

Studentų praktikų ir baigiamųjų darbų pasiūlymas

dr. Mantas Povilaitis

Atviro kodo skaičiuojamosios fluidų dinamikos sprendyklės *flameFoam* patobulinimas, struktūros ir kodo pertvarkymas

Studentų specializacija Pagrindinės arba gretutinės informatikos studijos ir/arba programavimo patirtis

Praktika siūloma Visų bakalauro ir magistrantūros kursų studentams/-ėms

flameFoam – <https://github.com/flameFoam> – yra atviro kodo CFD sprendyklė kuriama Lietuvos energetikos institute naudojant OpenFOAM – <https://github.com/OpenFOAM> – bibliotekas (C++) ir skirta dujų degimui bei sprogimui modeliuoti.

Sprendyklė yra aktyviai vystoma ir naudojama, tačiau jos kodui yra reikalingas pertvarkymas dėl keletu priežasčių:

- sprendyklė suprogramuota fragmentiškai pagal tyrimų poreikius
- sprendyklė suprogramuota neišnaudojant visų OpenFOAM suteikiamų objekcinio programavimo galimybių
- OpenFOAM bibliotekos yra aktyviai vystomos, flameFoam kodą ir struktūrą reikia atnaujinti pagal naujausius OpenFOAM pakeitimus
- sprendyklės galimybių išplėtimui

```
11 if (Correlation[i] == "PitschDuchamp")
12 {
13     scalar b1 = 2;
14     scalar b3 = 1;
15     scalar Scd = 0.5;
16     scalar b3b1 = pow(b3,2)/pow(b1,2);
17     volScalarField udDa = S_L*deltaByLf;
18     Xi = 1 + udByS_L*b3*sqrt(udDa/(ud*Scd-b3b1*udDa));
19 }
20 else if (Correlation[i] == "Charlette")
21 {
22     scalar Ck = 1.5;
23     scalar Ck_mult1 = 0.245454545454545; // 27/110
24     scalar Ck_mult2 = 0.648567745274438; // 4*sqrt(27/110)*18/55
25
26     scalar n43 = 4.0/3.0;
27     scalar pi43 = Foam::pow(Foam::constant::mathematical::pi, n43);
28     volScalarField Red = 4*deltaByLf*udByS_L*SMALL;
29
30     scalar beta = 0.5;
31
32     volScalarField fu = Ck_mult2*Foam::pow(Ck, 1.5)*pow(udByS_L, 2);
33     volScalarField fd = pow(Ck_mult1*Ck*pi43*max(0.0, pow(deltaByLf, n43) - 1), 0.5);
34     volScalarField fRe = pow(0.163636363636364*exp(-1.5*Ck*pi43/Red), 0.5)*pow(Red, 0.5);
35
36     volScalarField d = 0.6 + 0.2*exp(-0.1*udByS_L)-0.2*exp(-0.01*deltaByLf);
37
38     volScalarField gamma =
39     pow(
40     pow(
41     pow(fu+SMALL, -d)
42     +
43     pow(fd+SMALL, -d),
44     -1/d),
45     -1.4)
46     +
47     pow(fRe+SMALL, -1.4),
48     -0.714285714285714
49     );
50
51     Xi = pow(1 + min(deltaByLf, gamma*udByS_L), beta);
52 }
53 }
54 else
55 {
56     FatalErrorInFunction
57     << "Unknown turbulent flame speed model." << endl
58     << exit(FatalError);
59 }
60 }
```

kodas taps profesionalnesnis, tvarkingesnis ir artimesnis OpenFOAM bibliotekų programavimui bei atnaujintai standartinių sprendyklių struktūrai. Praktikos metu galima ir suprogramuoti naujas sprendyklės galimybes.

Studentui/-ei pageidaujant, praktikos metu galima tartis ir dėl baigiamojo darbo temos.

Iki praktikos pradžios studentui/-ei reiktų būti susipažinus su OpenFOAM CFD bibliotekomis ir standartinių sprendyklių struktūra. Neturint šių žinių, susitarus supažindinsime ir padėsime įsisavinti per porą savaičių – mėnesį prieš praktikos pradžią.

Kontaktai

Dr. Mantas Povilaitis
mantas.povilaitis@lei.lt
+370 (37) 401 920

Adresas

Lietuvos energetikos institutas
Breslaujos g. 3, Kaunas
254-AK

