

風雪・融雪シミュレーション開発ソルバー
テクニカルリファレンス

OpenFOAM 開発ソルバー名

「SnowFoam」

「snowMelt」

令和 5 年 3 月

目次

1. はじめに	1
2. 基礎式	2
2.1 SnowFoam	2
2.2 snowMelt	4
3. 解析手法	8
3.1 snowFoam	8
3.2 snowmelt	12
4. 物性値	16
4.1 snowFoam	16
4.2 snowMelt	17
5. 境界条件	19
5.1 パッチの設定	20
5.2 境界タイプの概要.....	22
5.3 積雪量：snowDepth.....	23
5.4 飛雪空間密度：snowDriftDensity.....	24
5.5 積雪重量：snowMass.....	25

1. はじめに

風雪・融雪シミュレーションについて、気流解析では標準ソルバーを使用しています。ただし、飛雪空間密度の輸送、積雪、融雪に関しては標準ソルバーでは表現できないことから、標準ソルバーをカスタマイズし、風雪・融雪ソルバーである「風雪：snowFoam」「融雪：snowMelt」を開発しました。

本書では、開発ソルバー「snowFoam」「snowMelt」について示します。

OpenFOAMの標準ソルバーの例		今回カスタマイズを実施したソルバー
<u>乱流/層圧縮性流体</u>		
rhoicoFoam	非定常層流解析ソルバー	
rhosimpleFoam	定常乱流解析ソルバー	
rhopisoFoam	非定常乱流解析ソルバー	
rhopimpleFoam	非定常乱流解析ソルバー (PIMPLE法 = PISO法 + SIMPLE法)	雪の計算に必要な対流拡散方程式を加え カスタマイズを行う。
<u>熱流動</u>		
buoyantSimpleFoam	定常流熱流動解析ソルバー	
buoyantPimpleFoam	非定常流熱流動解析ソルバー	
<u>その他</u>		
thermoFoam	凍結した流れ場での熱輸送、熱力学の計算のソルバー	融雪に関してカスタマイズを行う。

図 1 カスタマイズ元の OpenFOAM の標準ソルバーについて

参考文献

- 1) OpenFOAMライブラリリファレンス、森北出版、2020年4月発行
<https://www.morikita.co.jp/books/mid/069161>

2. 基礎式

2.1 SnowFoam

風雪シミュレーションは、OpenFOAMの標準ソルバーである非定常乱流解析ソルバー

「rhopimpleFoam」をベースに、**飛雪空間密度の輸送** 及び **雪面における積雪の侵食・堆積**を組み込みました。なお、これらのモデルについては、新潟工科大学富永教授らによって構築されたモデルを採用しています。

カスタマイズソルバー「snowFoam」の基礎式を以下に示します。

表 1 参考文献(風雪シミュレーション)

風雪シミュレーション	堆積・侵食のフラックスのモデル化に関する物性値 雪粒子の落下速度 (-0.5m/s) 雪面の摩擦速度、限界摩擦速度 (0.2m/s) 比例定数 氷の密度等	1)富永, 大風, 持田, 志田, 吉野, 雪面の侵食・堆積のモデル化に関する基礎的検討, 日本建築学会環境系論文集, 第 74 巻 2009
	屋根上吹き溜まり係数と屋根勾配の関係式	2)富永, 大風, 持田, 単体切り妻屋根建物を対象として屋根上積雪深の予測手法の検証, 日本建築学会構造論文集, 第 81 巻, 第 725 号, 2016
	流れ場の予測結果の風洞実験値	3)村山, 五十嵐, 富永, 2 段屋根建物の積雪分布の再現性の検討 CFD を用いた風による屋根雪辺分布の予測, 日本建築学会構造論文集, 第 84 巻, 第 762 号, 2019

(1) 飛雪空間密度の輸送方程式

雪粒子の運動形態は、浮遊(Suspension)をモデル化しています。

浮遊については、単位体積に含まれる雪粒子の密度が輸送されるオイラー的手法を採用しています。単位体積に含まれる雪粒子の質量を飛雪空間密度と定義し、次式の輸送方程式を解いています。

$$\frac{\partial \Phi}{\partial t} + \frac{\partial U_i \Phi}{\partial x_i} + \frac{\partial W_f \Phi}{\partial x_3} = \frac{\partial}{\partial x_i} \left(\frac{v_t}{\sigma_s} \frac{\partial \Phi}{\partial x_i} \right)$$

Φ : 飛雪空間密度[kg/m³]、 U_i : (x,y,z)方向の風速、 W_f : 飛雪の落下速度、 v_t : 渦動粘性係数、 σ_s : 乱流シュミット数

(2) 雪面における積雪の侵食・堆積

雪面における積雪の侵食・堆積については、雪面に働く摩擦応力と限界摩擦応力の関係から、堆積と侵食を判定し、それに応じた積雪深を計算しています。雪面における堆積フラックス q_{dep} 、侵食フラックス q_{ero} は次式を用いました。

$$q_{dep} = W_f \Phi_P \left(\frac{U_{*t}^2 - U_*^2}{U_{*t}^2} \right) \quad (U_* < U_{*t})$$

$$q_{ero} = -A_e \rho_i U_* \left(1 - \frac{U_{*t}^2}{U_*^2} \right) \quad (U_* > U_{*t})$$

q_{dep} 、 q_{ero} ：堆積・侵食フラックス[kg/m²s]、 U_{*t} ：雪面の限界摩擦速度[m/s]、 U_* ：雪面の摩擦速度[m/s]、 A_e ：比例定数、 ρ_i ：氷の密度

積雪フラックス q_{total} は次式の通りです。

$$q_{total} = q_{dep} + q_{ero}$$

積雪深の形状変化については、積雪深の増加分 Δh を次式で算出しました。

$$\Delta h = \frac{q_{total} \cdot \Delta t}{\rho_s}$$

Δt ：経過時間、 ρ_s ：積雪密度

2.2 snowMelt

融雪シミュレーションは、OpenFOAMの標準ソルバーである熱輸送解析ソルバー「thermoFoam」をベースに、**雪面の熱収支による融雪計算**を組み込みました。なお、これらのモデルについては、新潟工科大学富永教授らによって構築されたモデルを採用しています。

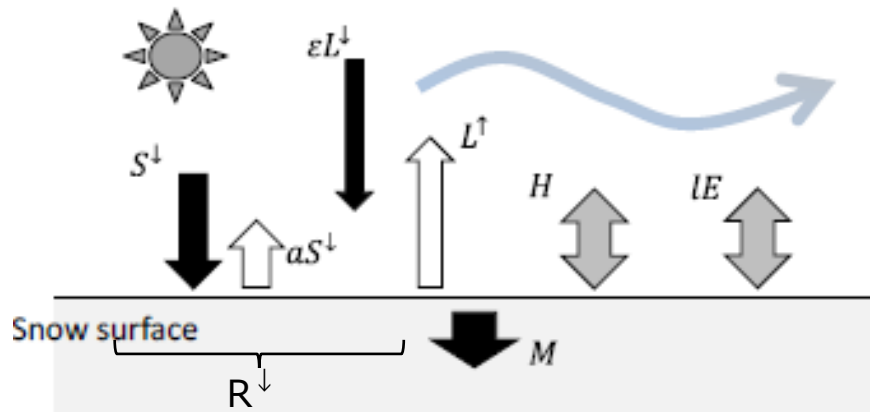
カスタマイズソルバー「snowMelt」の基礎式を以下に示します。

表 2 参考文献(融雪シミュレーション)

融雪シミュレーション	積雪表面層の熱収支に関する構成	4)富永, 本吉, 気象観測データと熱収支に基づく地上積雪量推定法, 日本建築学会構造系論文集, 第 86 巻, 第 782 号, 2021
------------	-----------------	------------------------------------------------------------------------

(1) 雪面の熱収支による融雪

融雪現象は、雪面の熱収支より融雪に費やされるエネルギーから融雪量を算出しています。
雪面の熱収支（融雪エネルギー、入力放射量、各熱輸送量）の構成を以下に示しました。



出典：富永ら，気象観測データと熱収支に基づく地上積雪量推定法，2021

積雪面の熱収支を次式に示しました。

$$M = R \downarrow - L \uparrow - H - lE$$

M：融雪エネルギー[W/m²]、R↓：入力放射量[W/m²]、L↑：上向き長波放射量[W/m²]、
H：顕熱輸送量[W/m²]、lE：潜熱輸送量[W/m²]

上式により得られた融雪エネルギーMから積雪深の変化量(融雪量)Δhは求められます。

$$\Delta h = \frac{M \times 3,600}{\rho_s \times I_c}$$

ρ_s：積雪密度[kg/m³]、I_c：氷の融解熱

a) 入力放射量 $R \downarrow$

$$R \downarrow = (1 - a)S \downarrow + \varepsilon L \downarrow$$

a : アルベド数、 $S \downarrow$: 全天候日射量[W/m²]、 $L \downarrow$: 大気放射量[W/m²]、 ε : 積雪の放出率

b) 上向き長波放射量 $L \uparrow$

$$L \uparrow = \varepsilon \sigma T_s^4$$

σ : ステファン・ボルツマン定数[W/m²/K⁴]、 T_s : 積雪表面温度[K]

c) 顕熱輸送量 H

$$H = C_p \rho C_H U (T_s - T)$$

C_p : 空気の比熱[J/kg/K]、 ρ : 空気の密度[kg/m³]、 C_H : 顕熱輸送のバルク輸送係数、 U : 高度1.0mでの風速[m/s]、 T : 気温[K]

d) 潜熱輸送量 lE

$$lE = l \rho C_E U [(1 - rh) q_{sat}(T) + \Delta \cdot (T_s - T)]$$

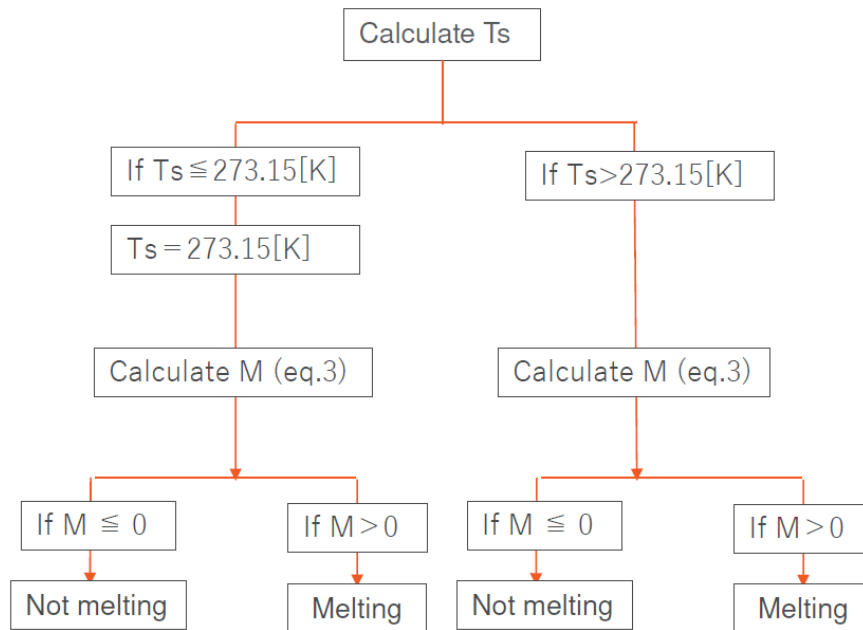
l : 氷の昇華の潜熱または水の昇化の潜熱[J/kg]、 C_E : 潜熱輸送のバルク輸送係数、 rh : 相対湿度、 $q_{sat}(T)$: 気温に対する飽和比湿[kg/kg]、 Δ : 飽和比湿の湿度に対する変化率

e) その他の計算式

ア) 積雪表面温度 T_s

$$T_s = \frac{R \downarrow - \varepsilon \sigma T^4 - l \rho C_H U (1 - rh) q_{sat}(T)}{4 \varepsilon \sigma T^3 + (l \Delta + C_p) \rho C_H U} + T$$

このように積雪表面温度の計算を線形化することで、収束解を求めるための反復計算を行う必要がなく、簡便に熱収支計算を行うことが可能となります。本式で求めた T_s が 0°C 以下であれば、融雪は起きない ($M=0$) として次のステップに進みます。 $T_s > 0^\circ\text{C}$ となったときは、 $T_s = 0^\circ\text{C}$ とおいて、 M を計算します。 $M > 0$ なら $T_s = 0^\circ\text{C}$ とおいたことと整合するので、 M を用いて融雪量を計算します。 $M < 0$ の場合、積雪内の液体水を再凍結させることになりますが、 $T_s = 0^\circ\text{C}$ は一般的に高すぎることから、 $T_s = T$ として M を計算しなおして凍結量を求めます。



イ) 気温 T における飽和比湿 $q_{sat}(T)$

$$q_{sat}(T) = \frac{0.622 e_{sat}(T)}{p_s - 0.378 e_{sat}(T)}$$

$$e_{sat}(T) = 6.108 \times 10^{7.5T/(237.5+T)}$$

$e_{sat}(T)$: 飽和水蒸気圧[hPa]、 p_s : 大気圧

ウ) 飽和比湿の温度 T に対する変化率 Δ

$$\Delta \equiv \frac{dq_{sat}}{dT} = \frac{de_{sat}}{dT} \times \frac{0.622 p_s}{(p_s - 0.378 e_{sat}(T))^2} = \frac{6.1078(2500 - 2.4T)}{0.4615(237.15 + T)^2}$$

3. 解析手法

風雪・融雪ソルバーである「風雪：snowFoam」「融雪：snowMelt」の解法について説明します。

3.1 snowFoam

ここでは、風雪現象を組み込んだ非定常乱流解析ソルバー「snowFoam」の飛雪空間密度の輸送及び積雪の侵食・堆積に関する解析手法について説明します。なお、気流解析に用いる定常乱流解析ソルバー「rhoSimpleFoam」等に関する基礎方程式やその解析手法については、各種参考資料を参照してください。

- ① 気流解析を行うため、定常乱流解析ソルバー「rhoSimpleFoam」でセル界面の物性値を計算しておきます。定常計算であるため、計算が安定するイタレーション数(反復回数)を設定しておく必要があります。
- ② 「snowFoam」で使用する気流条件(速度 U 、圧力 p など)は、①で算出された物性値を設定します。
- ③ 飛雪空間密度の方程式から、飛雪空間密度 Φ を得ます。仮に、 $\Phi < 0$ となった場合は、 $\Phi = 0$ と補正します。

```
11 fvScalarMatrix snowDriftDensityEqn
12 (
13     fvm::ddt(snowDriftDensity)
14     + 1 / rho * fvm::div(phi, snowDriftDensity)
15     + Wf*gradS.component(vector::Z)
16     ==
17     1.0 / sigmaS *
18     (
19         gradN.component(vector::X) * gradS.component(vector::X) + turbulence->nut() * tensorS.component(tensor::XX)
20         + gradN.component(vector::X) * gradS.component(vector::Y) + turbulence->nut() * tensorS.component(tensor::XY)
21         + gradN.component(vector::X) * gradS.component(vector::Z) + turbulence->nut() * tensorS.component(tensor::XZ)
22
23         + gradN.component(vector::Y) * gradS.component(vector::X) + turbulence->nut() * tensorS.component(tensor::YX)
24         + gradN.component(vector::Y) * gradS.component(vector::Y) + turbulence->nut() * tensorS.component(tensor::YY)
25         + gradN.component(vector::Y) * gradS.component(vector::Z) + turbulence->nut() * tensorS.component(tensor::YZ)
26
27         + gradN.component(vector::Z) * gradS.component(vector::X) + turbulence->nut() * tensorS.component(tensor::ZX)
28         + gradN.component(vector::Z) * gradS.component(vector::Y) + turbulence->nut() * tensorS.component(tensor::ZY)
29         + gradN.component(vector::Z) * gradS.component(vector::Z) + turbulence->nut() * tensorS.component(tensor::ZZ)
30     )
31 );
```

図 2 飛雪空間密度の方程式
(カスタマイズソースコード「snowDriftDensityEqn.H」)

④雪面の摩擦速度 U_* を得ます。摩擦速度 U_* は一般化対数則を用いて次式により計算されます。

$$U_*^2 = \frac{\kappa C_\mu^{1/4} k_P^{1/2} U_P}{\ln \left(\frac{E C_\mu^{1/4} k_P^{1/2} y_P}{\nu} \right)}$$

u^* : 摩擦速度、 κ : フォンカルマン定数(=0.4187)、 k_P : 点Pにおける乱流運動エネルギー、
 U_P : 点Pにおける流体の平均速度、 E : 実験定数(=9.793)、 C_μ : 係数(=0.09)、 y_P : 点Pから壁面までの距離、 ν : 動粘性係数

```
21 | Ustar = Foam::pow
22 | (
23 |     mag(
24 |         0.4187 * Foam::pow(0.09,0.25) *
25 |         sqrt_k * mag(UMean_[faceCelli])
26 |     )
27 |     /
28 |     (
29 |         Foam::log
30 |         (
31 |             9.793 * Foam::pow(0.09,0.25) *
32 |             sqrt_k * yp[patch][facei]
33 |         )
34 |         /
35 |         (t_nu[faceCelli] + SMALL)
36 |     )
37 |     + SMALL
38 | )
39 | 0.5
40 | );
```

図 3 摩擦速度 U_* の算出
(カスタマイズソースコード「snowDepthEqn.H」)

- ⑤摩擦速度 U^* 、限界摩擦速度 U^*_c により、積雪の侵食・堆積フラックスを得ます。侵食・堆積フラックスにより、経過時間 Δt 、積雪密度 ρ_s を用いて積雪変化量(Δz)に換算されます。なお、侵食フラックスが発生した場合は、飛雪空間密度 Φ に侵食量が還元されます。

```

42 if (Ustar <= Utstar.value())
43 {
44     // deposition
45     q_dep = 0.5
46         * mag(snowDriftDensity.boundaryFieldRef()[patch][facei])
47         * (
48             (Foam::pow(Utstar.value(), 2) - Foam::pow(Ustar, 2))
49             /
50             Foam::pow(Utstar.value(), 2)
51         );
52     // total
53     q_tot = q_dep;
54
55     // change in snowDepth
56     deltaZ = q_tot * dtt / rho_s.value();
57
58     //Info<< "snowDriftDensity " << snowDriftDensity.boundaryFieldRef()[patch][facei] << nl;
59 }
60 else
61 {
62     // No snow deposition, snow erosion occur
63     // erosion (Ustar > Utstar.value())
64     q_er = -(1.0e-5) * 916.7 * Ustar
65         * ( 1 - Foam::pow(Utstar.value(), 2) /
66             (Foam::pow(Ustar, 2) + SMALL)
67         );
68
69     // Decrease in snow depth due to erosion
70     deltaZ_er = abs(q_er * dtt / rho_s.value());
71
72     //If the amount of erosion is greater than the amount of existing snow,
73     // only the amount of existing snow is eroded.
74
75     if ( deltaZ_er < snowDepth.boundaryFieldRef()[patch][facei])
76     {
77         q_er = q_er;
78     }
79     else
80     {
81         q_er = -1 * snowDepth.boundaryFieldRef()[patch][facei] * rho_s.value();
82     }
83
84     // total
85     q_tot = q_er;
86
87     // change in snowDepth
88     deltaZ = q_tot * dtt / rho_s.value();
89
90     //reintroduce the snow contribution from erosion in the snowDrift field
91     snowDriftDensity.boundaryFieldRef()[patch][facei] =
92     snowDriftDensity.boundaryFieldRef()[patch][facei]
93     + q_tot * dtt / (Foam::pow(mesh.magSf().boundaryField()[patch][facei], 0.5));
94
95 }

```

図 4 積雪変化量 Δz の算出
(カスタマイズソースコード「snowDepthEqn.H」)

- ⑥雪面の積雪量を更新します。仮に、積雪量がマイナスになった場合は、積雪量をゼロに修正します。

```
98 // test snowDriftDensity patch variation
99
100 snowDepth.boundaryFieldRef()[patch][facei] = snowDepth.boundaryFieldRef()[patch][facei] + deltaZ;
101
102 if (snowDepth.boundaryFieldRef()[patch][facei] < 0.0)
103 {
104     snowDepth.boundaryFieldRef()[patch][facei] = 0.0;
105 }
106
```

図 5 雪面の積雪量の更新
(カスタマイズソースコード「snowDepthEqn.H」)

- ⑦積雪重量を得ます。積雪重量は、次式により算出されます。

$$W = h_i \times A_i \times \rho_s$$

W : 積雪重量、 h_i : 雪面 i の積雪量、 A_i : 雪面 i の面積、 ρ_s : 積雪密度

```
109 // compute snow mass on facei
110 snowMass.boundaryFieldRef()[patch][facei] =
111     // depth of snow
112     snowDepth.boundaryFieldRef()[patch][facei]
113     // area of each face belonging to the patch
114     * mesh.magSf().boundaryField()[patch][facei]
115     // density of the snow
116     * rho_s.value();
117
```

図 6 雪面の積雪重量の算出
(カスタマイズソースコード「snowDepthEqn.H」)

- ⑧ ステップ③～⑦を設定した回数(終了時間、時間間隔で設定)繰り返します。

3.2 snowMelt

ここでは、融雪現象を組み込んだ熱輸送解析ソルバー「snowMelt」の雪面熱収支による融雪に関する解析手法について説明します。

なお、融雪に関する気流解析に用いる定常乱流解析ソルバー「buoyantsimpleFoam」に関する基礎方程式や解析手法については、各種参考資料を参照してください。

- ①気流解析である定常乱流解析ソルバー「buoyantsimpleFoam」でセル界面の物性値を計算しておきます。定常計算であるため、計算が安定するイタレーション数(反復回数)を設定しておく必要があります。

この定常計算は、各時刻で融雪量を計算するごとに計算を行います。

- ②「snowMelt」で使用する気流条件(速度 U 、圧力 p など)は、①で算出された物性値を設定します。

- ③雪面から 1m 上の座標を取得し、その位置での速度 U を取得します。

```
17 // compute center of the patch
18 vector centerPatch = sumFacesCenters / (totFacesOfPatch * Pstream::nProcs());
19
20 // center of the chosen patch:
21 //Info<< "\tCenter of patch " << mesh.boundary()[patch].name() << " is " << centerPatch << nl;
22
23 // print velocity of point 1 meter above the center of the patch
24 vector probePoint = centerPatch;
25 // the probe point will be 1 meter above the center of the patch therefore
26 probePoint.z() = probePoint.z() + 1.0;
27
28 //Info<< "\tProbe point position for patch " << mesh.boundary()[patch].name() << " is " << probePoint << nl;
29
30 // find cell containing the probe:
31 label cell_i = mesh.findCell(probePoint);
32
33 //Info<< "Cell ID containing the probe is " << cell_i << nl;
34 // retrieve velocity magnitude at 1 meter above the center of the patch
35 Info<< "Velocity magnitude 1m over " << mesh.boundary()[patch].name() << " patch is " << mag(U[cell_i]) << nl;
36
37 UoneMetUp = mag(U[cell_i]) + SMALL;
```

図 7 雪面から 1m 上の速度を取得
(カスタマイズソースコード「snowMeltEqn.H」)

④雪面への入力放射量を得ます。

```

102 // input radiation to snow surface
103 R_in = (1 - albedoSnow) *
104 // heat flux from visible spectral region
105 (
106     qprimaryRad_0.boundaryFieldRef()[patch][facei]
107     +
108     qprimaryRad_1.boundaryFieldRef()[patch][facei]
109 )
110 +
111 // heat flux from infrared spectral region
112 qprimaryRad_1.boundaryFieldRef()[patch][facei]
113 * eps_Snow; // * eps_atmosphere;
114 //Info<< "R_in " << R_in << nl;

```

図 8 入力放射量 R_{\downarrow} の算出
(カスタマイズソースコード「snowMeltEqn.H」)

⑤雪面への上向きの長波放射量を得ます。長波放射量の計算には、積雪表面温度 T_s 、飽和比湿 q_{sat} 、飽和比湿の温度 T に対する変化率 Δ を求めておく必要がある。

```

116 e_sat = 6.108 * Foam::pow(
117     (
118         7.5 * (T_ref-273.15)
119         /
120         (237.5 + (T_ref-273.15))
121     )
122 );
123
124
125 p_s = 101325;
126
127 // saturated specific humidity
128 q_sat = 0.622 * e_sat / ( p_s - 0.378 * e_sat);
129 //Info<< "q_sat " << q_sat << nl;
130
131 // rate of change of saturated specific humidity with temperature change
132 deltaSatHum = 4098.03
133 //
134 Foam::pow(
135     (237.3 + (T_ref-273.15) + VSMALL), 2
136 )
137 * e_sat;
138 //Info<< "deltaSatHum " << deltaSatHum << nl;
139
140 // snow surface temperature
141 T_s = (
142     R_in - eps_Snow * sigmaSB * Foam::pow(T_ref, 4)
143     - i_subl * rho_air * C_h * UoneMetUp * (1-reHu) * q_sat
144 )
145 /
146 (
147     4 * eps_Snow * sigmaSB * Foam::pow(T_ref, 3)
148     + (i_subl * deltaSatHum + Cp_air)
149     * rho_air * C_h * UoneMetUp
150 )
151 + T_ref;
152 //Info<< "T_s " << T_s << nl;
153
154 // upward long-wavelength radiation
155 L_out = eps_Snow * sigmaSB * Foam::pow(T_s, 4);
156 //Info<< "L_out " << L_out << nl;

```

図 9 上向き長波放射量 L_{\uparrow} の算出
(カスタマイズソースコード「snowMeltEqn.H」)

- ⑥顕熱輸送量、潜熱輸送量を求めます。長波放射量同様に積雪表面温度 T_s などを求めておく必要があります。

```
159 // sensible heat flux
160 H_sensFlux = Cp_air * rho_air * C_h * UoneMetUp * (T_s-T_ref);
161 //Info<< "H_sensFlux " << H_sensFlux << nl;
162
163 // latent heat flux
164 E_latentFlux = i_subl * rho_air * C_e * UoneMetUp *
165 ( (1-reHu) * q_sat + deltaSatHum * (T_s - T_ref) );
166 //Info<< "E_latentFlux " << E_latentFlux << nl;
```

図 10 顕熱輸送量 H、潜熱輸送量 IE の算出
(カスタマイズソースコード「snowMeltEqn.H」)

- ⑦各時刻の太陽放射で発生する融雪エネルギーによる積雪量の変化を求めます。

```
177 // snow depth change due to the snow evaporation occurring du
178 deltaZ = (R_in - L_out + H_sensFlux + E_latentFlux) * 3600
179         / ( rho_s * I_c );
```

図 11 融雪による積雪量の変化
(カスタマイズソースコード「snowMeltEqn.H」)

- ⑧雪面の積雪量を更新します。仮に、積雪量がマイナスになった場合は、積雪量をゼロに修正します。

```
193 snowDepth.boundaryFieldRef()[patch][facei] = snowDepth.boundaryFieldRef()[patch][facei] - deltaZ.value();
194
195 if (snowDepth.boundaryFieldRef()[patch][facei] < 0.0)
196 {
197     snowDepth.boundaryFieldRef()[patch][facei] = 0.0;
198 }
```

図 12 融雪による雪面の積雪量の更新
(カスタマイズソースコード「snowMeltEqn.H」)

⑨積雪重量を得ます。

```
200 // compute snow mass on facei
201 snowMass.boundaryFieldRef()[patch][facei] =
202     // depth of snow
203     snowDepth.boundaryFieldRef()[patch][facei]
204     // area of each face belonging to the patch
205     * mesh.magSf().boundaryField()[patch][facei]
206     // density of the snow
207     * rho_s.value();
---
```

図 13 雪面の積雪重量の算出
(カスタマイズソースコード「snowMeltEqn.H」)

⑧ ステップ①～⑨を設定した時刻分、繰り返します。

4. 物性値

OpenFOAMでは流体の物性値を設定するファイルとして、非圧縮性流体の解析では「transportProperties」ファイル、圧縮性流体の解析では「thermophysicalProperties」ファイルを使用します。詳細については、各種参考資料を確認してください。

ここでは、「snowFoam」「snowMelt」でのカスタマイズで追加した、雪に関する物性値を設定する「snowProperties」ファイルについて示します。以下のファイルを参照してください。

../Case/constant/snowProperties

4.1 snowFoam

風雪シミュレーションのカスタマイズソルバー「snowFoam」では、積雪に関する物性値を設定できます。

```
17 // snow density [kg/m^3]
18 rho_s      100;
19
20 Utstar     0.25;
21
22 // Turbulent Schmidt number to be used in the snow drift density transport equation
23 sigmaS     1.0;
24
25 // snow fall velocity [m/s]
26 Wf         -0.5;
27
28 // set the following keyword to yes in order to compute snowDepth and snowMass on ground and roof patches ONLY.
29 ground_and_roofs_only yes; // no;
30 // If set to "no" snowDepth and snowMass fields will be computed on ALL wall patches inside the fluid domain.
31
```

図 14 雪に関する物性値 (snowFoam)
(物性値の設定ファイル「snowProperties」)

表 3 「snowProperties」の設定値 (snowFoam)

設定項目	初期値	意 味
rho_s	100.0 [kg/m ³]	積雪密度
Utstar	0.25	限界摩擦速度
sigmaS	1.0	乱流シュミット数
Wf	-0.5 [m/s]	雪粒子の落下速度
ground_and_roofs_only	yes	雪面の指定、基本的に yes にしておく yes : 「ground」「roof」で始まる面を雪面とする no : すべての面を雪面とする

4.2 snowMelt

融雪シミュレーションのカスタマイズソルバー「snowFoam」では、融雪に関する物性値を設定できます。

```
17 // albedo of the snow
18 albedoSnow 0.6;
19
20 // bulk transport coefficient of sensible heat
21 C_e 0.002;
22 C_h 0.002;
23
24 // specific heat of air
25 Cp_air 1004.4; // [J/(kg K)]
26
27 // emissivity of snow surface
28 eps_Snow 0.98; // snow is nearly a blackbody
29
30 // set the following keyword to yes in order to compute snowDepth and snowMass on ground and roof patches ONLY.
31 ground_and_roofs_only yes; // no;
32 // If set to "no" snowDepth and snowMass fields will be computed on ALL wall patches inside the fluid domain.
33
34 // heat of fusion of ice
35 I_c 334000; // [J/kg]
36
37 // Latent heat for ice sublimation [J/kg]
38 i_subl 2838000; // as reported in Springer "Encyclopedia of Snow, Ice and Glaciers pp 703" https://link.s
39
40 // relative humidity
41 reHu 0.5;
42
43 // Turbulent Schmidt number to be used in the snow drift density transport equation
44 sigmaS 1.0;
45
46 // air density [kg/m^3] it is estimated in the code as dependent on T_ref and P_ref
47 // rho_air 1.007; // 2000m above sea level
48 // rho_air 1.112; // 1000m above sea level
49 rho_air 1.225; // at sea level
50
51 // snow density [kg/m^3]
52 rho_s 450;
53 // snow density usually between 0.1 to 0.8 g/cm^3
54 // https://en.wikipedia.org/wiki/Snow
55 // in this test we take the average value (0.1+0.8)/2 = 0.45 g/cm^3
56
57 // reference temperature of the air [in K]
58 T_ref 283.15; //10.0[C]
59
60 // snow fall velocity [m/s]
61 Wf -0.5;
```

図 15 雪に関する物性値 (snowMelt)
(物性値の設定ファイル「snowProperties」)

表 4 「snowProperties」の設定値(snowMelt)

設定項目	初期値	意 味
albedoSnow	0.6	アルベド数
C_e、C_h	0.002	潜熱・顕熱のバルク係数
Cp_air	1004.4 [J/(kg K)]	空気の比熱
eps_Snow	0.98	積雪面の放射率
ground_and_roofs_only	yes	雪面の指定、基本的に yes にしておく yes : 「ground」「roof」で始まる面を雪面とする no : すべての面を雪面とする
L_c	334,000 [J/kg]	氷の融解熱
i_subl	2,838,000 [J/kg]	氷の昇華潜熱/水の昇化潜熱
reHu	0.5	相対湿度
sigmaS	1.0	乱流シュミット数
rho_air	1.225 [kg/m ³]	空気の密度
rho_s	450	積雪の密度(=根雪)
T_ref	283.15 [K]	大気の温度(=10℃)
P_s	1013.25[hPa]	大気圧

5. 境界条件

OpenFOAMでは物性値の境界条件をフィールド変数ファイルのboundaryField辞書内に記述します。フィールド変数ファイルについては、以下のフォルダを参照してください。

■気流解析について

../Case/0.orig_steady

■風雪解析について

../Case/0.orig_transient

なお、融雪現象を組み込んだ熱輸送解析ソルバー「snowMelt」では、風雪現象を組み込んだ非定常乱流解析ソルバー「snowFoam」の計算結果を参照しています。そのため、「snowMelt」で雪に関する境界条件を設定することはありません。

以上より、ここでは雪に関するフィールド変数ファイル「積雪量：snowDepth」「飛雪空間密度：snowDriftDensity」「積雪重量：snowMass」について示します。気流解析の設定については、各種参考資料を参照してください。

5.1 パッチの設定

フィールド変数ファイルのboundaryField辞書では、境界パッチの各面中心に定義されるfvPatchField型のフィールド変数に対する設定を示します。

```
23 boundaryField
24 {
25     //- Set patchGroups for constraint patches
26     #include "${WM_PROJECT_DIR}/etc/caseDicts/setConstraintTypes"
27
28     boundaryInlet
29     {
30         type            zeroGradient;
31     }
32
33     boundaryOutlet
34     {
35         type            fixedValue;
36         value           $internalField;
37     }
38
39     roof
40     {
41         type            zeroGradient;
42     }
43
44     wall
45     {
46         type            zeroGradient;
47     }
48 }
```

図 16 圧力 p の設定 (一部抜粋)

パッチ名称「boundaryInlet」「boundaryOutlet」は流入出境界面を示しており、メッシュを作成する際に名称や境界位置を定義しています。

※参照：../Case/system/blockMeshDict

```
52 boundary
53 (
54     boundaryInlet
55     {
56         type patch;
57         faces
58         (
59             (0 4 7 3)
60         );
61     }
62     boundaryOutlet
63     {
64         type patch;
65         faces
66         (
67             (1 2 6 5)
68         );
69     }
70 )
```

図 17 blockMeshDict の設定 (一部抜粋)

パッチ名称「wall」「roof」などは流入境界と同様にメッシュを作成する段階で名称や境界位置を定義していますが、地形(STL形式の構造物データ)ファイルの値を参照しています。

以下の例で、「terrain_wadayama.stl」というSTLデータを参照しており、事前に設定したsolid名称ごとにパッチ名称を割り当てています。

※参照：../Case/system/snappyHexMeshDict
../Case/constant/triSurface/〇〇.stl

```
24 geometry
25 {
26     terrain_wadayama.stl
27     {
28         type triSurfaceMesh;
29         name terrain_wadayama ;
30         regions
31         {
32             roofs
33             {
34                 name roof;
35             }
36             walls
37             {
38                 name wall;
39             }
40             grounds
41             {
42                 name ground;
43             }
44         }
45     }
```

図 18 snappyHexMeshDict の設定
(STL データの solid 名称からパッチ名称を割り当て)

```
1 solid roofs
2 facet normal 0 0 1
3 outer loop
4 vertex 47574.9 -76048.3 129.5
5 vertex 47571.2 -76046.7 129.5
6 vertex 47571.4 -76048.7 129.5
7 endloop
8 endfacet
9 facet normal 0 0 1
10 outer loop
11 vertex 47574.9 -76048.3 129.5
12 vertex 47574.7 -76046.3 129.5
13 vertex 47571.2 -76046.7 129.5
14 endloop
15 endfacet
16 facet normal 0 0 1
```

図 19 STL データの設定
(「solid 〇〇」というように設定しておく)

5.2 境界タイプの概要

ここでは、フィールド変数ファイル「積雪量：snowDepth」「飛雪空間密度：snowDriftDensity」「積雪重量：snowMass」で設定されている境界タイプについて示します。

(1) fixedValue

a) 概 要

境界値を固定値として指定する

b) 設定項目

設定項目	設定値	意 味
type	fixedValue	固定値として指定
value	0	

(2) zeroGradient

a) 概 要

内部セル値を境界値とする

b) 設定項目

設定項目	設定値	意 味
type	zeroGradient	内部セル値を境界値とする

5.3 積雪量 : snowDepth

積雪量の境界条件を設定した「snowDepth」ファイルは、以下のように記述します。

```
23 boundaryField
24 {
25     //- Set patchGroups for constraint patches
26     #include "${WM_PROJECT_DIR}/etc/caseDicts/setConstraintTypes"
27
28     boundaryInlet
29     {
30         type            fixedValue;
31         value            uniform 0;
32     }
33
34     boundaryOutlet
35     {
36         type            zeroGradient;
37     }
38
39     wall
40     {
41         type            fixedValue;
42         value            uniform 0;
43     }
44
45     roof
46     {
47         type            fixedValue;
48         value            uniform 0;
49     }
```

図 20 積雪量 snowDepth の設定 (一部抜粋)

表 5 境界パッチごとの境界タイプ (snowDepth)

境界パッチ	概 要	タイプ名	目 的
boundaryInlet	流入境界	fixedValue	境界値を固定値として指定
boundaryOutlet	流出境界	zeroGradient	内部セル値を境界値とする
その他	その他の面	fixedValue	境界値を固定値として指定

5.4 飛雪空間密度：snowDriftDensity

飛雪空間密度の境界条件を設定した「snowDriftDensity」ファイルは、以下のように記述します。

```
23 boundaryField
24 {
25
26     // snow entering both from inlet and top patches:
27     boundaryInlet
28     {
29         type            fixedValue;
30         value            uniform $snowDriftDensityInlet;
31     }
32
33     boundaryOutlet
34     {
35         type            zeroGradient;
36     }
37
38     wall
39     {
40         type            zeroGradient;
41     }
42
43     roof
44     {
45         type            fixedValue;
46         value            uniform $snowDriftDensityInlet;
47         //type            zeroGradient;
48     }
```

図 21 飛雪空間密度 snowDriftDensity の設定(一部抜粋)

表 6 境界パッチごとの境界タイプ(snowDriftDensity)

境界パッチ	概 要	タイプ名	目 的
boundaryInlet	流入境界	fixedValue	境界値を固定値として指定
boundaryOutlet	流出境界	zeroGradient	内部セル値を境界値とする
roof	雪面(屋根面)	fixedValue	境界値を固定値として指定
ground	雪面(地面)	fixedValue	境界値を固定値として指定
その他	その他の面	zeroGradient	内部セル値を境界値とする

5.5 積雪重量 : snowMass

積雪重量の境界条件を設定した「snowMass」ファイルは、以下のように記述します。

```
23 boundaryField
24 {
25
26     // snow entering both from inlet and top patches:
27     boundaryInlet
28     {
29         type            fixedValue;
30         value            uniform $snowDriftDensityInlet;
31     }
32
33     boundaryOutlet
34     {
35         type            zeroGradient;
36     }
37
38     wall
39     {
40         type            zeroGradient;
41     }
42
43     roof
44     {
45         type            fixedValue;
46         value            uniform $snowDriftDensityInlet;
47         //type            zeroGradient;
48     }
```

図 22 積雪重量 snowMass の設定 (一部抜粋)

表 7 境界パッチごとの境界タイプ (snowMass)

境界パッチ	概 要	タイプ名	目 的
boundaryInlet	流入境界	fixedValue	境界値を固定値として指定
boundaryOutlet	流出境界	fixedValue	境界値を固定値として指定
その他	その他の面	fixedValue	境界値を固定値として指定