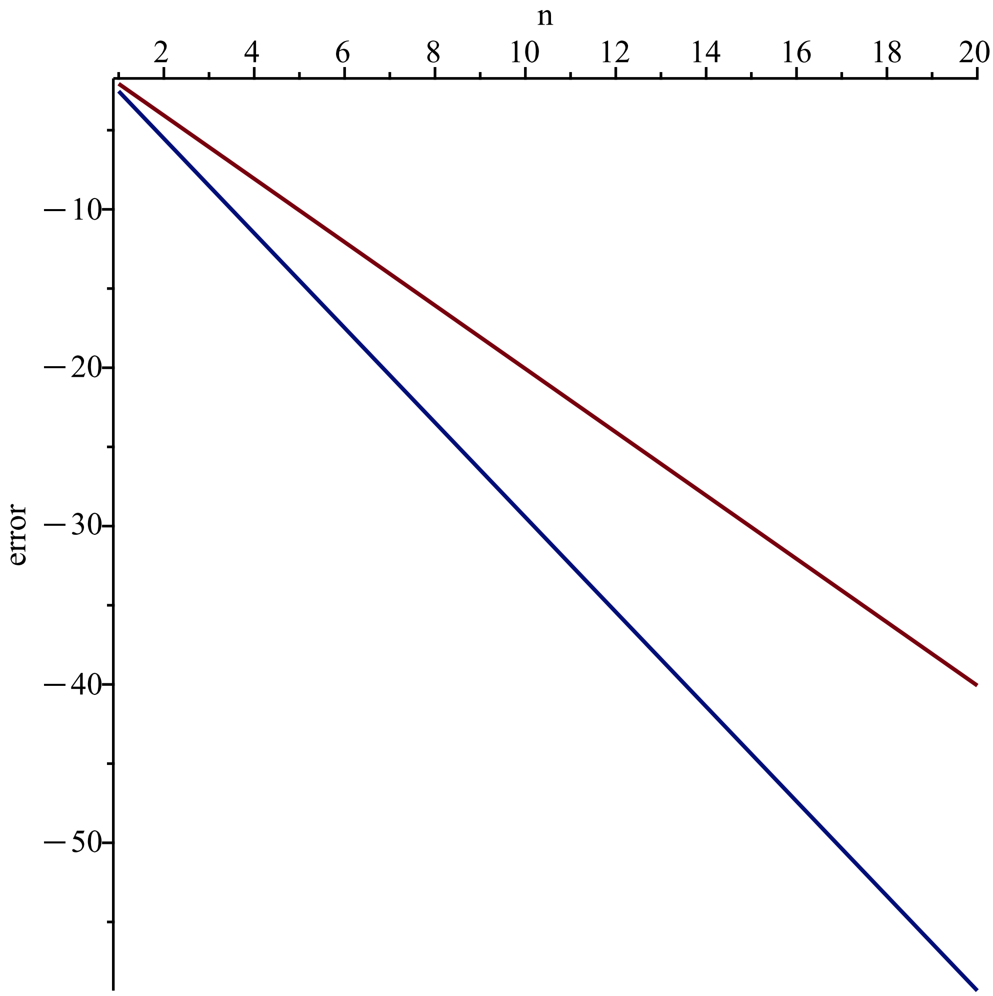
```
> g_error_dfz := unapply(CurveFitting:-LeastSquares([error_dfz],
  x), x);
```

$g\_error\_dfz := x \mapsto -0.06659653517363575336477431510785253260657$ **(9)**
$- 2.000213878536930719487342938588617307848 \cdot x$

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

$g\_error\_df3 := x \mapsto 0.4530396913133648704976325965517781426793$ **(10)**
$- 2.989534002156040780738628698831418177094 \cdot x$

```
> plot([g_error_dfz(x), g_error_df3(x)], x=1..20, labels=["n",
  "error"], labeldirections=[horizontal,vertical]);
```



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.