In [ ]:

```
#
#   Example 3: Finding maximum arrival rate to ensure that mean
#   waiting time is below critical
#
#   Author: Paulius 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"]
# 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.
# Results are stored in pandas dataframe.
#
arrival_rate_array = np.arange(arrival_rate_start,
                               arrival_rate_stop+arrival_rate_step,
                               arrival_rate_step)

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)

# ------------------------------------------------------------
# Having such data it is possible to estimate how system parameters depend on
# arrival rate. It can be used to ensure quality of service requirements.
# As it was shown in the Example 2 - system parameters highly depend
# on arrival rate or utilization.
# For example, if it is necessary to achieve that mean waiting time is
# below critical, then the maximum arrival rate can be determined.
#
waiting_time_critical = 0.02

# Finding what is the maximum arrival rate to ensure that mean waiting time is < critical value
max_arrival_rate_at_critical_waiting_time = {}
for q_system in system_notation:
    row_found = param_df[param_df['w']<=waiting_time_critical) & (param_df['qs']==q_system)].tail(1)
    max_arrival_rate_at_critical_waiting_time[q_system] = float(row_found['ar'].iloc[0])

print("Max arrival rate at critical mean waiting time",
      max_arrival_rate_at_critical_waiting_time)


# ------------------------------------------------------------
# Plot Waiting times vs Arrival rate
#
plt.figure()
for q_system in system_notation:
    plot_data_x = param_df[(param_df["qs"]==q_system)]['ar']
    plot_data_y = param_df[(param_df["qs"]==q_system)]['w']
    plt.plot(plot_data_x,plot_data_y,label=q_system+
                ": arrival rate (at critical mean waiting time) = %.2f"%
                max_arrival_rate_at_critical_waiting_time[q_system])
    plt.axvline(max_arrival_rate_at_critical_waiting_time[q_system],
                color='black')
plt.axhline(waiting_time_critical,color='red')
plt.text(0,waiting_time_critical,'critical mean waiting time')
plt.xlabel("Arrival rate")
plt.ylabel("Mean waiting time")
plt.title("Mean waiting time vs Arrival rate")
plt.grid()
plt.legend()
plt.draw()
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
..     ...   ...    ...       ...       ...   ...     ...    ...       ...
478  D/M/1  78.0  100.0  0.012821  0.000000  0.01  0.0001  0.780  1.826072
479  D/M/1  78.5  100.0  0.012739  0.000000  0.01  0.0001  0.785  1.877794
480  D/M/1  79.0  100.0  0.012658  0.000000  0.01  0.0001  0.790  1.932027
481  D/M/1  79.5  100.0  0.012579  0.000000  0.01  0.0001  0.795  1.988955
482  D/M/1  80.0  100.0  0.012500  0.000000  0.01  0.0001  0.800  2.048780

          lq        wq         w
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
..        ...       ...       ...
478  1.046072  0.013411  0.023411
479  1.092794  0.013921  0.023921
480  1.142027  0.014456  0.024456
481  1.193955  0.015018  0.025018
482  1.248780  0.015610  0.025610

[483 rows x 12 columns]
Max arrival rate at critical mean waiting time {'M/M/1': 50.0, 'M/D/1': 66.5, 'D/M/1': 73.5}
```