In [ ]:
```
#
#   Example 2: Single server queueing system parameters vs arrival rate
#
#   Author: Pavlius Tervydis
#   Date: 2023-07-09
#
# ============================================================
#             Single Server Queueing System (ssqs)
#          +-----------------------------------+
#   Arrival |                Server             |    Output
#          |    Queue     +-----------------+   |
#  arrival_rate | ---------> |                 |   |
#  ------------->| | | | | | | service_rate | |  ------------>
#          |    +---------> |                 |   |
#          |                +-----------------+   |
#          +-----------------------------------+
#
# System notation (A/B/1):
# A - distribution of inter-arrival time
# B - distribution of service time is determined
# 1 - single server
#
# Distribution types:
# M - exponential,
# D - determined (fixed),
# G - general (time variance must be provided))
# ============================================================

from qsystems import ssqs
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# ------------------------------------------------------------
# System initial parameters:
# ------------------------------------------------------------
# List of queueing systems to be evaluated
system_notation = ["M/M/1","M/D/1","D/M/1","D/D/1"]
# Arrival parameters:
arrival_rate_start = 0 # [entities/t.u.], for example [users/hour]
arrival_rate_stop = 80
arrival_rate_step = 0.5
# Server parameters
service_rate = 100 # [entities/t.u.]
# Note: For stable system arrival_rate must be < service_rate
# ------------------------------------------------------------

#
# Example how to estimate dependence of system parameters on arrival time.
#
arrival_rate_array = np.arange(arrival_rate_start,
                               arrival_rate_stop+arrival_rate_step,
                               arrival_rate_step)

# Results are stored in pandas dataframe
param_df = pd.DataFrame()
for q_system in system_notation:
    for arrival_rate in arrival_rate_array:
        sys_params = ssqs(qs=q_system, ar=arrival_rate, sr=service_rate)
        df = pd.DataFrame(sys_params,index=[0])
        param_df  = pd.concat([param_df ,df], axis=0)
param_df.reset_index(drop=True, inplace=True)

print(param_df)

# ------------------------------------------------------------
# Plot parameters
#
def plot_system_param_versus(q_system,param_df,param,versus):
    param_text =  {"w" :"Mean waiting time",
                   "wq":"Mean waiting time in queue",
                   "l" :"Mean number of entities in system",
                   "lq":"Mean number of entities in queue",
                   "ar":"Arrival rate",
                   "u" :"System utilization"}
    fig_title = param_text[param]+" vs "+param_text[versus]
    plt.figure(fig_title)
    plot_data_x = param_df[(param_df["qs"]==q_system)][versus]
    plot_data_y = param_df[(param_df["qs"]==q_system)][param]
    plt.plot(plot_data_x,plot_data_y,label=q_system)
    plt.xlabel(param_text[versus])
    plt.ylabel(param_text[param])
    plt.legend()
    plt.grid(True)
    plt.draw()

for q_system in system_notation:
    plot_system_param_versus(q_system,param_df,'w','ar')
    plot_system_param_versus(q_system,param_df,'wq','ar')
    plot_system_param_versus(q_system,param_df,'lq','ar')
    plot_system_param_versus(q_system,param_df,'l','ar')
    plot_system_param_versus(q_system,param_df,'w','u')
    plot_system_param_versus(q_system,param_df,'wq','u')
    plot_system_param_versus(q_system,param_df,'lq','u')
    plot_system_param_versus(q_system,param_df,'l','u')
plt.show()
```

```
      qs    ar     sr           a        va     s        vs        u         l  \
0    M/M/1   0.0  100.0         inf       inf  0.01  0.0001  0.000  0.000000
1    M/M/1   0.5  100.0  2.000000  4.000000  0.01  0.0001  0.005  0.005025
2    M/M/1   1.0  100.0  1.000000  1.000000  0.01  0.0001  0.010  0.010101
3    M/M/1   1.5  100.0  0.666667  0.444444  0.01  0.0001  0.015  0.015228
4    M/M/1   2.0  100.0  0.500000  0.250000  0.01  0.0001  0.020  0.020408
..     ...   ...    ...       ...       ...   ...     ...    ...       ...
639  D/D/1  78.0  100.0  0.012821  0.000000  0.01  0.0000  0.780  0.780000
640  D/D/1  78.5  100.0  0.012739  0.000000  0.01  0.0000  0.785  0.785000
641  D/D/1  79.0  100.0  0.012658  0.000000  0.01  0.0000  0.790  0.790000
642  D/D/1  79.5  100.0  0.012579  0.000000  0.01  0.0000  0.795  0.795000
643  D/D/1  80.0  100.0  0.012500  0.000000  0.01  0.0000  0.800  0.800000

           lq        wq         v
0    0.000000  0.000000  0.010000
1    0.000025  0.000050  0.010050
2    0.000101  0.000101  0.010101
3    0.000228  0.000152  0.010152
4    0.000408  0.000204  0.010204
..        ...       ...       ...
639  0.000000  0.000000  0.010000
640  0.000000  0.000000  0.010000
641  0.000000  0.000000  0.010000
642  0.000000  0.000000  0.010000
643  0.000000  0.000000  0.010000

[644 rows x 12 columns]
```