

OpenFOAM LES乱流モデルカスタマイズ

yotakagi77

オープンCAE勉強会@関西

2015年6月13日, 大阪大学

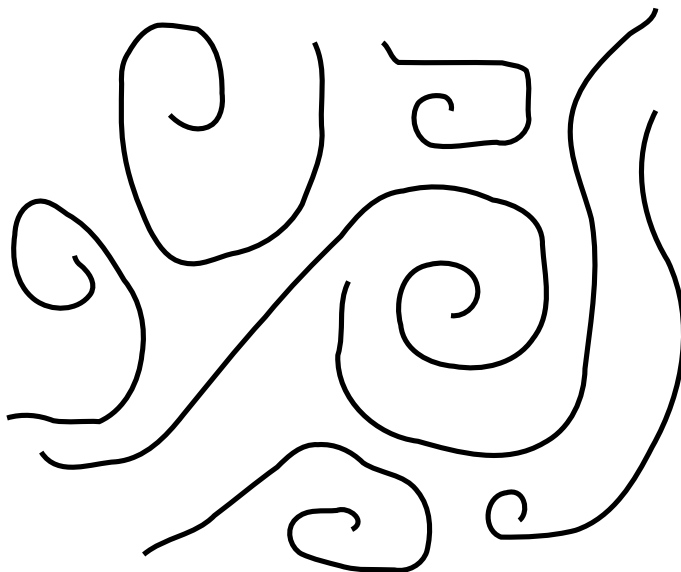
Agenda

- 乱流モデルの基礎知識
- テンソル数学
- 演習1: WALEモデルのコンパイル・実行
- 演習2: コヒーレント構造Smagorinskyモデルの実装
- 実習課題

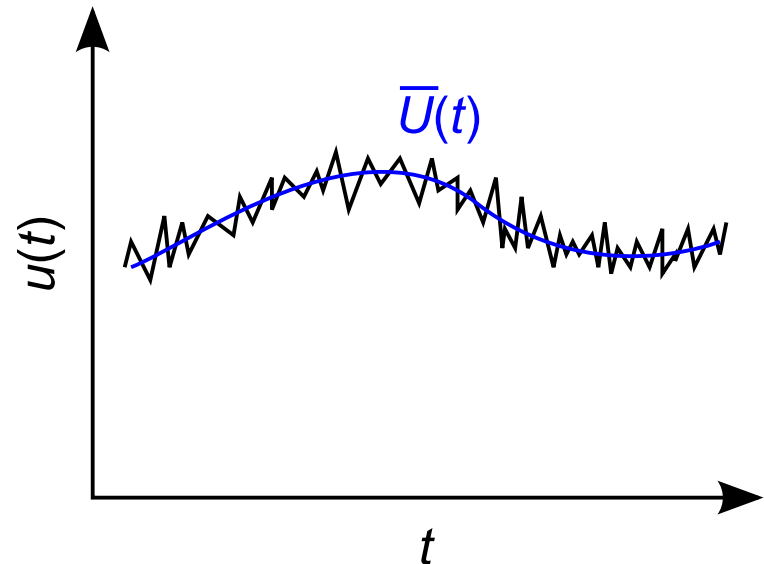
乱流モデルの基礎知識

乱流シミュレーション

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



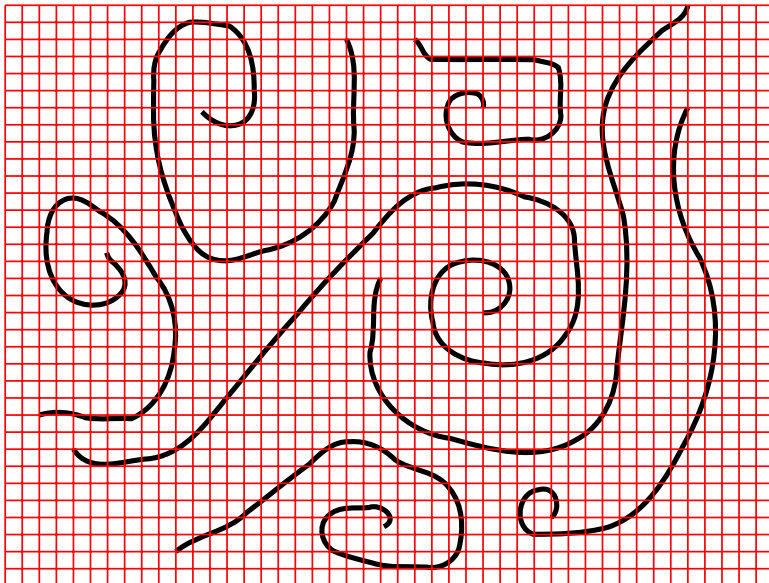
渦流れ



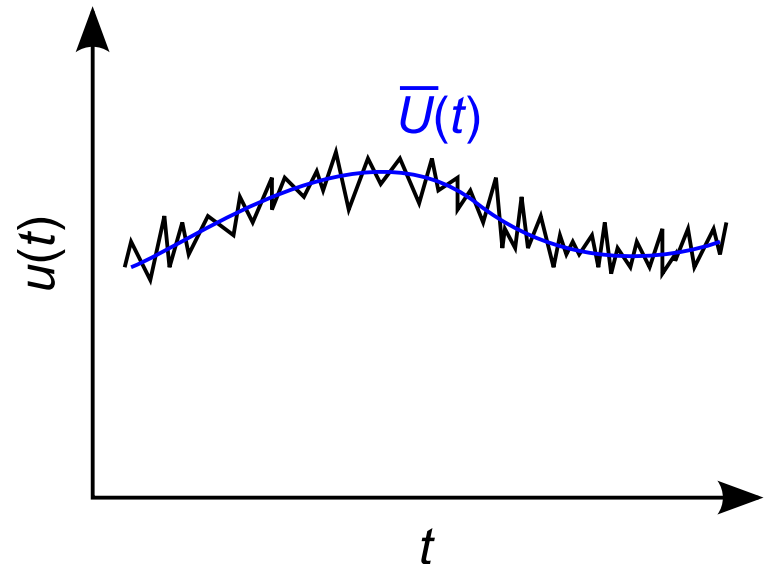
レイノルズ平均

乱流シミュレーション

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



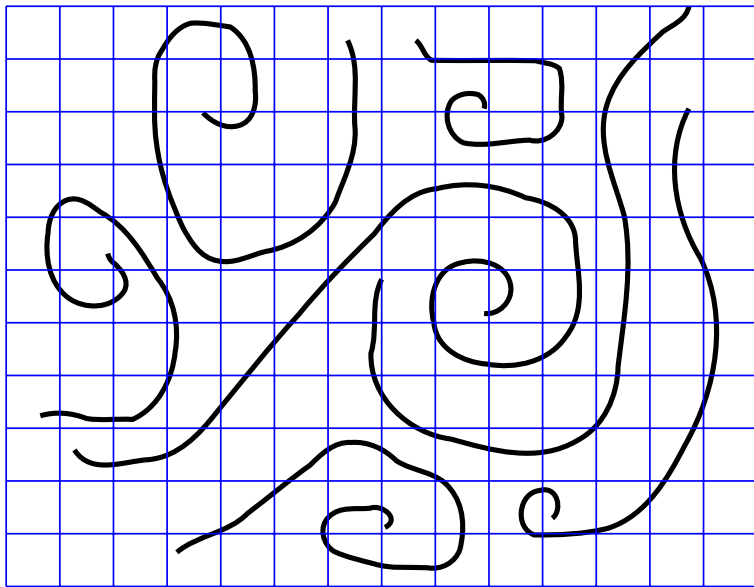
DNSでの格子, u



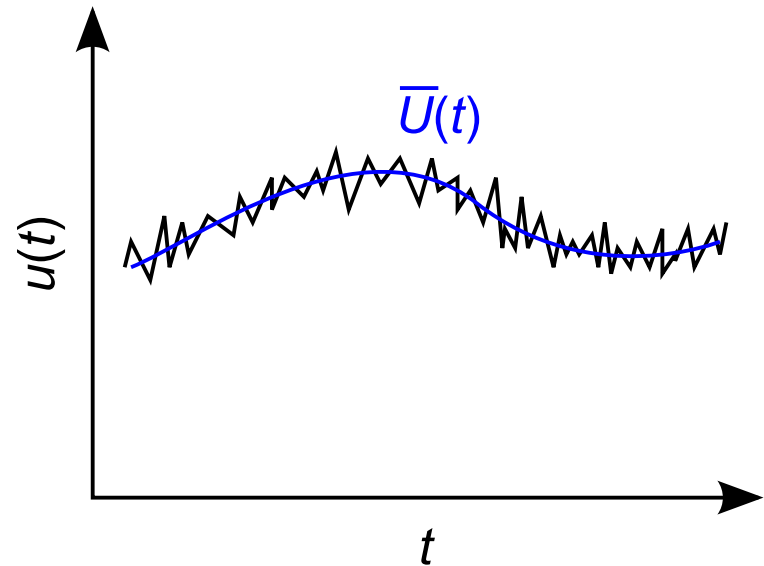
レイノルズ平均

乱流シミュレーション

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



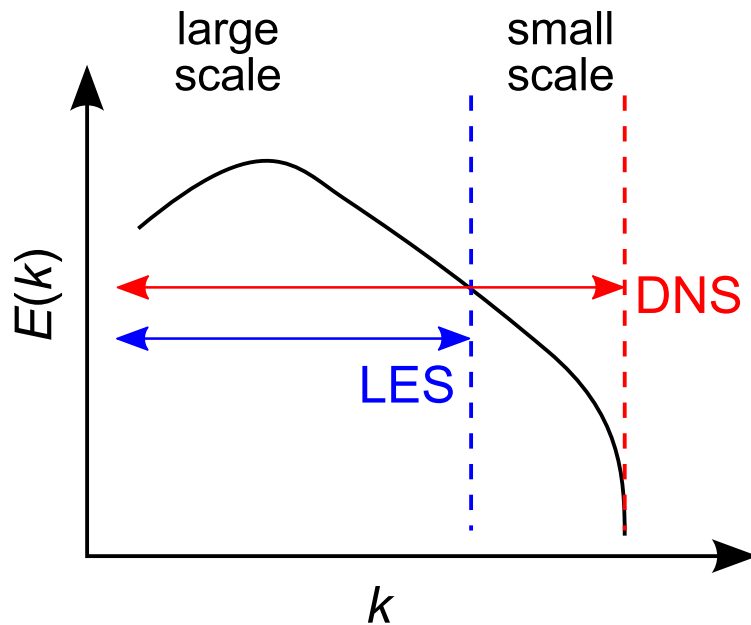
LESでの格子, $\bar{u} = u - u'$



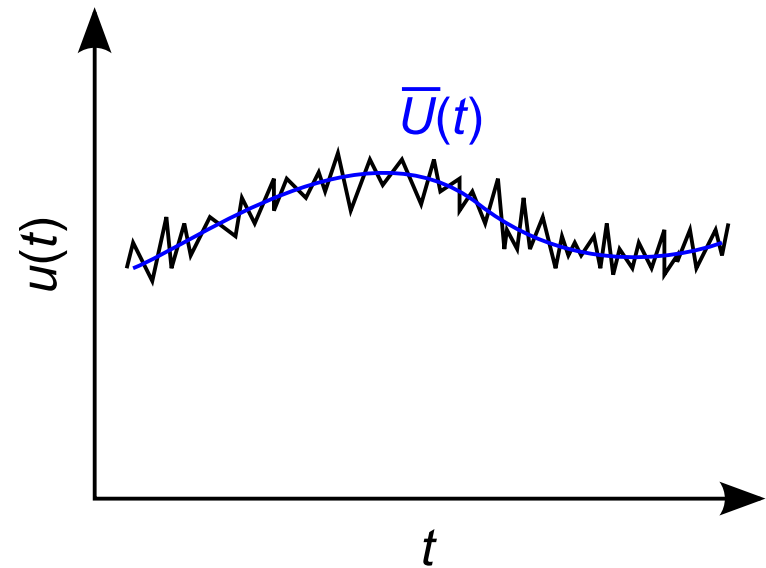
レイノルズ平均

乱流シミュレーション

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



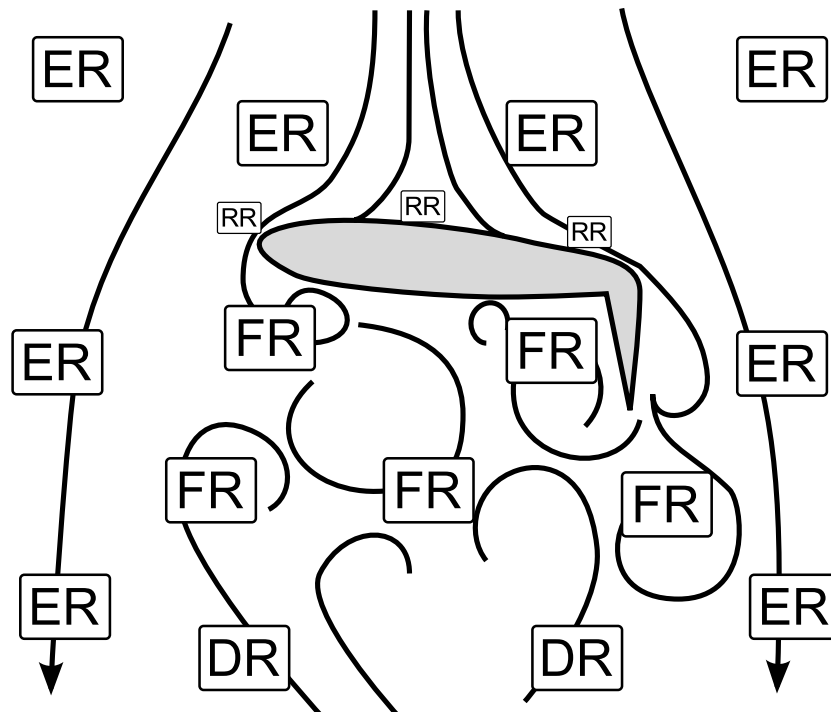
フィルター操作



レイノルズ平均

Detached-eddy simulation (DES)

- P. R. Spalart (1997):
 - *We name the new approach “Detached-Eddy Simulation” (DES) to emphasize its distinct treatments of attached and separated regions.*

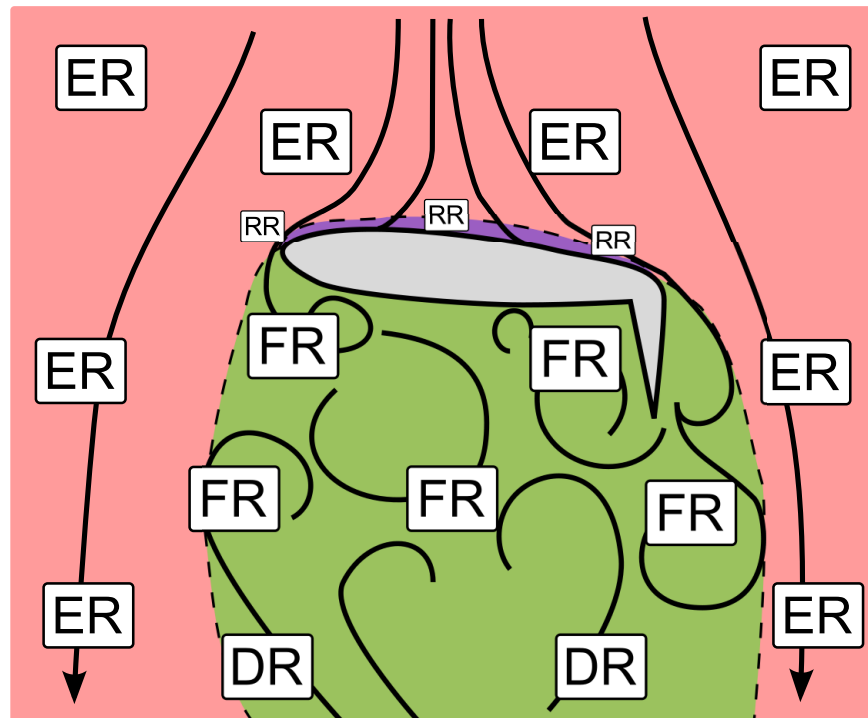


Spalart (2001)

Super-Region	Region
Euler (ER)	
RANS (RR)	Viscous (VR)
	Outer (OR)
LES (LR)	Viscous (VR)
	Focus (FR)
	Departure (DR)

Detached-eddy simulation (DES)

- P. R. Spalart (1997):
 - *We name the new approach “Detached-Eddy Simulation” (DES) to emphasize its distinct treatments of attached and separated regions.*



Spalart (2001)

Super-Region	Region
Euler (ER)	
RANS (RR)	Viscous (VR)
	Outer (OR)
LES (LR)	Viscous (VR)
	Focus (FR)
	Departure (DR)

運動量輸送方程式の渦粘性係数を介した連結

- RANS

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla \cdot \left((\mathbf{v} + \underline{\mathbf{v}_t}) (\nabla \mathbf{U} + (\nabla \mathbf{U})^T) \right) = \nabla p$$

乱流粘性

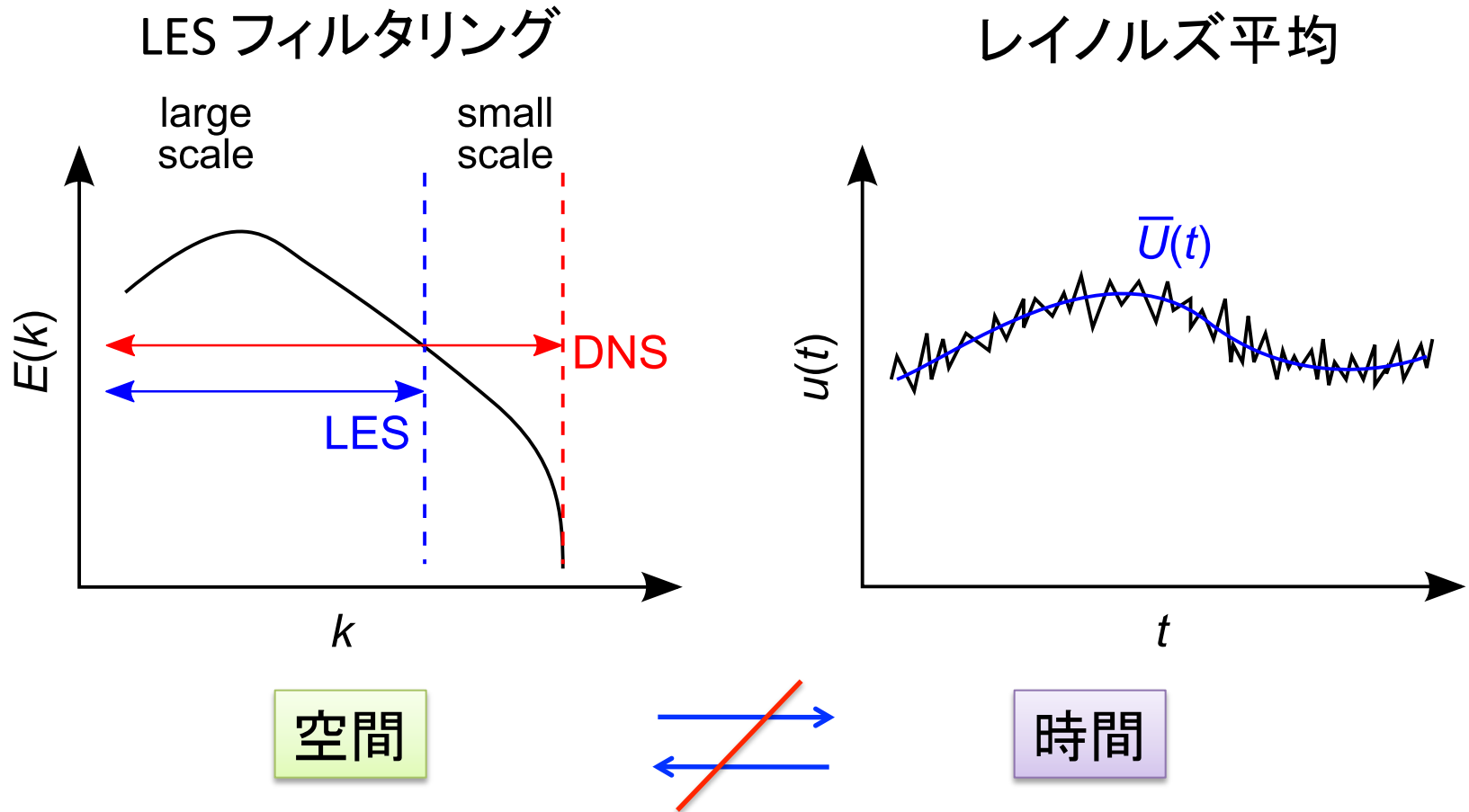
- LES

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla \cdot \left((\mathbf{v} + \underline{\mathbf{v}_{SGS}}) (\nabla \mathbf{U} + (\nabla \mathbf{U})^T) \right) = \nabla p$$

サブグリッドスケール粘性

粘性係数を変えるだけ!

問題:フィルター(平均化)操作の違い



LES領域-RANS領域界面での不整合

OpenFOAMでの標準SGSモデル

Library name	Note
Smagorinsky	Smagorinsky model
Smagorinsky2	Smagorinsky model with 3-D filter
homogeneousDynSmagorinsky	Homogeneous dynamic Smagorinsky model
dynLagrangian	Lagrangian two equation eddy-viscosity model
scaleSimilarity	Scale similarity model
mixedSmagorinsky	Mixed Smagorinsky / scale similarity model
homogeneousDynOneEqEddy	One Equation Eddy Viscosity Model for incompressible flows
laminar	Simply returns laminar properties
kOmegaSSTAS	k - ω SST scale adaptive simulation (SAS) model

OpenFOAMでの標準SGSモデル

Library name	Note
oneEqEddy	k -equation eddy-viscosity model
dynOneEqEddy	Dynamic k -equation eddy-viscosity model
spectEddyVisc	Spectral eddy viscosity model
LRDDiffStress	LRR differential stress model
DeardorffDiffStress	Deardorff differential stress model
SpalartAllmaras	Spalart-Allmaras model
SpalartAllmarasDDES	Spalart-Allmaras delayed detached eddy simulation (DDES) model
SpalartAllmarasIDDES	Spalart-Allmaras improved DDES (IDDES) model
vanDriestDelta	Simple cube-root of cell volume delta used in incompressible LES models

テンソル数学

テンソル

- ランク 0: 'スカラー', 例: 体積 V , 圧力 p .
- ランク 1: 'ベクトル', 例: 速度ベクトル \mathbf{u} , 面ベクトル \mathbf{S} . 表記: $\mathbf{a} = a_i = (a_1, a_2, a_3)$.
- ランク 2: 'テンソル', 例 ひずみ速度テンソル S_{ij} , 回転テンソル Ω_{ij} .

表記:

$$\mathbf{T} = T_{ij} = \begin{pmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{pmatrix}$$

対称/非対称テンソル

- 速度勾配テンソルはひずみ速度テンソル(対称テンソル)と渦度テンソル(非対称テンソル)に分解できる.

$$D_{ij} = \frac{\partial u_i}{\partial x_j}, S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)$$

$$D_{ij} = S_{ij} + \Omega_{ij}$$

- 乱流モデリングでは, s_{ij} と Ω_{ij} がよく用いられる.

テンソルに対する数学操作

$$\mathbf{T} = \frac{1}{2}(\mathbf{T} + \mathbf{T}^T) + \frac{1}{2}(\mathbf{T} - \mathbf{T}^T) = \text{symm } \mathbf{T} + \text{skew } \mathbf{T},$$

$$\text{tr } \mathbf{T} = T_{11} + T_{22} + T_{33},$$

$$\text{diag } \mathbf{T} = (T_{11}, T_{22}, T_{33}),$$

$$\mathbf{T} = \mathbf{T} - \frac{1}{3}(\text{tr } \mathbf{T})\mathbf{I} + \frac{1}{3}(\text{tr } \mathbf{T})\mathbf{I} = \text{dev } \mathbf{T} + \text{hyd } \mathbf{T},$$

$$\det \mathbf{T} = \begin{vmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{vmatrix}$$

OpenFOAMのテンソルクラス

演算	数学表記	クラス
加算	$a + b$	<code>a + b</code>
引算	$a - b$	<code>a - b</code>
スカラー乗算	sa	<code>s * a</code>
スカラー割り算	a / s	<code>a / s</code>
外積	$a b$	<code>a * b</code>
内積	$a \cdot b$	<code>a & b</code>
二重内積	$a : b$	<code>a && b</code>
クロス積	$a \times b$	<code>a ^ b</code>
平方	a^2	<code>sqr(a)</code>
絶対値平方	$ a ^2$	<code>magSqr(a)</code>
絶対値	$ a $	<code>mag(a)</code>
べき乗	a^n	<code>pow(a, n)</code>

OpenFOAM tensor classes

Operation	Mathematical	Class
転置	\mathbf{T}^T	T.T()
対角	diag \mathbf{T}	diag(T)
トレース	tr \mathbf{T}	tr(T)
偏差成分	dev \mathbf{T}	dev(T)
対称成分	symm \mathbf{T}	symm(T)
非対称成分	skew \mathbf{T}	skew(T)
行列式	det \mathbf{T}	det(T)
余因子	cof \mathbf{T}	cof(T)
逆行列	inv \mathbf{T}	inv(T)

演習1: WALEモデルのコンパイル・実行

非圧縮LESの基礎方程式

- 連続の式・ナビエーストークス方程式

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0,$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_i} (-\tau_{ij} + 2\nu \bar{S}_{ij})$$

ここで、

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$$

SGS渦粘性モデル

- 乱流エネルギーの分解

$$\bar{k} = \frac{1}{2} \overline{u_k u_k} = \frac{1}{2} \overline{\bar{u}_k \bar{u}_k} + \frac{1}{2} \overline{(u_k u_k - \bar{u}_k \bar{u}_k)}$$

k_{GS} k_{SGS}

- GS乱流エネルギー k_{GS} の輸送方程式

$$\frac{\partial k_{GS}}{\partial t} + \bar{u}_j \frac{\partial k_{GS}}{\partial x_j} = \tau_{ij} \bar{S}_{ij} - \varepsilon_{GS} + \frac{\partial}{\partial x_i} \left(-\bar{u}_i \tau_{ij} - \frac{\bar{p} \bar{u}_j}{\rho} + \nu \frac{\partial k_{GS}}{\partial x_j} \right)$$

- SGS乱流エネルギー k_{SGS} の輸送方程式

$$\frac{\partial k_{SGS}}{\partial t} + \bar{u}_j \frac{\partial k_{SGS}}{\partial x_j} = -\tau_{ij} \bar{S}_{ij} - \varepsilon_{SGS} + \frac{\partial}{\partial x_i} \left[\bar{u}_i \tau_{ij} - \frac{1}{2} \overline{(u_i u_i u_j + \bar{u}_j u_i u_i)} - \frac{\overline{p u_j} - \bar{p} \bar{u}_j}{\rho} + \nu \frac{\partial k_{SGS}}{\partial x_j} \right]$$

Smagorinskyモデル

- SGS生成速度とSGSエネルギー散逸の局所平衡仮定:

$$\varepsilon_{SGS} \left(\equiv \nu \overline{\frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j}} - \nu \frac{\partial \bar{u}_i}{\partial x_j} \frac{\partial \bar{u}_i}{\partial x_j} \right) = -\tau_{ij} \bar{S}_{ij}$$

- 渦粘性近似:

$$\tau_{ij}^a = -2\nu_{SGS} \bar{S}_{ij}$$

- 次元解析・スケーリングにより,

$$\nu_{SGS} = (C_S \Delta)^2 |\bar{S}|, |\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}, C_S : \text{Smagorinsky定数}$$

WALEモデル

(Nicoud and Ducros, 1999)

- 平方速度勾配テンソルのトレースフリー対称部分:

$$\begin{aligned} S_{ij}^d &= \frac{1}{2}(\bar{D}_{ij}^2 + \bar{D}_{ji}^2) - \frac{1}{3}\delta_{ij}\bar{D}_{kk}^2 \\ &= \bar{S}_{ik}\bar{S}_{kj} + \bar{\Omega}_{ik}\bar{\Omega}_{kj} - \frac{1}{3}\delta_{ij}[\bar{S}_{mn}\bar{S}_{mn} - \bar{\Omega}_{mn}\bar{\Omega}_{mn}] \\ S_{ij}^d S_{ij}^d &= \frac{1}{6}(S^2 S^2 + \Omega^2 \Omega^2) + \frac{2}{3}S^2 \Omega^2 + 2IV_{S\Omega}, \\ S^2 &= \bar{S}_{ij}\bar{S}_{ij}, \Omega^2 = \bar{\Omega}_{ij}\bar{\Omega}_{ij}, IV_{S\Omega} = \bar{S}_{ik}\bar{S}_{kj}\bar{\Omega}_{jl}\bar{\Omega}_{li} \end{aligned}$$

- WALEモデルでの渦粘性:

$$\nu_{SGS} = (C_w \Delta)^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\bar{S}_{ij}\bar{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}}$$

WALEモデルのパラメータ

(Nicoud and Ducros, 1999)

	Field a	Field b	Field c	Field d	Field e	Field f
C_w^2/C_s^2	10.81	10.52	10.84	10.55	10.70	11.27

If $C_s = 0.18$, $0.55 \leq C_w \leq 0.6$.

If $C_s = 0.1$, $0.32 \leq C_w \leq 0.34$.

モデルパラメータ C_w は Smagorinsky定数 C_s に依存するので、流れ場によって変更する必要がある。

WALEモデルのソースコード

- V&V委員会, オープンCAE学会

<https://github.com/opencae/VandV/tree/master/OpenFOAM/2.2.x/src/libraries/incompressibleWALE>

- OpenFOAM-dev

<https://github.com/OpenFOAM/OpenFOAM-dev/tree/master/src/TurbulenceModels/turbulenceModels/LES/WALE>

ダウンロード, コンパイル

1. WALEモデルのソースコードをオープンCAE学会のV&Vレポジトリよりダウンロードし, コンパイルする。

```
$ mkdir -p $FOAM_RUN
$ cd
$ git clone https://github.com/opencae/VandV
$ cd VandV/OpenFOAM/OpenFOAM-BenchmarkTest/
  channelReTau110
$ cp -r src $FOAM_RUN/..
$ run
$ cd ../src/libraries/incompressibleWALE
$ wmake libso
$ ls $FOAM_USER_LIBBIN
```

チャンネル流れ計算

2. $Re_\tau = 395$ のチャンネル流れのチュートリアルケースを自分のrunディレクトリにコピーする.

```
$ run  
$ cp -r $FOAM_TUTORIALS/incompressible/pimpleFoam/  
channel395/ ./ReTau395WALE  
$ cd ReTau395WALE
```

3. constant/LESProperties と system/controlDict を編集する.

```
$ gedit constant/LESProperties
```

```
LESModel          WALE;  
printCoeffs       on;  
delta              cubeRootVol;  
...
```

チャンネル流れの計算

```
$ gedit system/controlDict
```

```
...
```

```
libs ("libincompressibleWALE.so");
```

この1行は新しいWALEモデルライブラリをソルバーで利用するために必要.

4. その他の解析条件・パラメータを確認後, ソルバーを実行する.

```
$ ./Allrun
```

5. 計算が正常に終了したら, ログを確認し, ParaViewで流れ場を可視化してみる. また, postChannelで生成されたプロファイルをプロットしてみる.

$Re_\tau = 110$ でのチャンネル流れの計算

6. V&V委員会で提供されているテストケースを利用する場合は, templateケースをコピーし, 設定を編集する.

```
$ run
$ cp -r ~/VandV/OpenFOAM/OpenFOAM-BenchmarkTest/
  channelReTau110/template $FOAM_RUN/ReTau110WALE
$ cd ReTau110WALE
$ gedit caseSettings
```

```
controlDict
{
    deltaT          0.002;
    endTime         0.022;
    libs            "libincompressibleWALE.so";
}
```

$Re_\tau = 110$ でのチャンネル流れの計算

```
turbulenceProperties
{
  simulationType LESModel;
}

LESProperties
{
  LESModel WALE;
  delta cubeRootVol;
}
```

オリジナルの caseSettings は大規模計算環境でのDNS用なので、blockMeshDict や decomposeParDict のパラメータを変更したほうがよい。

$Re_{\tau} = 110$ でのチャンネル流れの計算

7. その他の解析条件やパラメータを確認後, ソルバーを実行する.

```
$ ./Allrun
```

8. If the solver calculation is normally finished, you check the logs and visualize the flow field with ParaView. If the integration time is not sufficient for the flow field to become fully developed state, run longer simulations.

演習2: コヒーレント構造Smagorinsky モデルの実装

SGSモデルのオリジナルソースコード

1. SGSモデルのオリジナルコードを確認する.

```
$ src  
$ cd turbulenceModels/incompressible/LES/  
$ ls
```

2. Smagorinskyのコードを見してみる.

```
$ gedit Smagorinsky/Smagorinsky.*
```

3. 比較のために、ダイナミックモデルのコードも見してみる.

```
$ ls *[Dd]yn*
```

4. 関連するコードの構造や書き方を比較してみる (*.C and *.H).

Private member functions: updateSubGridScaleFields

Smagorinsky.C

```
void Smagorinsky::updateSubGridScaleFields  
(const volTensorField& gradU)  
{ nuSgs_ = ck_*delta()*sqrt(k(gradU));  
  nuSgs_.correctBoundaryConditions(); }
```

dynLagrangian.C

```
void dynLagrangian::updateSubGridScaleFields  
(const tmp<volTensorField>& gradU)  
{ nuSgs_ = (flm_/fmm_)*sqr(delta())*mag(dev(symm(gradU)));  
  nuSgs_.correctBoundaryConditions(); }
```

dynOneEqEddy.C

```
void dynOneEqEddy::updateSubGridScaleFields  
( const volSymmTensorField& D,  
  const volScalarField& KK )  
{ nuSgs_ = ck(D, KK)*sqrt(k_)*delta();  
  nuSgs_.correctBoundaryConditions(); }
```

コードによる定式化の理解

- コヒーレント構造スモゴリンスキーモデル(CSM)にはどのような計算, 演算, 変数が必要か調べてみる. 関連するコードを見ながらモデルの定式化・実装を理解する.
- CSMモデルでは, 速度勾配テンソルの第二不変量 Q が用いられる:

$$Q = \frac{1}{2} (\bar{\Omega}_{ij} \bar{\Omega}_{ij} - \bar{S}_{ij} \bar{S}_{ij}) = -\frac{1}{2} \frac{\partial \bar{u}_j}{\partial x_i} \frac{\partial \bar{u}_i}{\partial x_j}$$

ここで,

$$S_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \quad \Omega_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} - \frac{\partial \bar{u}_j}{\partial x_i} \right)$$

非回転流でのコヒーレント構造 スマゴリンスキーモデル(NRCSM)

- 渦粘性近似を用いたスマゴリンスキーモデル (SM)

$$\tau_{ij}^a = -2C\Delta^2 |\bar{S}| \bar{S}_{ij}$$

$$(\tau_{ij}^a = -2\nu_t \bar{S}_{ij}, \nu_t = C\Delta^2 |\bar{S}|)$$

- モデル定数 C は以下の様に定義される.

$$C = C_1 |F_{CS}|^{3/2}$$

ここで,

$$C_1 = \frac{1}{20}, F_{CS} = \frac{Q}{E}$$

また,

$$E = \frac{1}{2} (\bar{\Omega}_{ij} \bar{\Omega}_{ij} + \bar{S}_{ij} \bar{S}_{ij}) = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} \right)^2$$

NRCSMモデルは回転流れには不適當である.

コヒーレント構造スマゴリンスキーモデル (CSM)

- 渦粘性近似を用いたスマゴリンスキーモデル (SM)

$$\tau_{ij}^a = -2C\Delta^2 |\bar{S}| \bar{S}_{ij}$$

$$(\tau_{ij}^a = -2\nu_t \bar{S}_{ij}, \nu_t = C\Delta^2 |\bar{S}|)$$

- モデル定数 C は以下の様に定義される.

$$C = C_2 |F_{CS}|^{3/2} F_{\Omega}$$

ここで,

$$C_2 = \frac{1}{22}, F_{CS} = \frac{Q}{E}, F_{\Omega} = 1 - F_{CS}$$

また,

$$E = \frac{1}{2} \left(\bar{\Omega}_{ij} \bar{\Omega}_{ij} + \bar{S}_{ij} \bar{S}_{ij} \right) = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} \right)^2$$

CSMモデルは回転流れに対しても適用できる.

新しいライブラリ作成の準備

1. WALEモデルのコードをコピーしてコンパイルする.

```
$ run
$ cd ../src/libraries
$ cp -r incompressibleWALE/WALE/ ./NRCSM
$ cp -r incompressibleWALE/Make ./NRCSM
$ cd NRCSM
$ rename WALE NRCSM *
$ rm -r NRCSM.dep
$ rm -rf Make/linux64Gcc47DPOpt
$ gedit Make/files
```

```
NRCSM.C
```

```
LIB = $(FOAM_USER_LIBBIN)/libNRCSM
```

```
$ sed -i 's/WALE/NRCSM/g' NRCSM.C
$ sed -i 's/WALE/NRCSM/g' NRCSM.H
```

新しいライブラリ作成の準備

```
$ wmake libso  
$ ls $FOAM_USER_LIBBIN
```

名前を変更してコンパイルしたライブラリ (libNRCSM.so)を見つけたら, 新しいNRCSMモデルライブラリの準備が完了.

2. Q 項・ E 項を計算するためのコードは, postProcessingユーティリティで理解できる.

```
$ util  
$ cd postProcessing/velocityField/Q  
$ gedit Q.C &
```

Q の計算方法は, (1)速度勾配テンソルを用いる, (2)SS項と $\Omega\Omega$ 項を用いる, の2通りがある.

モデル定数 c_1 の導入

3. すべての'cw'を'c1'に置換し(geditまたはsed), 値を0.05に変更する.

```
$ run
$ cd ../src/libraries/NRCSM/
$ gedit NRCSM.C NRCSM.H
```

NRCSM.C

```
    c1_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "c1",
            coeffDict_,
            0.05
        )
    )
)
```

```
$ wmake libso
```

QとEの計算

4. NRCSM.Cで, Q と E の計算を適当な箇所に挿入する (Q.Cから該当する部分をコピー&ペースト). 保存&コンパイルする.

```
$ gedit NRCSM.C
```

```
NRCSM.C
```

```
volScalarField Q  
(  
    0.5*(sqr(tr(gradU)) - tr(((gradU)&(gradU))))  
);  
volScalarField E  
(  
    0.5*(gradU && gradU)  
);
```

```
$ wmake libso
```

F_{CS} と C の計算

5. NRCSM.C で, F_{CS} と C (渦粘性モデル定数) の計算部分を追加する. 保存&コンパイルする.

```
$ gedit NRCSM.C
```

NRCSM.C

```
volScalarField Fcs  
(  
    Q/  
    max(E,dimensionedScalar("SMALL",E.dimensions(),SMALL))  
);  
volScalarField ccsm_  
(  
    c1_*pow(mag(Fcs),1.5)  
);
```

```
$ wmake libso
```

v_{SGS} の計算

6. NRCSM.C で, nuSGS_ の計算を修正する. ダイナミックモデルでの updateSubGridScaleFields 関数も参考にしてみる.

```
$ gedit NRCSM.C
```

```
NRCSM.C
```

```
nuSgs_ = ccsm_*sqr(delta())*mag(dev(symm(gradU)));
```

保存&コンパイルする.

```
$ wmake libso
```

7. 最後に, 必要の無い計算部分をコメントアウト or 削除する (WALEモデル計算部分). 保存&コンパイルする.

```
$ wmake libso
```

k_{SGS} の計算

8. k_{SGS} の計算は正しくないが, k_{SGS} の値はNRCSMモデルを用いたLESでは必要としない. もし, 適切な k_{SGS} の値が必要ならば, オリジナルの論文(Kobayashi,PoF, 2005)を参照.

NRCSM.H

```
//- Return SGS kinetic energy
//  calculated from the given velocity gradient
tmp<volScalarField> k(const tmp<volTensorField>& gradU) const
{
    return (2.0*c1_/ce_)*sqr(delta())*magSqr(dev(symm(gradU)));
}
```

チャンネル流れを用いた検証

9. $Re_\tau = 395$ のチャンネル流れチュートリアルを自分のrunディレクトリにコピーする.

```
$ run
$ cp -r $FOAM_TUTORIALS/incompressible/pimpleFoam/
channel395/ ./ReTau395NRCSM
$ cd ReTau395NRCSM
```

10. constant/LESProperties と system/controlDict を編集する.

```
$ gedit constant/LESProperties
```

```
LESModel          NRCSM;
printCoeffs       on;
delta              cubeRootVol;
...
```

チャンネル流れを用いた検証

```
$ gedit system/controlDict
```

```
...  
libs ("libNRCSM.so");
```

この1行は新しいNRCSM ライブラリをソルバー実行時に呼び出すために必要.

11. その他の解析条件とパラメータを確認し, ソルバーを実行する.

```
$ ./Allrun
```

5. 計算が正常に終了したら, ログを確認し, ParaViewで流れ場を可視化してみる. また, postChannelで生成されたプロファイルをプロットしてみる.

自習課題

1. openfoam-devで提供されているWALEモデルをコンパイル・テストしてみる. Makeディレクトリを自分で用意する必要がある.
2. CSMモデルの実装. F_Ω 項と係数 C_2 を追加する.
3. Q 項・ E 項を SS 項・ $\Omega\Omega$ 項で計算してみる. 演習2の解と比較してみる.
4. カスタマイズしたモデルを, その他の流れ(円管, バックステップ, 円柱, 回転流, など)で検証してみる.

References

- OpenFOAM User Guide
- OpenFOAM Programmer's Guide
- 梶島, 乱流の数値シミュレーション 改訂版, 養賢堂 (2014).
- P. R. Spalart et al., “Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach”, 1st ASOSR CONFERENCE on DNS/LES (1997).
- P. R. Spalart, “Young-Person's Guide to Detached-Eddy Simulation Grids”, NASA CR-2001-211032 (2001).
- F. Nicoud and F. Ducros, “Subgrid-scale modelling based on the square of velocity gradient tensor”, Flow, Turbulence and Combustion, 62, pp.183-200 (1999).

References

- 小林, “乱流構造に基づくサブグリッドスケールモデルの開発”, *ながれ*, 29, pp.157-160 (2010).
- H. Kobayashi, “The subgrid-scale models based on coherent structures for rotating homogeneous turbulence and turbulent channel flow”, *Phys. Fluids*, 17, 045104 (2005).
- H. Kobayashi, F. Ham and X. Wu, “Application of a local SGS model based on coherent structures to complex geometries”, *Int. J. Heat Fluid Flow*, 29, pp.640-653 (2008).