

Úloha pro uchazeče o pozici Python programátora

ENGINN EFFECT, s.r.o.

ČÁST I.

Inženýrské problémy často nemají přímočaré matematické řešení, a tak se při jejich řešení používají iterativní metody – začne se s přibližným odhadem řešení a to se potom po malých krocích zpřesňuje. Každý krok se přitom dopouští numerické chyby (tzv. rezidua) ve sledovaných veličinách (po dosazení do řešených rovnic vznikne malý, ale nenulový rozdíl mezi levou a pravou stranou).

V příloze naleznete soubor **log.run**, který obsahuje výpis z řešení [simulace vodní hladiny](#) pomocí open-source simulačního nástroje [OpenFOAM](#). Součástí výpisu je informace o časových krocích, kde jeden časový krok je zaznamenán takto:

```
Courant Number mean: 0.0016654654 max: 0.25123892
Interface Courant Number mean: 0.00020602875 max: 0.25123892
deltaT = 2.392917e-05
Time = 0.300024

PIMPLE: iteration 1
MULES: Solving for alpha.water
MULES: Solving for alpha.water
MULES: Solving for alpha.water
Phase-1 volume fraction = 0.58897389 Min(alpha.water) = -1.1432245e-25 Max(alpha.water) = 1.0048862
velocityDampingConstraint Ulimits damped 9 (0.063608736%) of cells
GAMG: Solving for p_rgh, Initial residual = 7.9232624e-06, Final residual = 3.2184215e-08, No Iterations 3
GAMG: Solving for p_rgh, Initial residual = 1.9800585e-07, Final residual = 6.8626421e-09, No Iterations 3
time step continuity errors : sum local = 6.1403646e-09, global = 4.2769024e-10, cumulative = -0.00080327351
GAMG: Solving for p_rgh, Initial residual = 3.8339894e-08, Final residual = 2.7057609e-09, No Iterations 3
GAMG: Solving for p_rgh, Initial residual = 1.3408015e-08, Final residual = 1.2505107e-09, No Iterations 3
time step continuity errors : sum local = 1.1188673e-09, global = 3.6851269e-11, cumulative = -0.00080327348
GAMG: Solving for p_rgh, Initial residual = 5.148228e-09, Final residual = 5.3787757e-10, No Iterations 3
GAMG: Solving for p_rgh, Initial residual = 2.1047953e-09, Final residual = 2.1559607e-10, No Iterations 3
time step continuity errors : sum local = 1.9290067e-10, global = 4.2343121e-12, cumulative = -0.00080327347
GAMG: Solving for p_rgh, Initial residual = 8.4983153e-10, Final residual = 8.791249e-11, No Iterations 3
GAMG: Solving for p_rgh, Initial residual = 3.5725952e-10, Final residual = 2.7704275e-11, No Iterations 3
time step continuity errors : sum local = 2.4787944e-11, global = 8.7868261e-13, cumulative = -0.00080327347
smoothSolver: Solving for omega, Initial residual = 1.0562149e-05, Final residual = 2.6312466e-09, No Iterations 1
smoothSolver: Solving for k, Initial residual = 1.8041503e-05, Final residual = 1.4599796e-08, No Iterations 1
ExecutionTime = 0.43 s ClockTime = 1 s

surfaceFieldValue quarter_water_volumetric_flow_rate write:
  total faces   = 18
  total area    = 3.2092885e-05

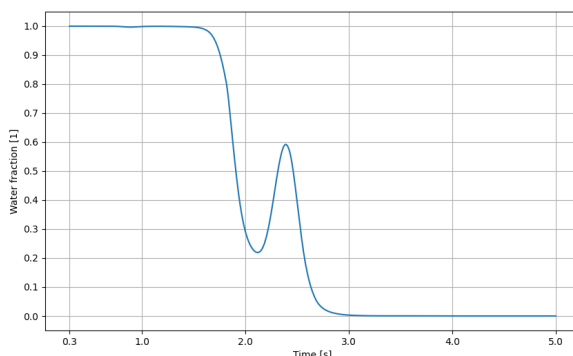
weightedSum(Vystup) of phi = 9.31427e-06
```

Vaším úkolem bude připravit nástroj pro sledování průběhu reziduí, aby si inženýr mohl zkontrolovat, že úloha běží v pořádku a rezidua výrazně nekolísají a nenarůstají v čase (to by mohlo vést k nevěrohodným výsledkům nebo pádu celé simulace).

- Připravte Python **třídu CheckResidual**, která:
- dovede načíst textový formát souboru **log.run**,
- najde **časové kroky** a **rezidua** veličin (viz žlutě výše – zajímá nás „Time“ tj. čas v sekundách a „Final residual“ pro jednotlivé veličiny v rámci daného časového kroku – je-li uvedeno více „Final residual“ pro téže veličinu, tak nás zajímá poslední z nich),
- **vykreslí do grafu** průběh reziduí (osa X je čas, osa Y je „Final residual“ v logaritmické škále, jsou vyneseny tři křivky „p_rgh“, „omega“ a „k“),
- **uloží do souboru** (ve formátu, který uznáte jako vhodný) čas a hodnoty reziduí pro další zpracování – aby nebylo nutné v budoucnu opět „lovit“ hodnoty v dlouhém textovém logu.

ČÁST II. (bonusová)

Z výše uvedené simulace jsme získali informaci o přítomnosti vody ve sledované oblasti. Textový soubor **alpha.water** obsahuje údaj o podílu vody v okolí daného bodu. (Zvláštní termín „podíl vody“ je důsledkem použité metody, která nesleduje detailně tvorbu vodní hladiny, bublin, apod., ale pouze relativní podíl vody a vzduchu v malých objemech řádově cm^3 – pod hladinou je „podíl vody“ blízký 1,0 a nad hladinou 0,0; v okolí hladiny je „něco mezi“).



Časový průběh ukazuje, že k zajímavému jevu dochází mezi 2. a 3. sekundou, potom se děj ustálí. Protože každá sekunda simulovaného děje typicky trvá hodiny až desítky hodin reálného času běhu počítače, je žádoucí simulaci co nejdříve ukončit.

- Připravte Python **třídu** **DynamicTermination**, která:
- dovede načíst textový formát souboru **alpha.water** a
- **nalezne a vypíše čas T** , kdy došlo ke splnění obou kritérií pro ukončení výpočtu:
 - hodnota „Water fraction“ v čase T je méně než 0,5 a
 - v předcházejících 20 000 řádcích logu „alpha.water“ (období cca 0,5 sekundy před T) je rozdíl minimální a maximální hodnoty „Water fraction“ méně než 0,1.

