# Scientific Computing

# Task 3 Find the root

Felix Diemling

Tim Seeliger

Handed in: 7. Dezember 2022

# 1 Im the Solution

The Meaningfulness and Error is depending mainly on the method "time_to_destinationäs discussed Task 2, so we chose $n = 2000$. The Error of the method it self can be directly read off by taking the difference. An higher $n$ would lead to a higher run-time, therefor the $n$ needs to big enough to gain the wanted accuracy but not bigger than that to reduce the run-time.

## 1.1 Route

How far will Anna and Elsa get along in 30 min?

- Anna: 51.1 km
- Elsa: 37.0 km

## 1.2 Range

How far would Anna and Elsa get on their respective routes with a battery charge of C = 10, 000 Wh?

- Anna: 52.7 km
- Elsa: 65.0 km

# 2 Code

```
def distance(T, route,n=10**7):
    strecke = load_route(route)[0]
    def _distance(T,min,max):
        ttt = time_to_destination(min+(max -min)/2, route, n)
        if 0<=T-ttt <=10**(-10):
            print("Timediff:",T - ttt)
            return (min+(max -min)/2)


        elif ttt < T:
            return _distance(T,min+(max -min)/2,max)
        else:
            return _distance(T,min,max-(max -min)/2 )


    if time_to_destination(max(strecke), route, 1000)<T:
        print("Stop",time_to_destination(max(strecke), route, n))
        return max(strecke)
    else:
        return _distance(T,0 , max(strecke))
```

Her some Text

```


def reach(C, route, n = 10**7):
    strecke = load_route(route)[0]
    def _reach(x):
        leakage=total_consumption(x, route, n)
        if 0<leakage-C< 10**(-10):
            print("Succes:",leakage)
            return x
        else:
            return _reach(x-(leakage-C)/consumption(velocity(x, route)))


    if total_consumption(max(strecke),route,n)<C:
        print("Stop",total_consumption(max(strecke), route, n))
        return max(strecke)
    else:
        return _reach(0)
```