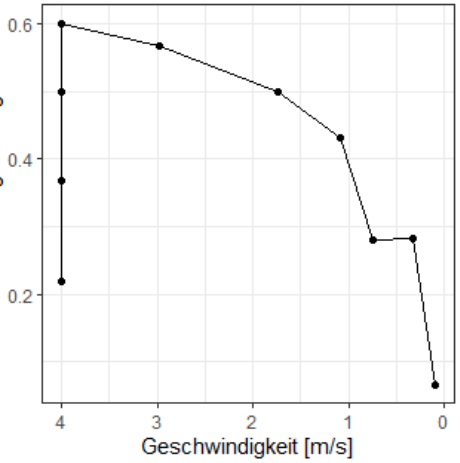
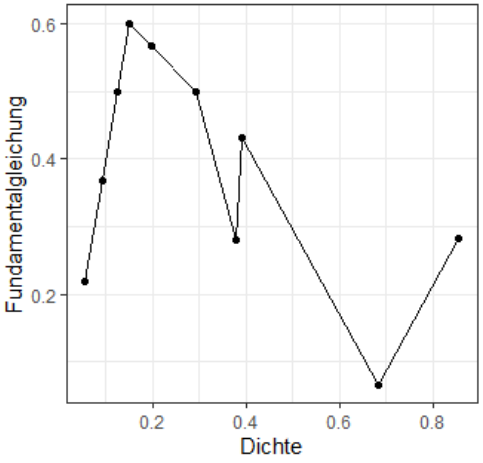
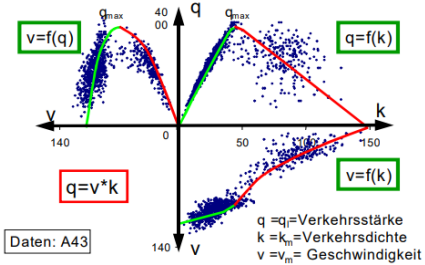
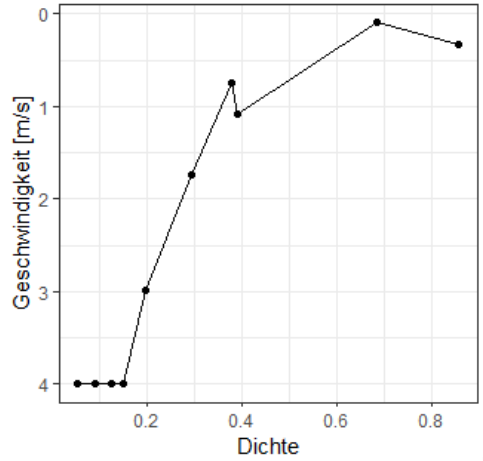






Quelle: Abbildung 2 aus „Verkehr auf Schnellstraßen im Fundamentaldiagramm“ von „https://homepage.ruhr-uni-bochum.de/ning.wu/pdf/FMDG_SVT_8_2000.pdf“



Arbeits- und Ergebnis Präsentation im Fach Wissenschaftliches Arbeiten

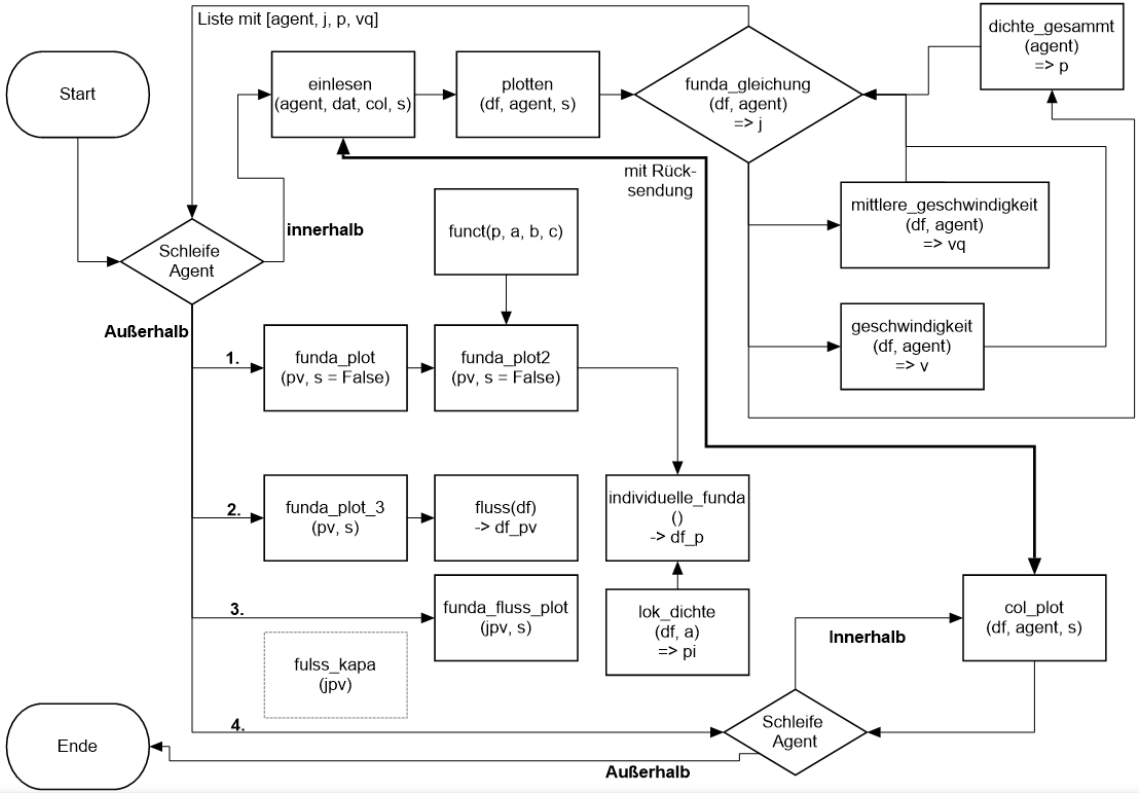
BERGISCHE UNIVERSITÄT WUPPERTAL

Überblick

1. Vorgehen
2. Quellcode
3. Ergebnisse

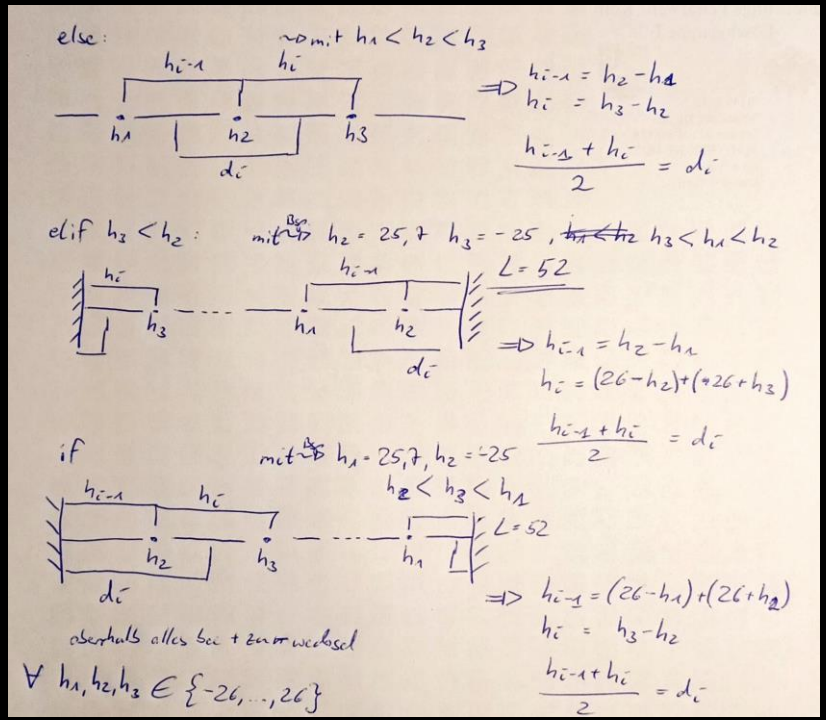
- 18 Stunden für die Simulation
- 63.346.551 Datenzeilen
- je Agent, 1 min Rechenzeit ∑ = 164 min

Darstellung des Programablaufs

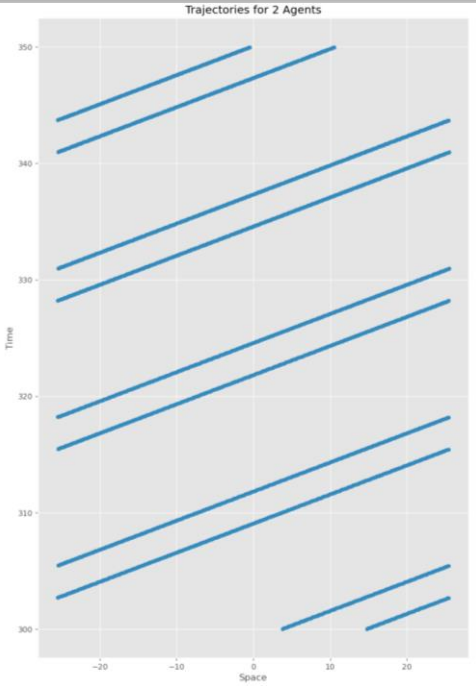


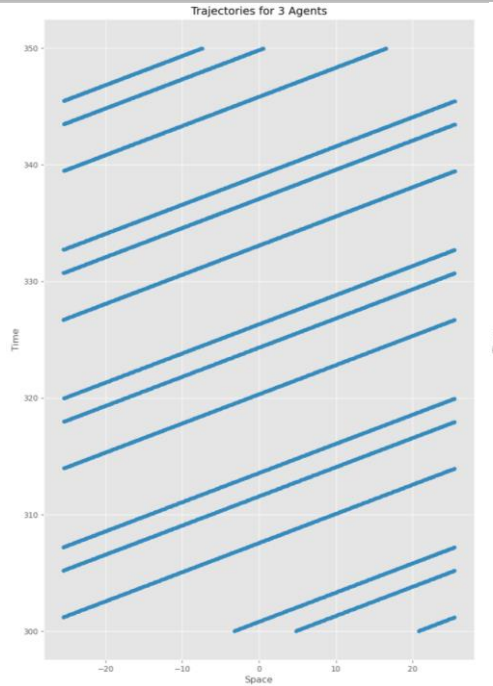
Code zu Aufgabe 4

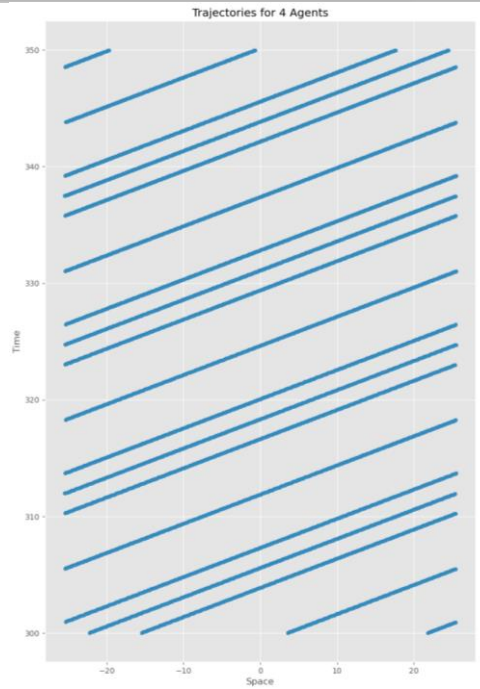
```python
def lok_dichte(df, a = 'N0N'):
    # Gl.6: pi = 1 / di mit di: siehe Abbildung 3
    p = [-999]

    for i in tqdm(range(1, len(df) - 1), desc =
            f'Dichte für {a} Agenten' ):
        h1 = df['x'].iloc[i-1]
        h2 = df['x'].iloc[i]
        h3 = df['x'].iloc[i+1]

        if h2 < h3 < h1: # Bruch zw h1 & h2
            di = ((26 - h1) + (26 + h2)) + np.sqrt((h3 - h2)**2)
        elif h3 < h1 < h2: # Bruch zw h2 & h3
            di = np.sqrt((h2 - h1)**2) + ((26 - h2) + (26 + h3))
        else:
            di = (np.sqrt((h2 - h1)**2) + np.sqrt((h2 - h1)**2))

        pi = 1 / (di / 2)
        p.append(pi)

    p.append(-999)

    return pi
```

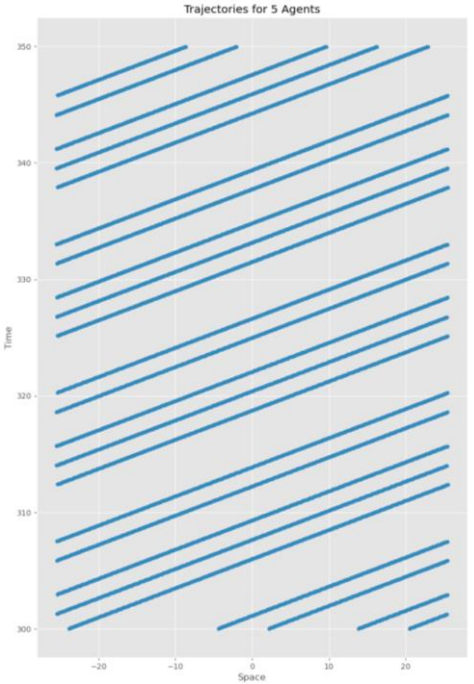


NetLogo Trajectorien

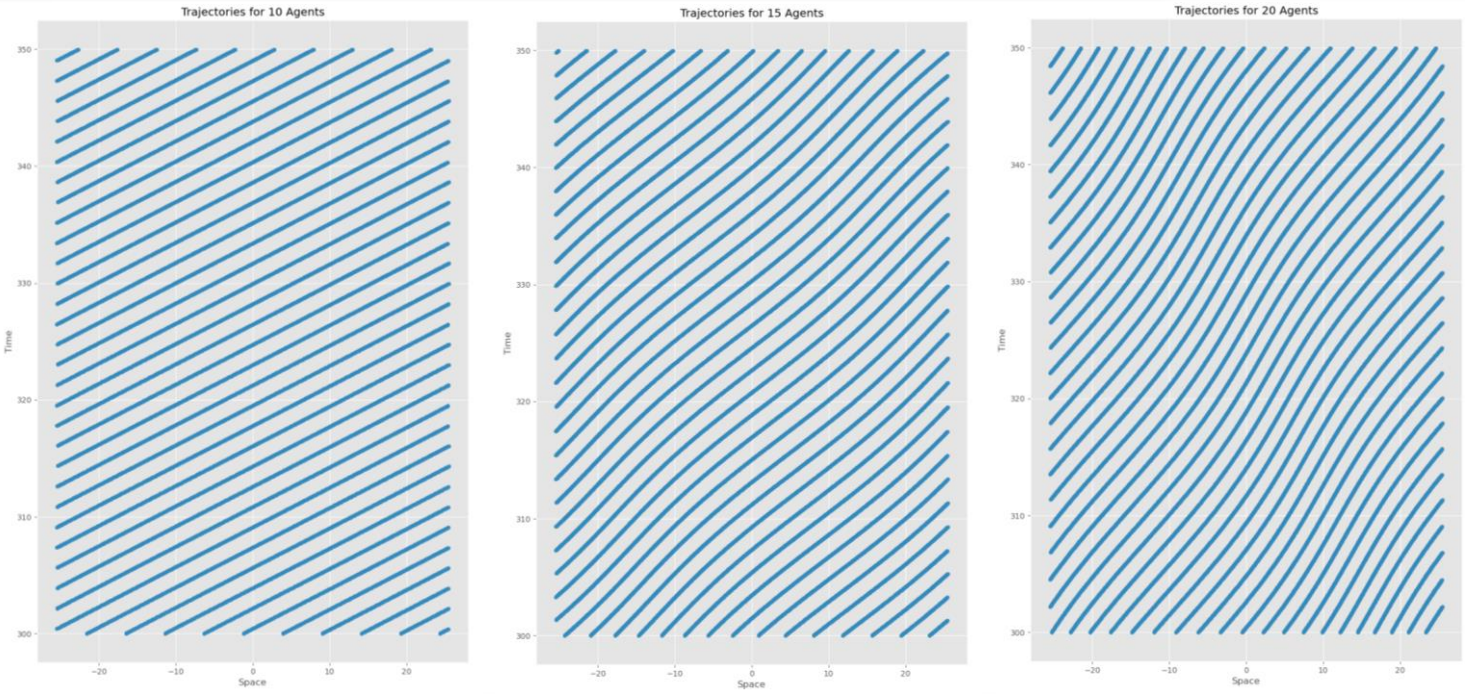




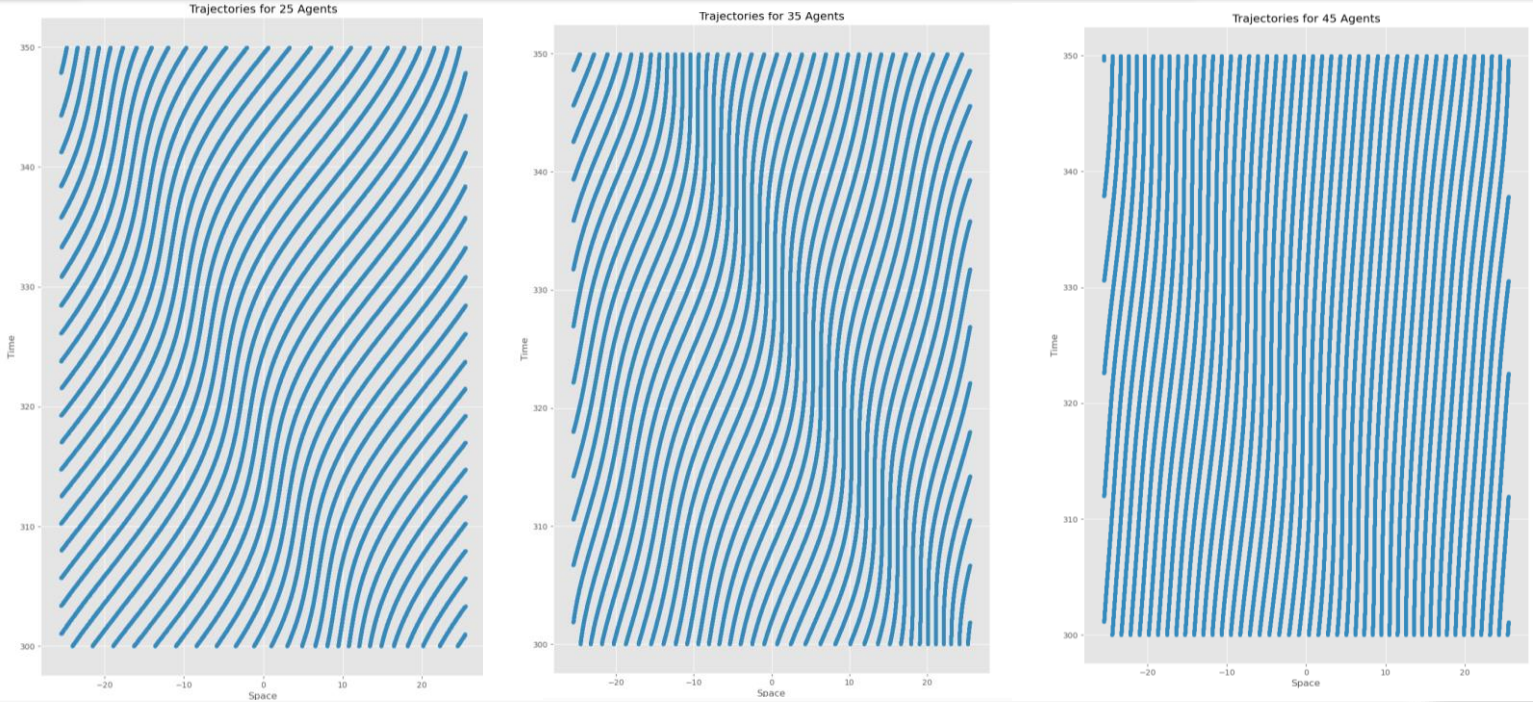




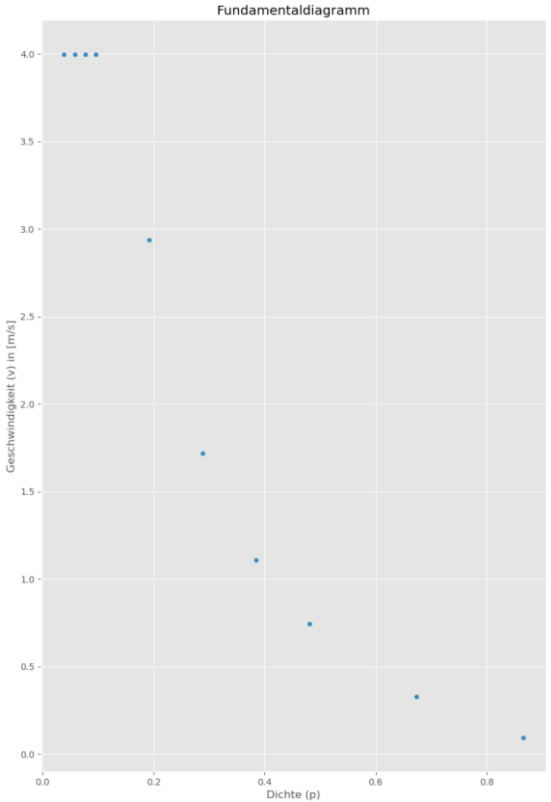
NetLogo Trajectorien

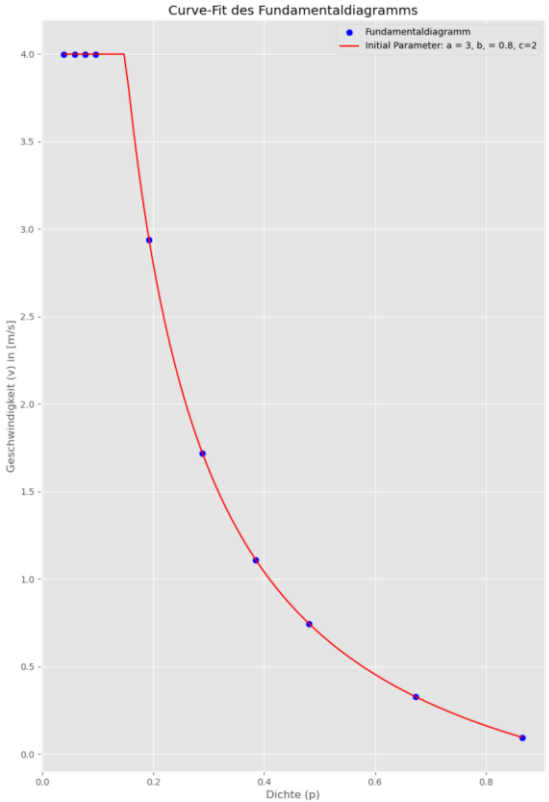


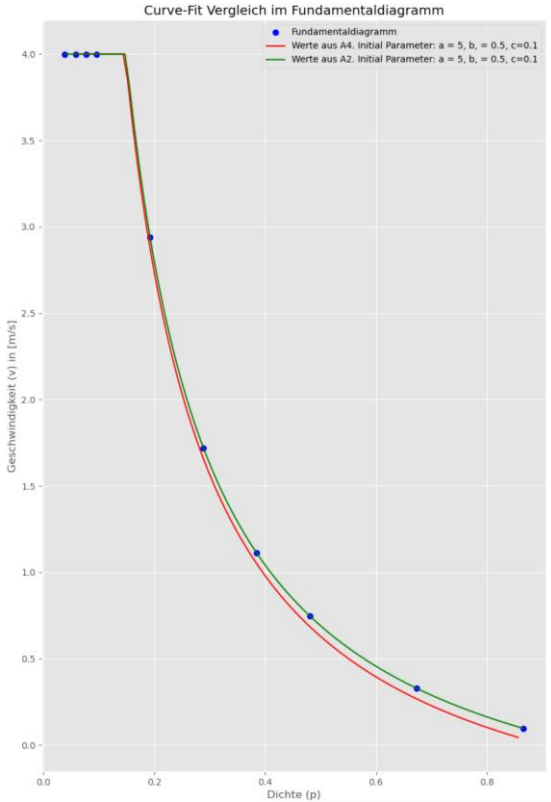
NetLogo Trajectorien



Fundamentaldiagramm



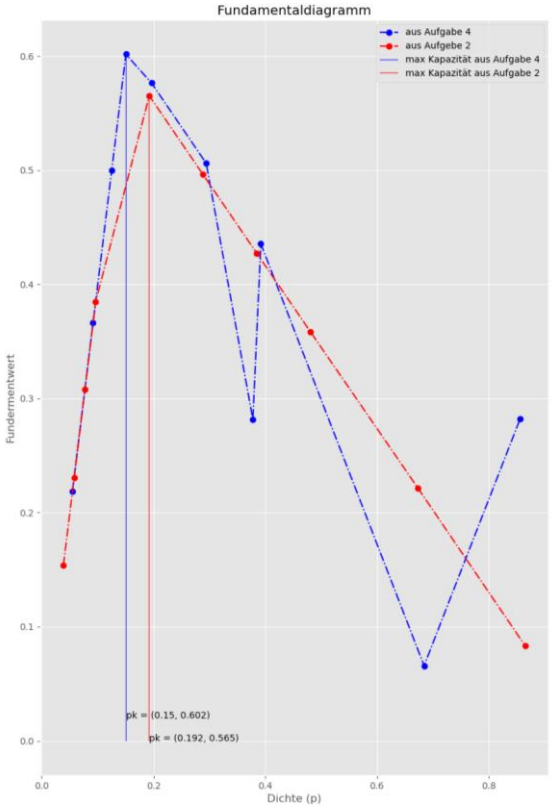


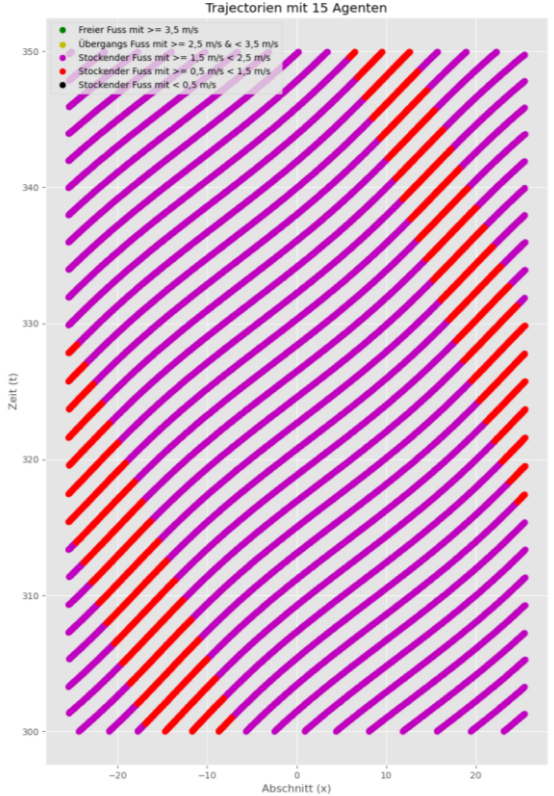


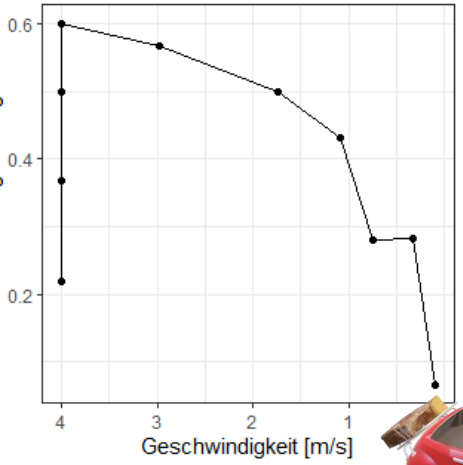
Kritische Dichte / Stop-and-Go Wellen

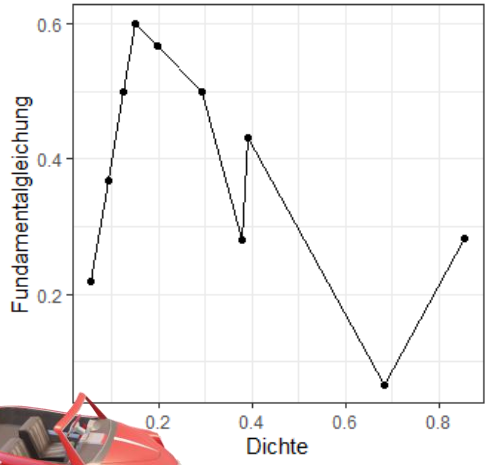
- Unterschiedliche Krit-Dichten
- Gleichmäßig vs Sprunghafter Abfall und Anstieg

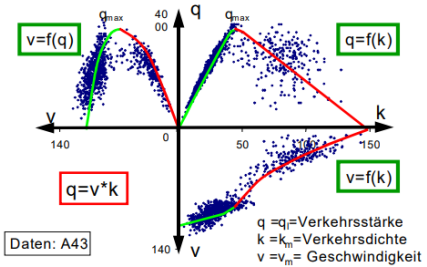
- Stop-and-Go Wellen Zu bereits an den Steigungen zu erkennen



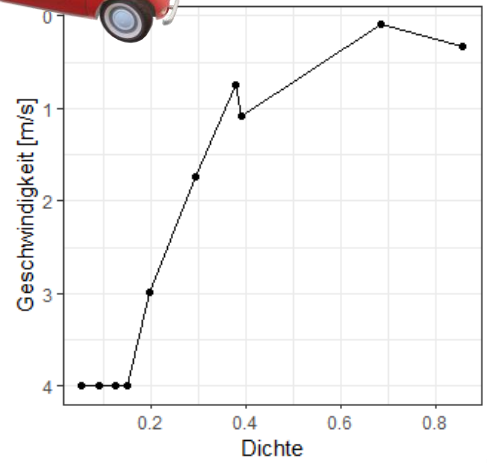








Quelle: Abbildung 2 aus „Verkehr auf Schnellstraßen im Fundamentaldiagramm“ von „https://homepage.ruhr-uni-bochum.de/ning.wu/pdf/FMDG_SVT_8_2000.pdf“



Danke für Ihre Aufmerksamkeit

BERGISCHE UNIVERSITÄT WUPPERTAL