Aufgabe	A1	A2	A3	A4	$\sum$
Punkte					

Aufgabe 1. a) Es ergibt sich durch Ausrechnen:

$$\begin{aligned} a_0 &= y_0 = \frac{1}{2} \\ a_1 &= \frac{y_1 - a_0 N_0(t_1)}{N_1(t_1)} = \frac{1-\frac{1}{2}}{1-\frac{1}{4}} = \frac{2}{3} \\ a_2 &= \frac{y_2 - a_0 N_0(t_2) - a_1 N_1(t_2)}{N_2(t_2)} = -\frac{4}{45} \end{aligned}$$

Damit folgt

$$p_2(t) = \frac{1}{2} + \frac{2}{3}\left(t - \frac{1}{4}\right) - \frac{4}{45}\left(t-\frac{1}{4}\right)(t-1).$$

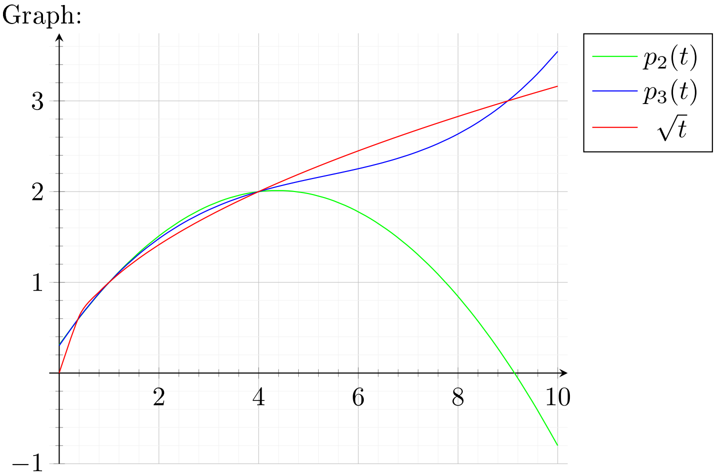
b) Es folgt für $t_3 = 9$ $y_3 = 3$, also

$$a_3 = \frac{y_3 - a_0 N_0(t_3) - a_1 N_1(t_3) - a_2 N_2(t_3)}{N_3(t_3)} = \frac{13}{1575}$$

Damit folgt

$$p_3(t) = \frac{1}{2} + \frac{2}{3}\left(t-\frac{1}{4}\right) - \frac{4}{45}\left(t-\frac{1}{4}\right)(t-1) + \frac{13}{1575}\left(t-\frac{1}{4}\right)(t-1)(t-4).$$

c) Graph:



Aufgabe 2. a) Bezeichne:

$$\begin{aligned} r_{i,0}(x) &:= y_i \\ r_{i,k}(x) &:= \frac{(x-x_i)p_{i+1,k-1}(x) - (x-x_{i+k})p_{i,k-1}(x)}{x_{i+k}-x_i}. \end{aligned}$$

Z.z.: $p_{i,0}(x) = r_{i,0}(x)$ und $p_{i,k}(x) = r_{i,k}(x)$.

$r_{i,0}$ bzw. $r_{i,k}$ sind Polynome vom Grad 0 bzw. k. D.h. es genügt zu zeigen, dass sie die Interpolationseigenschaft erfüllen. Die Behauptung folgt dann aus der Eindeutigkeit des Interpolationspolynoms.

Für $r_{i,0}(x)$ ist die Interpolationseigenschaft trivialerweise für die eine Stützstelle x_i erfüllt, denn $r_{i,0}(x_i) = y_i$.

Für $r_{i,k}$ gilt für $i \leq j \leq i+k$:

$$\begin{aligned} r_{i,k}(x_j) &= \frac{(x_j - x_i)p_{i+1,k-1}(x_j) - (x_j - x_{i+k})p_{i,k-1}(x_j)}{x_{i+k} - x_i} \\ &\overset{(*)}{=} \frac{(x_j - x_i)y_j - (x_j - x_{i+k})y_j}{x_{i+k} - x_i} \\ &= y_j \frac{x_j - x_i - x_j + x_{i+k}}{x_{i+k} - x_i} \\ &= y_j \frac{x_{i+k} - x_i}{x_{i+k} - x_i} \\ &= y_j. \end{aligned}$$

$(*)$: Falls $j = i$, dann ist $p_{i+1,k-1}(x_j)$ i.A. nicht y_j, aber dann ist $(x_j - x_i) = (x_i - x_i) = 0$, also gilt dennoch $(x_j - x_i)p_{i+1,k-1}(x_j) = (x_j - x_i)y_j$. Analog für $j = i+k$.

b) Durch Berechnung von $p_{0,3}(61.7)$ mit dem Schema aus (a) erhält man die Tageslänge am Ort E mit 19h $29,55$m.

Aufgabe 3. a)

- Für die Koeffizienten a_i gilt $a_i = y_i$. Also keine Operationen nötig.
- Auswertung von $L_i^{(n)}(\xi) = \prod_{j=0, j\neq i}^n \frac{(\xi - x_j)}{x_i - x_j}$: n Faktoren mit 3 Operationen plus $n-1$ Multiplikationen für das Produkt. Gesamt: $3n + n - 1 = 4n - 1$.

  Auswertung von $p(\xi)$: $n+1$ Summanden mit $4n-1$ Operationen plus Multiplikation mit y_i, ergibt $(n+1)\cdot(4n-1+1) = 4n^2$. Mit zusätzlich n Additionen für die Auswertung der Summe ergibt sich insgesamt $4n^2 + n = \mathcal{O}(n^2)$.

b)

- Berechnung der Koeffizienten erfordert die Lösung eines LGS mit unterer Dreicksmatrix. Dies erfordert $\mathcal{O}(n^2)$ Operationen.
- Auswertung von $N_i(\xi) = \prod_{j=0}^{i-1}(\xi - x_j)$ erfordert 1 Addition pro Faktor und insgesamt $i-1$ Multiplikationen für das Produkt, also insgesamt: $2i - 1$.

  Auswertung von $p(\xi)$ erfordert entsprechend die Auswertung von $N_i(\xi)$ und die Multiplikation mit dem Koeffizienten und Summation über alle $(n+1)$ Summanden. Ergibt also insgesamt mit kleinem Gauß $(n+1)(n+2) = n^2 + 3n + 2 = \mathcal{O}(n^2)$.

c)

- Berechnung der Koeffizienten erfordert die Lösung eines vollen LGS, also $\mathcal{O}(n^3)$ Operationen.
- Auswertung eines Summanden: $a_i t^i$ erfordert $i - 1 + 1 = i$ Multiplikationen.

  Auswertung von $p(x)$ erfordert das Summieren von $n+1$ Summanden, also n zusätzliche Additionen. Insgesamt folgt also wieder mit kleinem Gauß $\frac{(n+1)(n+2)}{2} + n = \frac{n^2+3n+2}{2} + n = \frac{n^2+5n+4}{2} = \mathcal{O}(n^2)$.

d) Auswertung von $p_{i,k}$ im Neville Schema erfordert

$$\begin{aligned} N(k) &= 2 + N(k-1) + 1 + 1 + 1 + N(k-1) + 1 + 1 \\ &= 7 + 2N(k-1) \\ &= 7 + 2\cdot 7 + 4 \cdot N(k-2) = \ldots = \sum_{j=0}^{k-1} 7 \cdot 2^j \\ &= 7\frac{2^k - 1}{2-1} \\ &= 7(2^k - 1) \\ &= \mathcal{O}(2^k). \end{aligned}$$

Aufgabe 4. a) Auszug aus *polynom.cc*

```
// Auswertung von p_{i,k}(t) mit Neville Schema bei t=x
template<typename REAL>
REAL evaluateNeville(REAL x, std::vector<REAL> &xs, std::vector<REAL> &ys, int i, int k) {
    if(k == 0) {
        return ys[i];
    } else {
```

```
        // nevile rekursionsformel
        return ((x - xs[i])*evaluateNeville(x, xs, ys,i+1,k-1) - (x - xs[i+k])*evaluateNeville(x, xs, ys, i, k-1))/(xs[i+k]-xs[i]);
    }
}

// Auswertung eines Interpolationspolynoms p(t) bei t=x
template<typename REAL>
REAL evaluate(REAL x, std::vector<REAL> &xs, std::vector<REAL> &ys) {
    // nutze neville verfahren mit p(x) = p_{0,n}(x)
    evaluateNeville(x, xs, ys, 0, xs.size()-1);
}
```

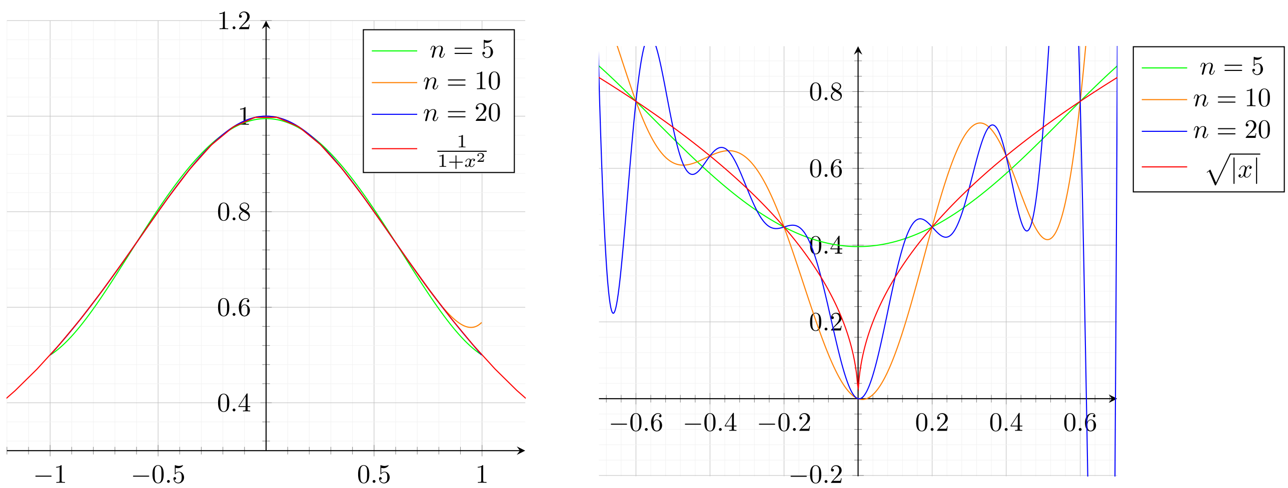
Auswertung des Interpolationspolynoms für vorgegebene Stüztstellen

b) Auszug aus *polynom.cc*

```
// Auswertung eines Interpolationspolynoms einer Funktion bei vorgegebenen Stuetzstellen
template <typename REAL>
REAL evaluateFunction(REAL x, std::vector<REAL> &xs, REAL(*f)(REAL)) {
    std::vector<double> ys(xs.size());
    for (int i = 0; i < xs.size(); i++) {
        // berechne stuetzstellen
        ys[i] = f(xs[i]);
    }
    // verwende polynominterpolation
    return evaluate(x, xs, ys);
}

// Auswertung eines Interpolationspolynoms einer Funktion bei aequidistanten Stuetzstellen
template <typename REAL>
REAL evaluateFunctionEqualDist(REAL x, REAL a, REAL b, REAL h, REAL(*f)(REAL)) {
    std::vector<double> xs(std::floor((b-a)/h));
    std::vector<double> ys(xs.size());
    for (int i = 0; i < xs.size(); i++) {
        // berechne stuetzstellen
        xs[i] = a + i*h;
        // und funktionswerte
        ys[i] = f(xs[i]);
    }
    // werte polynom aus
    return evaluate(x, xs, ys);
}
```

Auswertung von I-Polynomen für äquidistante Stützstellen



(a) $f_1 = \frac{1}{1+x^2}$ (b) $f_2 = \sqrt{|x|}$

c) Für $\frac{1}{1+x^2}$ funktioniert die Interpolation mit äquidistanten Stützstellen sehr gut. Für $\sqrt{|x|}$ entstehen durch die nicht differenzierbare Stelle bei $x = 0$ sehr große Abweichungen, die zu den Rändern mit wachsendem Polynomgrad sogar zunehmen.