Blatt 10

Vincent Kümmerle und Elvis Gnaglo

25. Dezember 2025

1 Problem des Handlungsreisenden

```python
import itertools
import math
import random
import matplotlib.pyplot as plt
import time

def distanz(p1, p2):
    """Berechnet die euklidische Distanz zwischen zwei Punkten p1 und p2."""
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

def pfad_laenge(pfad):
    """Berechnet die Gesamtlänge eines Pfades (Liste von Punkten)."""
    laenge = 0
    # Iterieration von 0 bis zum vorletzten Punkt und addition der Distanz zum Nachfolger
    for i in range(len(pfad) - 1):
        laenge += distanz(pfad[i], pfad[i+1])
    return laenge

def main():
    n = 10  # Anzahl der Punkte
    random.seed() # Setzt den Seed für reproduzierbare Ergebnisse, ohne Wert ist das Ergebnis für jede Ausführung des Programms unterschiedlich

    # Zufällige Punkte für x und y zwischen 0 und 100 generieren
    punkte = [(random.uniform(0, 100), random.uniform(0, 100)) for _ in range(n)]

    print(f"Berechne kürzesten Pfad für {n} Punkte...")
```

```
    print(f"Zu prüfende Permutationen: {math.factorial(n):,}")

    start_time = time.time()

    # Alle Permutationen berechnen und die mit der kürzesten Strecke finden
    kuerzeste_strecke = float('inf')
    bester_pfad = None

    # itertools.permutations erstellt alle möglichen Reihenfolgen
    for perm in itertools.permutations(punkte):
        aktuelle_laenge = pfad_laenge(perm)

        if aktuelle_laenge < kuerzeste_strecke:
            kuerzeste_strecke = aktuelle_laenge
            bester_pfad = perm

    end_time = time.time()
    print(f"Fertig in {end_time - start_time:.2f} Sekunden.")
    print(f"Kürzeste Strecke: {kuerzeste_strecke:.2f}")


    if bester_pfad:
        # Koordinaten für den Plot entpacken
        x_coords = [p[0] for p in bester_pfad]
        y_coords = [p[1] for p in bester_pfad]

        plt.figure(figsize=(8, 6))

        # Den Weg zeichnen
        plt.plot(x_coords, y_coords, color='black', linestyle='-', linewidth=2, zorder=1, label='Kürzester Pfad')

        # Punkte anzeigen
        plt.scatter(x_coords, y_coords, color='green', s=100, zorder=2, label='Orte')

        # Start- und Endpunkt beschriften
        plt.text(x_coords[0], y_coords[0], ' Start', verticalalignment='bottom', fontweight='bold')
        plt.text(x_coords[-1], y_coords[-1], ' Ende', verticalalignment='bottom', fontweight='bold')

        plt.title(f'Kürzester Pfad durch {n} zufällige Punkte\nLänge: {kuerzeste_strecke:.2f}')
```

```
66          plt.xlabel('X-Koordinate')
67          plt.ylabel('Y-Koordinate')
68          plt.grid(True)
69          plt.legend()
70
71          plt.show()
72
73  if __name__ == "__main__":
74      main()
```

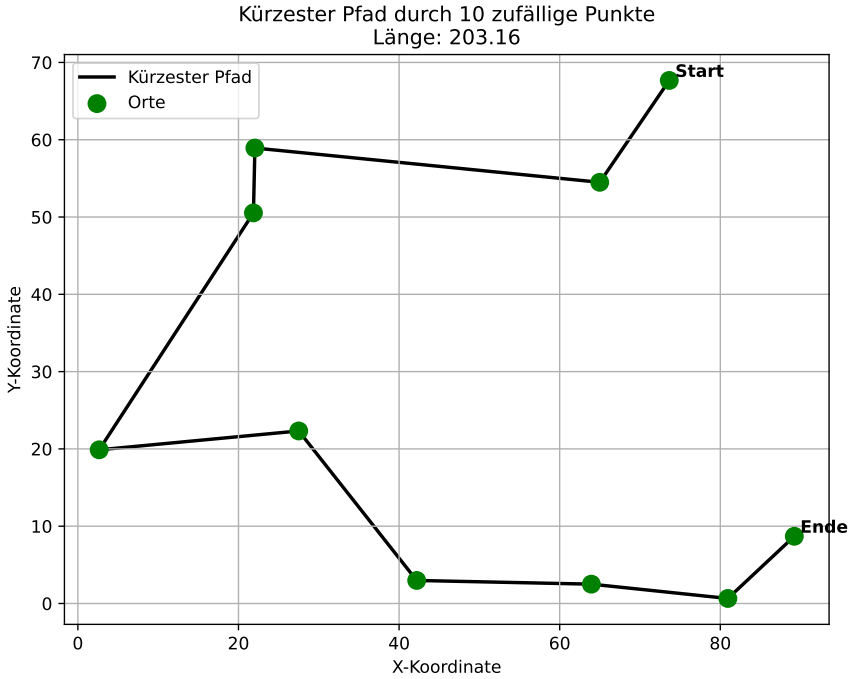

Abbildung 1: Plot einer zufälligen Reisestrecke

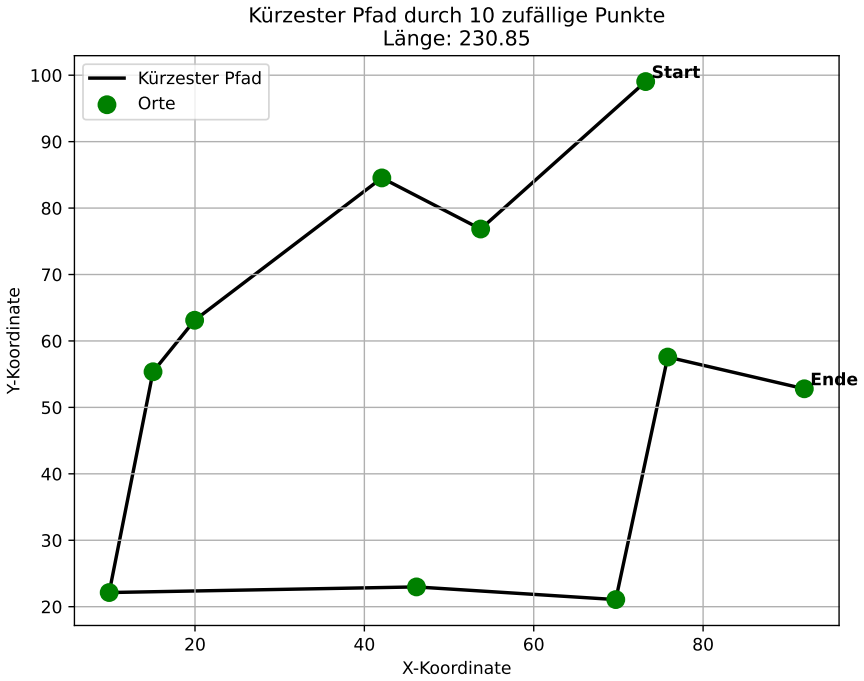

Abbildung 2: Plot einer anderen zufälligen Reisestrecke

2 Fehleranalyse einer gedämpften Schwingung