

Kód 5: Funkce pro výpočet hodnot tlaku a průtoku v místě zavzdušnění

```

def airInlet(p, pA, pB, QA, QB, hAirInlet, time):
    global beta, QC_var, pC_var, Vv, Qv, cA_var, cB_var
    cA(pA, QA)
    cB(pB, QB)
    if (cA_var + cB_var) / 2 > hAirInlet*rho*g + pa:
        p = (cA_var + cB_var) / 2
    else:
        p = hAirInlet*rho*g + pa - dpv
        beta = p / (hAirInlet * rho * g + pa)
        if Vv <= 0 and beta > 1: # Ventil je zaplněn vodou a v potrubí je
přetlak
            pC_var = (cA_var + cB_var) / 2
            QC_var = (cA_var - cB_var) / (2 * B)
        elif Vv > 0 and beta > 1: # Ve ventilu je vzduch a v potrubí je
přetlak
            Qv = math.pi / 4 * dv ** 2 * (cAI * (1 - (1/beta) ** ((kappa - 1) /
kappa))) ** (1 / 2)
            pC_var = (cA_var + cB_var) / 2
            QC_var = (cA_var - pC_var) / B - Qv
            Vv = Vv - Qv * dt
            if Vv < 0:
                Vv = 0
                QC_var = (cA_var - cB_var) / (2 * B)
                pC_var = cA_var - B * QC_var
            elif beta < 1: # V potrubí je podtlak a ventil nasává vzduch
                Qv = math.pi / 4 * Dv ** 2 * (cAI * (1 - (beta) ** ((kappa - 1) /
kappa))) ** (1 / 2)
                pC_var = hAirInlet*rho*g + pa
                QC_var = (cA_var - pC_var) / B + Qv
                Vv = Vv + Qv * dt
            elif beta == 1: # V potrubí je atmosférický tlak
                pC_var = (cA_var + cB_var) / 2
                if pC_var < hAirInlet*rho*g + pa:
                    pC_var = hAirInlet*rho*g + pa - dpv
                QC_var = (cA_var - pC_var) / B
            if pC_var < hAirInlet*rho*g + pa: # kontrola nově vypočteného
tlaku na podtlak
                pC_var = hAirInlet*rho*g + pa
                QC_var = (cA_var - pC_var) / B

```

Kód 6: Hlavní funkce numerického výpočtu

```

def MoC():

    print('výpočet množství úseků')
    # výpočet celkové délky trasy po elementárních úsecích
    L_steps = 0
    for i in range(0, len(topology)):
        if topology.iat[i,0] > 0:
            L_steps = L_steps + topology.iat[i,0]/dx
        else:
            L_steps = L_steps + 1
    L_steps = int(L_steps)

    # Definice polí
    print('definice polí')
    Q = [[0 for x in range(L_steps)] for y in range(t_steps)] # pole
    průtoků v čase (y) a prostoru (x)
    p = [[0 for x in range(L_steps)] for y in range(t_steps)] # pole
    tlaků v čase (y) a prostoru (x)
    dV = [A*dx for x in range(L_steps)] # pole zaplnění potrubí

    # Výpočet počátečního stavu
    print('výpočet ustáleného stavu')
    K = 1
    p[0][0] = p0 # tlak v prvním bodě systému v počátečním stavu
    Q[0][0] = Q0 # průtok v prvním bodě systému v počátečním stavu
    for j in range(1, len(topology)): # výpočet úseků potrubí
        if re.search('pump', topology.iat[j, 2]):
            Q[0][K] = Q0
            dpPump0 = p[0][K-1] + odporTrasy(Q0)
            p[0][K] = dpPump0
            K = K+1
        elif topology.iat[j,0] == 0: # výpočet součástí potrubí
            Q[0][K] = Q0
            p[0][K] = p[0][K-1]
            K = K+1
        else: # výpočet potrubí
            l_steps = int(topology.iat[j, 0] / dx) # délka úseku ve
            formě elementárních částí
            dz = dx * topology.iat[j, 1] / topology.iat[j, 0] #
            elementární převýšení
            for k in range(0, l_steps): # výpočet funkce úseku potrubí
                Q[0][K] = Q0
                p[0][K] = p[0][K-1]-R*Q[0][K]*abs(Q[0][K])
                K = K+1

    if j == len(topology)-1:
        global pOut
        pOut = p[0][K-1]

```

Pokračování na další straně.

Pokračování kódu 6:

```

# výpočet průběhu rázu
print('výpočet rázu')
for i in range (0, t_steps-1): # výpočet v čase od 0 sekund do t
    sekund po úsecích dt
    percentage(i)
    K = 0

    for j in range (0, len(topology)): # výpočet úseků potrubí

        if topology.iat[j,0] == 0: # 0 znamená okrajovou podmínku
            eval(topology.iat[j, 2]) # výpočet funkce příslušné
okrajové podmínky
            p[i + 1][K] = pC_var
            Q[i + 1][K] = QC_var
            K=K+1
        else:
            l_steps = int(topology.iat[j,0]/dx) # délka úseku ve
formě elementárních částí
            for k in range (0, l_steps): # výpočet funkce úseku
potrubí
                pipeline(p[i][K-1], p[i][K+1], Q[i][K-1], Q[i][K+1],
i, K) # výpočet funkce potrubí
                p[i+1][K]=pC_var
                Q[i+1][K]=QC_var
                K=K+1

# zápis dat do souboru MoC_output.csv
print('zápis výsledků do souboru')
with open('D:\škola\DP_data\MoC_ex_output_p.csv', 'w', newline='') as
outputFile:
    writer = csv.writer(outputFile, delimiter=';', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
    for i in range (0, t_steps-1):
        writer.writerow(p[i])
    with open('D:\škola\DP_data\MoC_ex_output_Q.csv', 'w', newline='') as
outputFile:
        writer = csv.writer(outputFile, delimiter=';', quotechar='|',
quoting=csv.QUOTE_MINIMAL)
        for i in range (0, t_steps-1):
            writer.writerow(Q[i])

```