OpenModelica超初級チュートリアル

7. プラントモデル

注意事項

- 本チュートリアルは以下の内容が理解できていることを前提としております。
  「OpenModelica超初級チュートリアル1.解析モデルの作成と実行」
  「OpenModelica超初級チュートリアル2.コーディング」
  「OpenModelica超初級チュートリアル3.モデルのカスタマイズ1」
  「OpenModelica超初級チュートリアル4.モデルのカスタマイズ2」
  「OpenModelica超初級チュートリアル5.モデルのカスタマイズ3」
- OpenModelica1.14.1 (64bit – windows版)を利用して本チュートリアルは作成されています。

プラントモデル

プラントモデルの概要を理解し、既存ライブラリを確認してみましょう

プラントモデルが理解できるようになると？

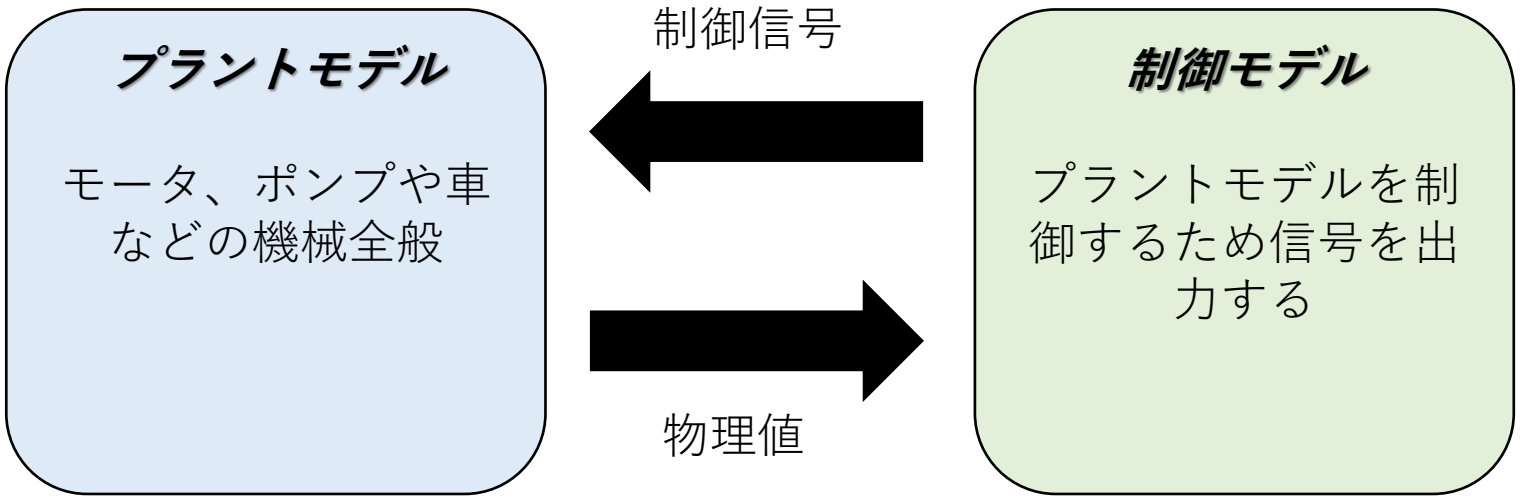
- ✓様々な物理現象を計算できるようになる
- ✓既存の物理ライブラリが何をしているか分かる
- ✓オリジナルの物理ライブラリを作れるようになる

プラントモデルと制御モデル

1DCAEの主要なモデルは大雑把に以下の２つに大別できます。

***プラントモデル*・・・機械の動きや流体の流れなど物理法則に従う挙動をシミュレートするためのモデル**

***制御モデル*・・・ プラントモデルをコントロールするための信号や演算をシミュレートするモデル**



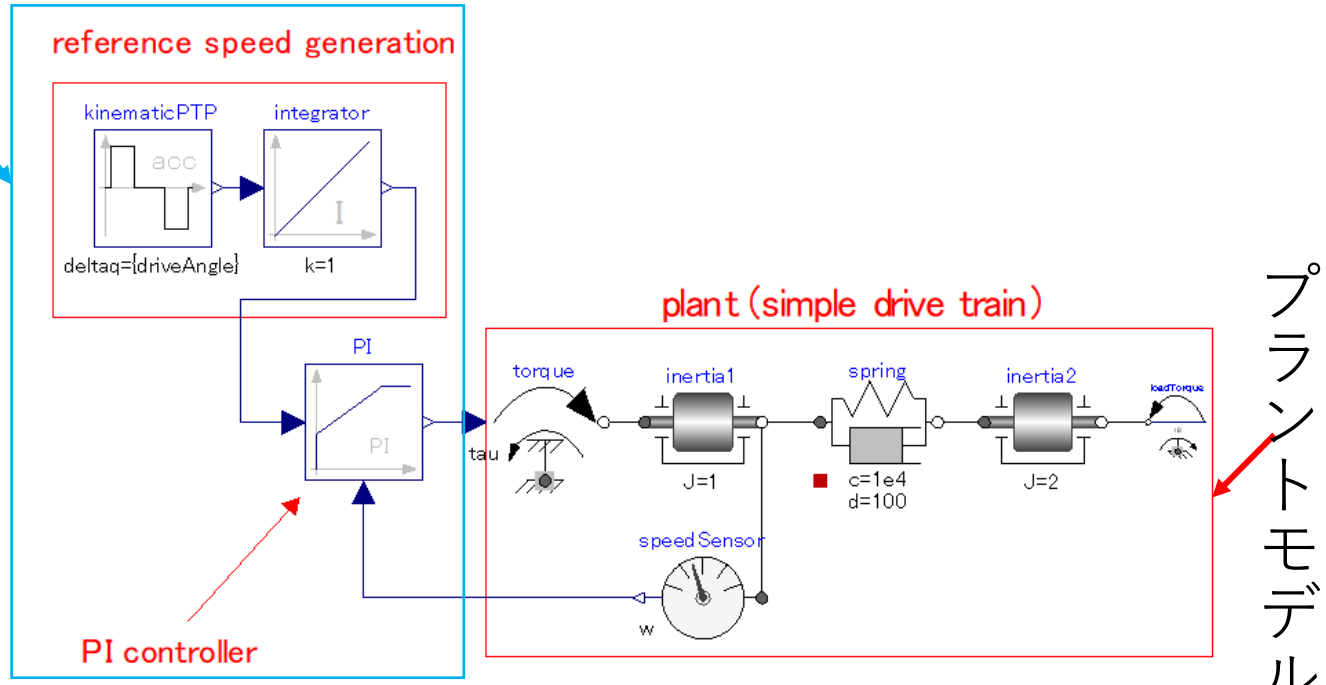
制御モデルがなくプラントモデルだけの場合もあります

プラントモデルと制御モデルの例

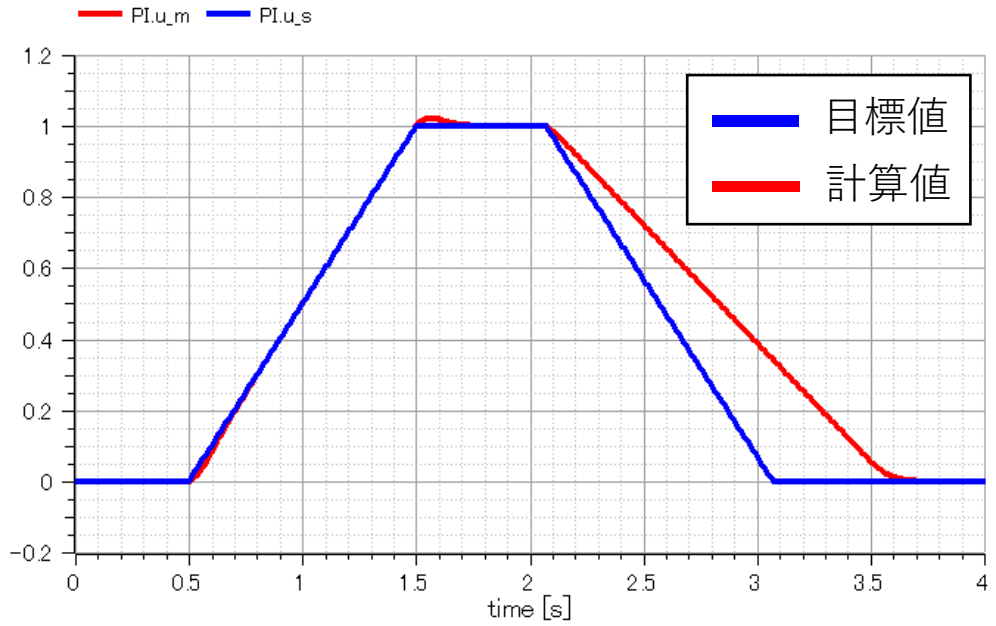
以下はMSL*のプラントモデルと制御モデルのサンプルです。プラントモデルは回転するドライブトレイン(シャフトと負荷)であり、制御モデルで目標の回転速度となるようにプラントモデルにトルクを与えています。

*Modelica Standard Library

制御モデル



プラントモデル



Modelica.Blocks.Examples.PID_Controller

目標の回転速度 vs プラントモデルの回転速度

プラントモデルと物理ライブラリ

プラントモデルは物理ライブラリのコンポーネントを使用して作成します。
代表的な物理現象とそれに対応するライブラリの例を以下に示します。

物理現象	物理ライブラリ
熱	Thermal.HeatTransfer
流体	Thermal.FluidHeatFlow
電気(アナログ)	Electrical.Analog
磁気	Magnetic.FluxTubes
並進運動	Mechanics.Translational
回転運動	Mechanics.Rotational

既存ライブラリを上手く活用することで効率的にモデリングできます

Modelicaを使用するメリット

効率よく直感的なプラントモデルを作成するためにModelicaは非常に便利です。
具体的には以下のようなメリットがあります

メリット①　モデリング言語

- モデルをグラフィカルに操作できる

メリット②　非因果モデル

- モデルが計算の順序に依存しない

メリット③　物理現象を表す様々な変数、オペレータ

- 物理現象を表すための変数、オペレータが数多く存在する

メリット④　豊富なライブラリ

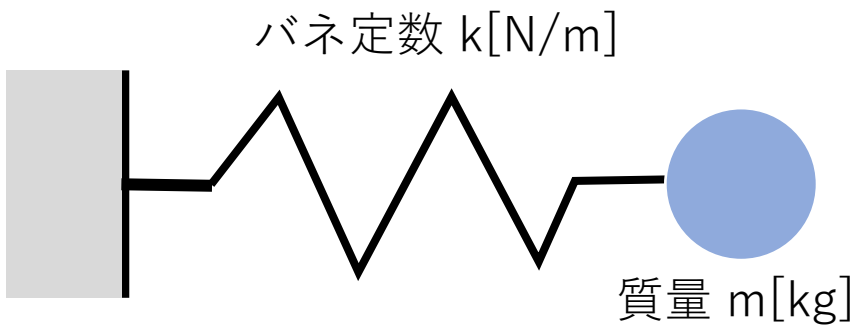
- Modelica Standard Libraryや数々の商用ライブラリがある

Modelicaを使用するメリット

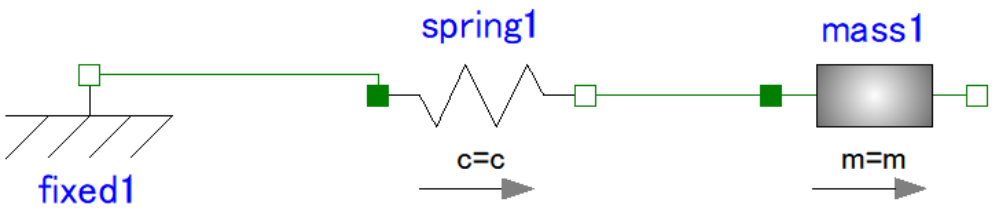
メリット①　モデリング言語

- モデルをグラフィカルに操作できる

これまで学習してきたとおり、GUIからドラッグ＆ドロップで計算プログラムを作成することが出来ます



解析対象の系



Modelicaモデル

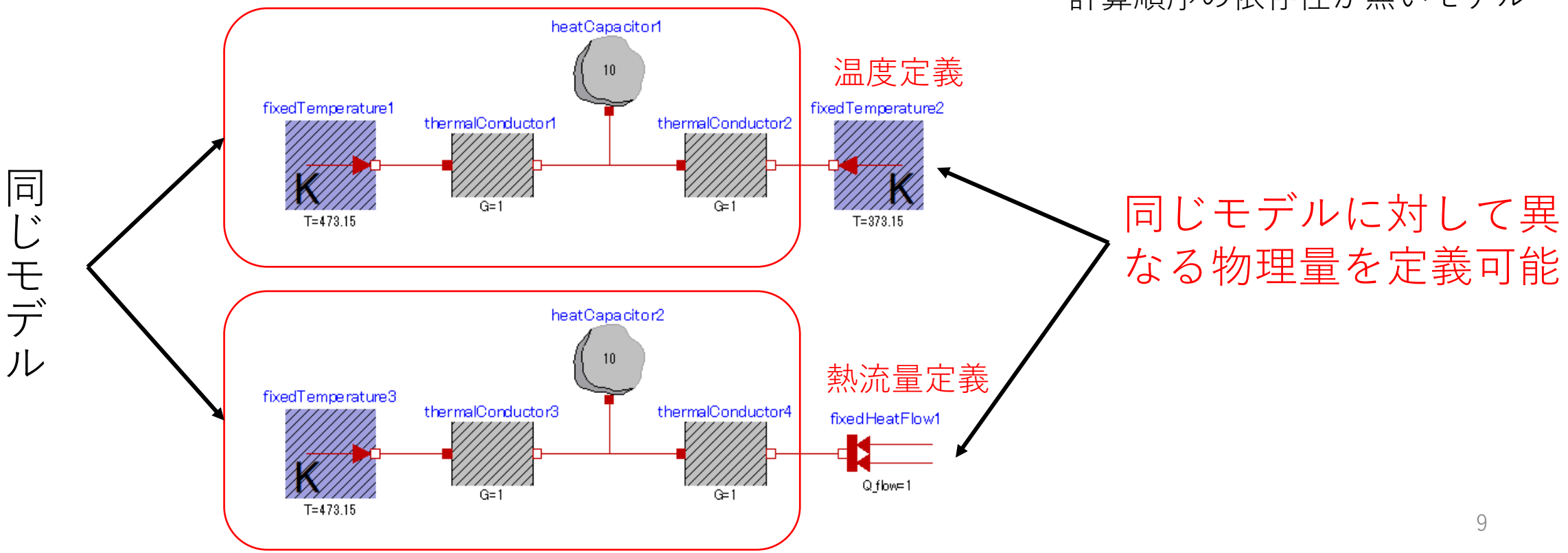
Modelicaを使用するメリット

メリット②　非因果モデル

- モデルが計算の順序に依存しない

非因果モデル*で計算可能なため、コンポーネントや境界条件を変更しても
同じモデルで計算が可能です

＊ 計算順序の依存性が無いモデル

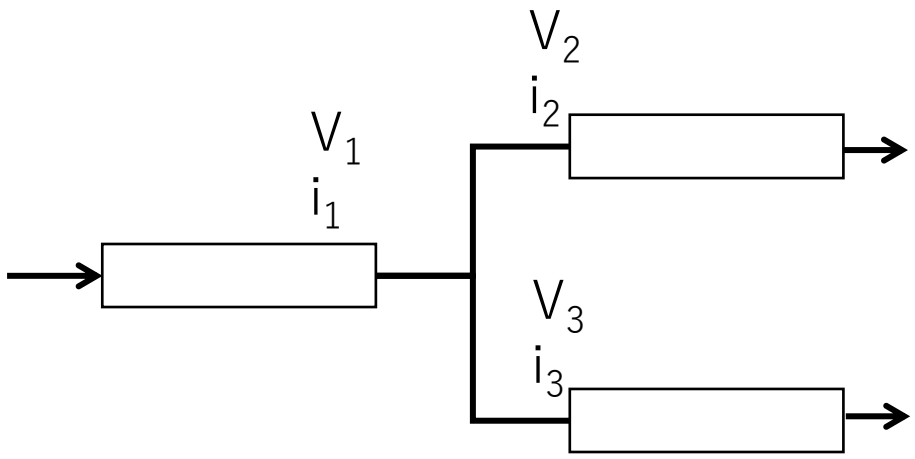


Modelicaを使用するメリット

メリット③　物理現象を表す様々な変数、オペレータ

- 物理現象を表すための変数、オペレータが数多く存在する

電圧、温度などはacross変数、電流、熱流量などはflow変数として宣言することでモデルの接続関係から自動的に方程式を組み立てます。
また比エンタルピーや質量分率を表すstream変数や物理計算を解く際に有用なオペレータ(homotopy, spatialDistribution etc.)などがあります。



並列接続された電気抵抗

across変数

$$V_1=V_2=V_3$$

flow変数

$$i_1=i_2+i_3$$

Modelicaを使用するメリット

メリット④　豊富なライブラリ

- Modelica Standard Libraryや数々の商用ライブラリがある

非常に多くのオープンソースライブラリや商用ライブラリが活発に開発、公開/販売されています

OpenModelicaにインポートされている物理ライブラリの一例

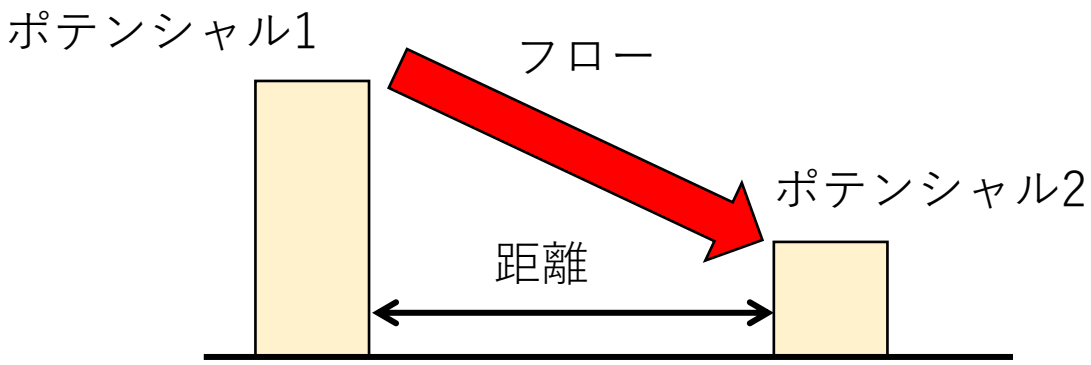
物理現象	解析対象
流体	車両
熱	建築
構造	風力発電
振動	光発電
騒音	電力システム
電磁気	生理現象
化学反応	核反応炉
生化学	サーボ
etc.	燃料電池
	etc.

プラントモデルを理解するために

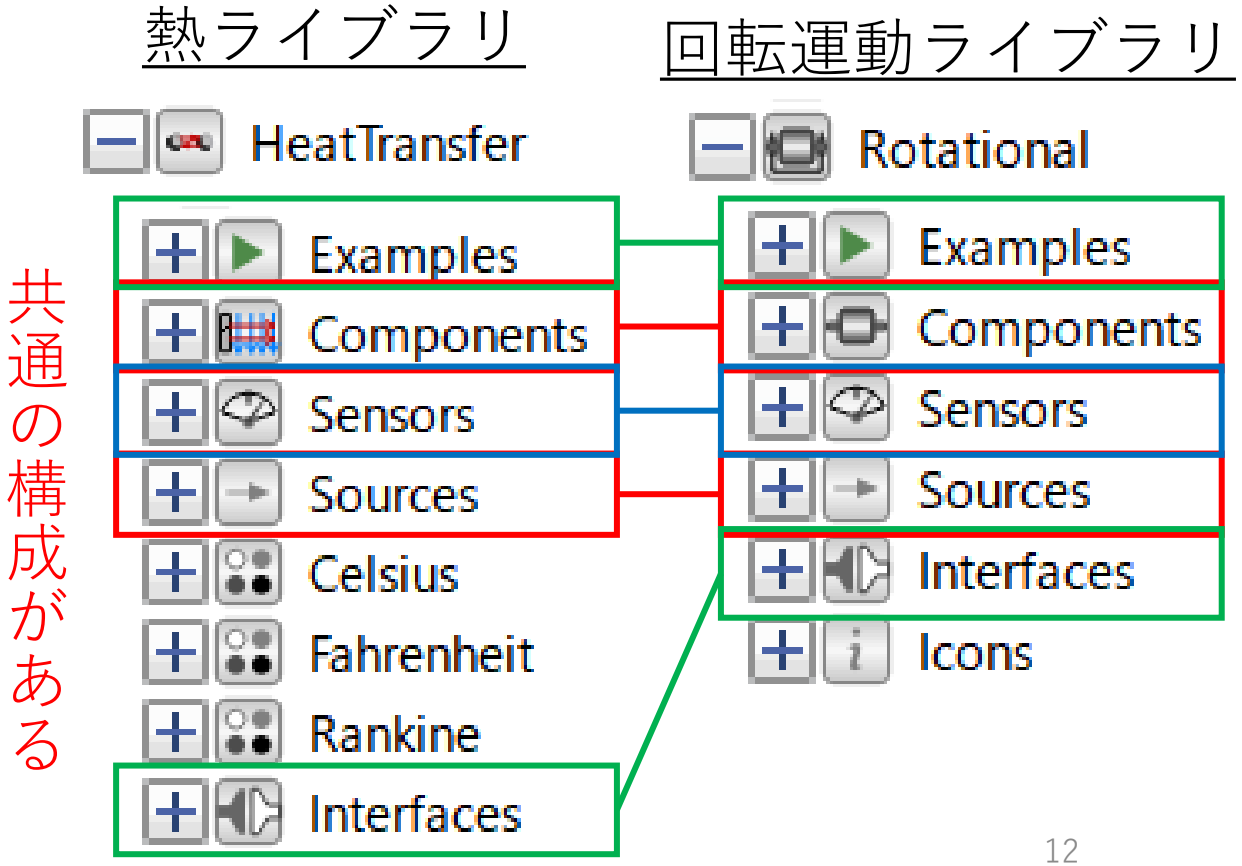
プラントモデルを上手く活用するためには、以下の一般的な考え方を理解することが重要です。

物理現象の記述

様々な物理現象を統一的な考え方で記述するため、ポテンシャルとフローという概念がある



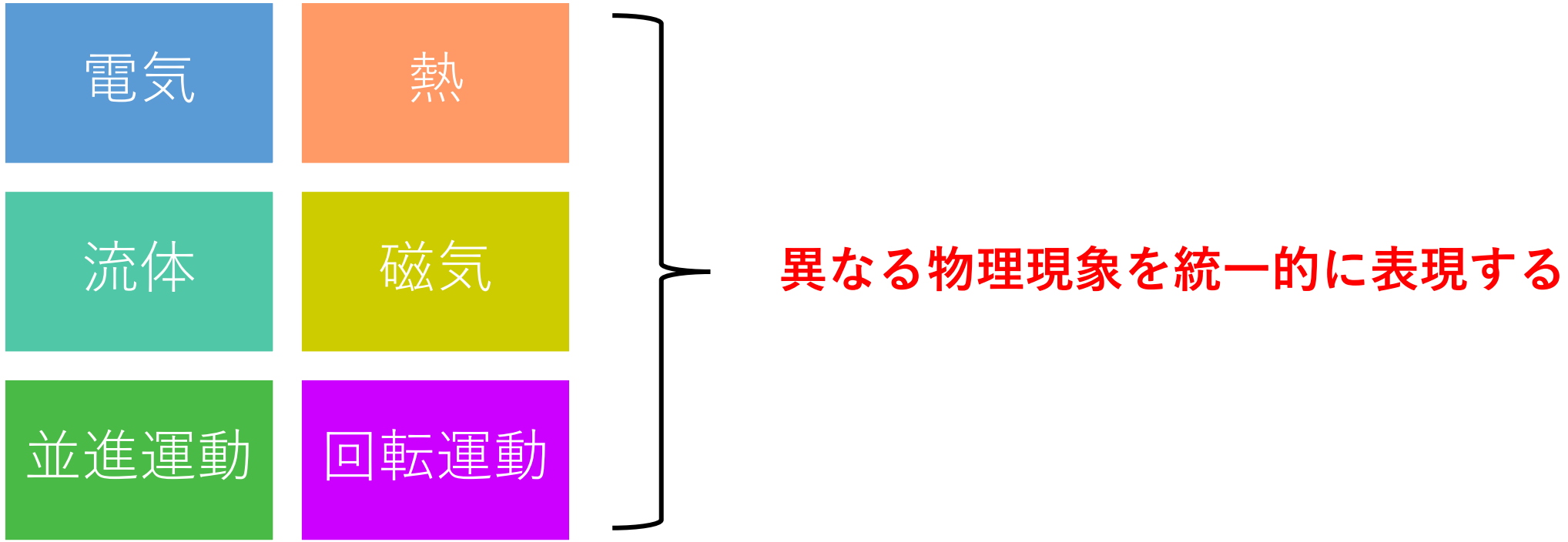
物理ライブラリの共通構成



物理現象の記述

物理現象の記述

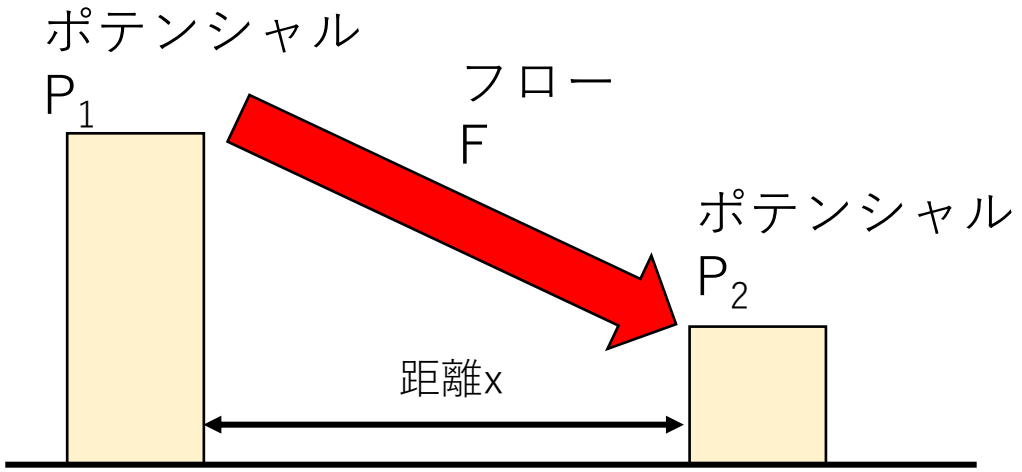
Modelicaではポテンシャルとフローという概念を使用して様々な物理現象を統一的に表現しています。
この概念によって異なる物理現象でも同じような考え方で表すことが可能です。ただし空間的な分布や精度の良い計算を実施したい場合にはこの概念だけでは十分ではない可能性があるので注意してください。



ポテンシャルとフロー

多くの物理現象はポテンシャル*1と、そのポテンシャルの勾配に応じて発生するフロー*1(流動量)によって統一的に表すことが出来ます。

ここでポテンシャルとは、何かを駆動させることができる潜在的なスカラー量程度にお考え下さい。



ポテンシャルとフローの関係式

*(フロー)=(コンダクタンス*2)×(ポテンシャルの差や勾配)*

$$F = -\lambda \Delta P \qquad \Delta P = \begin{cases} P_2 - P_1 \\ or \\ \dfrac{P_2 - P_1}{x} \end{cases}$$

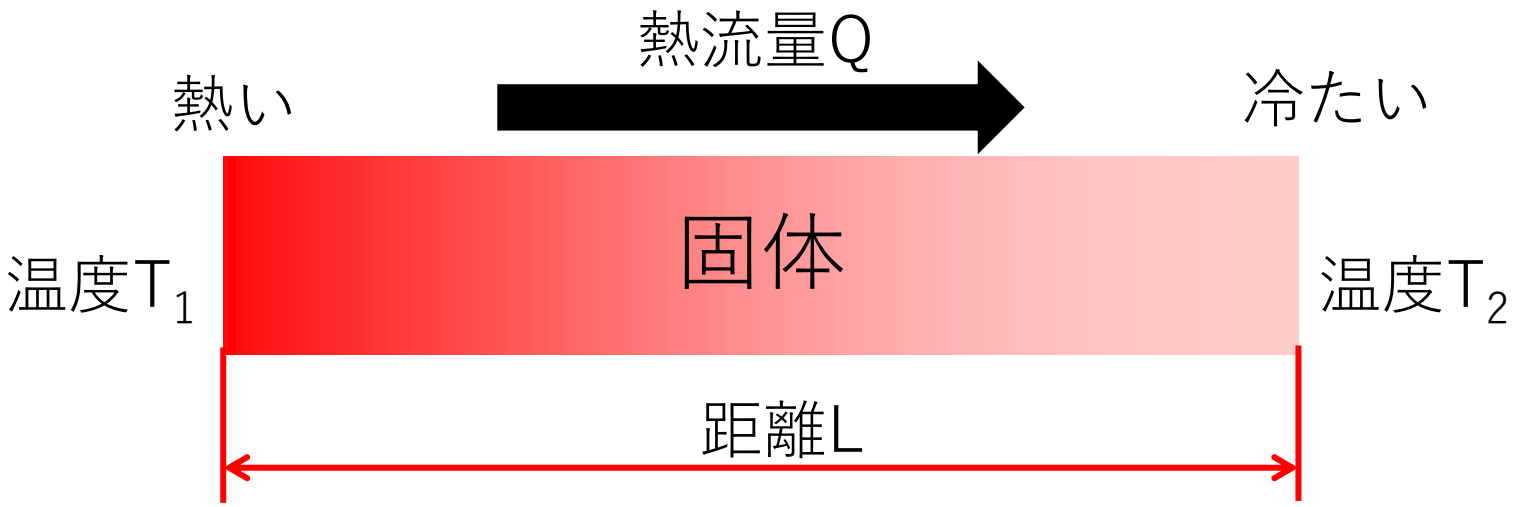
フローはポテンシャルの大きい(高い)ところから小さい(低い)ところへ流れます。

*1 分野によって呼び方は異なります

*2 フローの流れやすさを表す度合い

ポテンシャルとフロー - 固体の熱伝導の場合

温度をポテンシャル、熱流量をフローと考えると
熱の移動現象は以下のようになります。



温度と熱流量の関係式
(フーリエの法則)

$$Q = -kA\frac{T_2 - T_1}{x}$$

フロー

コンダクタンス

ポテンシャルの差分

k：熱伝導率
A：断面積

ポテンシャルとフロー – 各物理ドメイン

Modelicaでは、ポテンシャルをacross変数、フローをflow変数として定義します。
また各物理現象におけるポテンシャルとフローの関係は以下のようになっています。

MSLの代表的なポテンシャルとフロー

物理現象	MSL	ポテンシャル（across変数）	フロー（flow変数）	代表的な関係式	関係する法則や式
熱	Thermal.HeatTransfer	温度T	熱流量Q	$Q=\Lambda \times \Delta T$	フーリエの法則
流体	Thermal.FluidHeatFlow	圧力p	質量流量m	$m^2=k^* \Delta p$	ダルシー・ワイスバッハの式
電気	Electrical.Analog	電圧V_e	電流i	$i = G \times \Delta v_e$	オームの法則
磁気	Magnetic.FluxTubes	磁位V_m	磁束ϕ	$\phi = P \times \Delta V_m$	ホプキンスの法則
並進運動	Mechanics.Translational	位置s	力F	$F = K \times \Delta s$	フックの法則
回転運動	Mechanics.Rotational	回転角度ϕ	トルクτ	$\tau = C_R \times \Delta \Phi$	?

変数	記号
熱コンダクタンス	Λ
温度勾配	ΔT
損失係数	k
圧力勾配	Δp
電気コンダクタンス	G
電位差	ΔV
磁気コンダクタンス	P
磁位差	ΔV_m
ばね剛性	K
変位	Δs
回転剛性	J
ねじれ角	$\Delta \Phi$

across変数とflow変数の文法

across変数、flow変数はconnectorクラスに宣言されます。
across変数はスカラー（大きさのみで向きがない値)であり、
flow変数はベクトル(向きがある値)です。

例. across変数、flow変数の宣言方法(熱の場合)

```
connector HeatPort
    Real T;
    flow Real Q_flow;
end HeatPort;
```

温度　ポテンシャルに対応するacross変数

熱流量　フローに対応するflow変数

flow変数は接頭辞に「flow」と宣言します

across変数とflow変数の実装例

Modelicaでは、across変数、flow変数はconnectorクラス内で宣言します。
以下に典型的なconnectorクラスでの実装例を示します。

熱

```
connector HeatPort
  Modelica.SIunits.Temperature T;
  flow Modelica.SIunits.HeatFlowRate Q_flow;
end HeatPort;
```

流体(非圧縮性)

```
connector Incompressible
  Modelica.SIunits.Pressure p;
  flow Modelica.SIunits.VolumeFlowRate q;
end Incompressible;
```

電気(アナログ)

```
connector Pin
  SI.ElectricPotential v;
  flow SI.Current i;
end Pin;
```

磁気

```
connector MagneticPort
  SI.MagneticPotential V_m;
  flow SI.MagneticFlux Phi;
end MagneticPort;
```

並進運動

```
connector Flange
  SI.Position s;
  flow SI.Force f;
end Flange;
```

回転運動

```
connector Flange
  SI.Angle phi;
  flow SI.Torque tau;
end Flange;
```

コラム - ポテンシャルとフローの呼び名

ポテンシャルやフローはツールや学問領域によって呼び方が変わり、厳密には異なるかもしれませんが
おおむね以下のように呼ばれています。

ツールや学問領域	ポテンシャル	フロー
ボンドグラフ	エフォート	フロー
Matlab/Simulink	across変数 (横断変数)	through変数 （通過変数）
Modelica	across変数や ポテンシャル	flow変数や through
移動現象論	ポテンシャル	フラックス (単位時間・面積当たりの移動量)
VHDL-AMS	across変数	through変数

ポテンシャルとフローだけでは流体の輸送現象(比エンタルピーや質量分率)を表すことが煩雑になるため
Modelicaではstream変数という変数を導入しています。
stream変数については別資料「OpenModelica超初級チュートリアル7.5 番外編 stream変数」にて解説します。

ポテンシャルとフロー

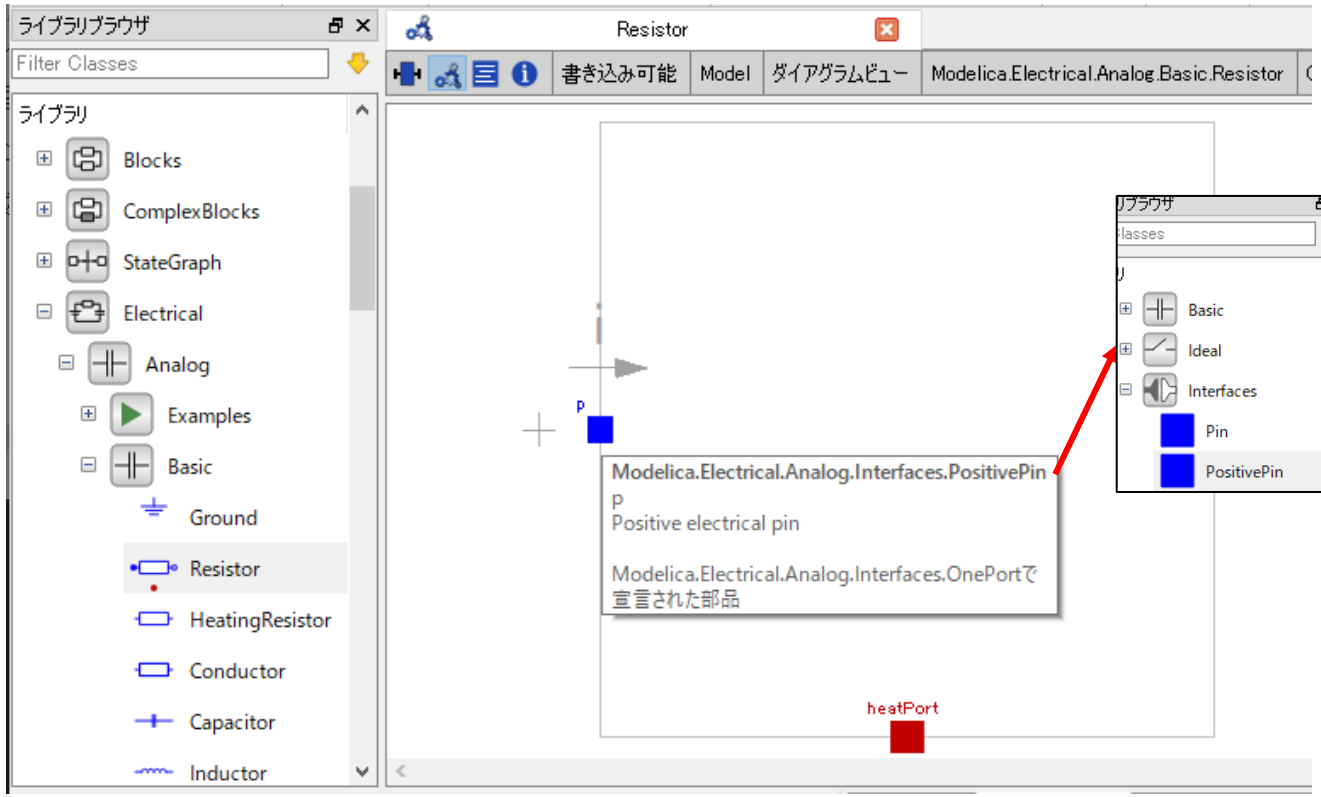
Exercise1

OpenModelicaを使用して、Modelica.Electrical.Analogパッケージの物理モデルに使用されているポートを確認してみてください。

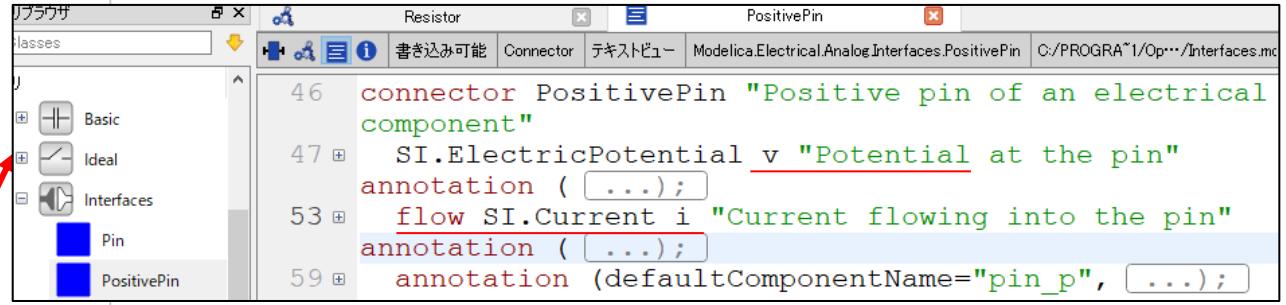
ポテンシャルとフロー

解答1

Modelica.Electrical.Analog内の適当なモデルを開きます。
ポートにカーソルを合わせるとインスタンス元のモデルのパスが示されます。

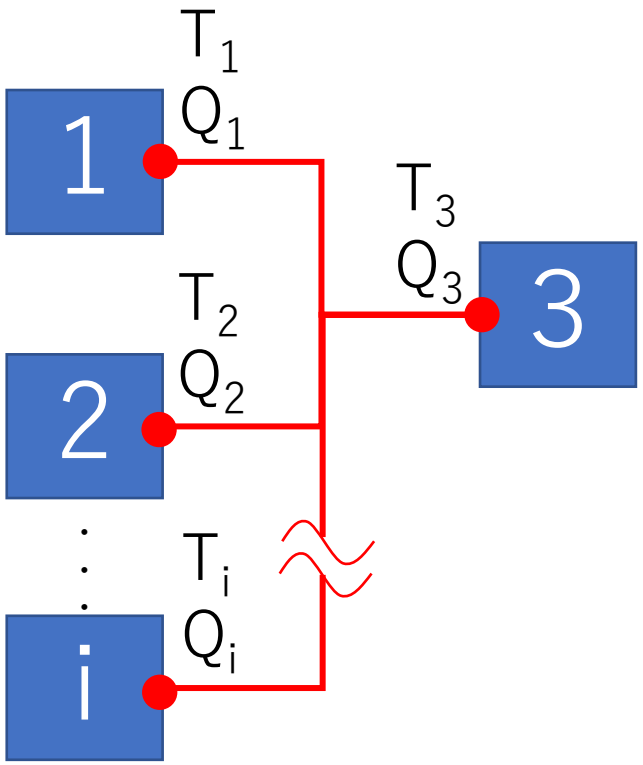


ポートモデルを確認するとポテンシャルとしてv
フローとしてiが定義されています



across変数とflow変数のメリット

across変数、flow変数を宣言することでモデル同士を接続した際に各変数が物理的に自然な挙動となるように自動的に計算式が組み立てられます。
across変数は各ポートの値が等しくなるように、flow変数は各ポートの総量が０（保存則）となるように計算式が生成されます。
これによりモデルをいくら繋いでも削除しても自動的に計算式が組み立てられるためシステムの変更が容易となります。



across変数の接続の式(キルヒホッフの電圧則)

$$T_1 = T_2 = T_3 = \cdots = T_i$$

flow変数の接続の式(キルヒホッフの電流則)

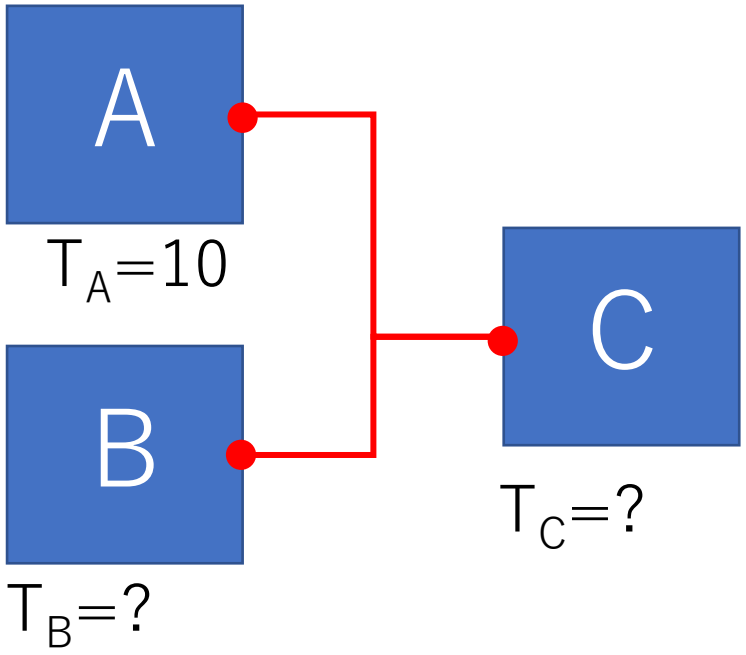
$$Q_1 + Q_2 + Q_3 + \cdots + Q_i = 0$$

$$\rightarrow \sum_{n=1}^{i} Q_n = 0$$

across変数の具体例

まずは基本となるacross変数の計算式を温度を例に取って解説します。

以下のように3モデルの接続図において、モデルAのポート温度が10°Cの時はモデルB,Cのポート温度はいくらしょうか？



across変数の接続の式

$$T_A = T_B = T_C$$

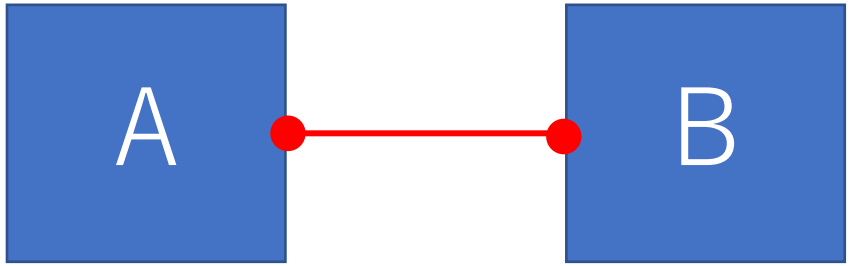
モデルB,Cのポート温度は10°Cになります

across変数は値を変更せずに受け渡します

flow変数の具体例 – 2モデルの接続

続いてflow変数の計算式を熱流量Qを例に解説します

以下のモデルにおいて、Aモデルのポート熱流量をQ_A(=10W)とします。
Bモデルのポート熱流量はいくらになるでしょうか？



Q_A=10　　Q_B=?

flow変数の接続の式

$$Q_A + Q_B = 0$$

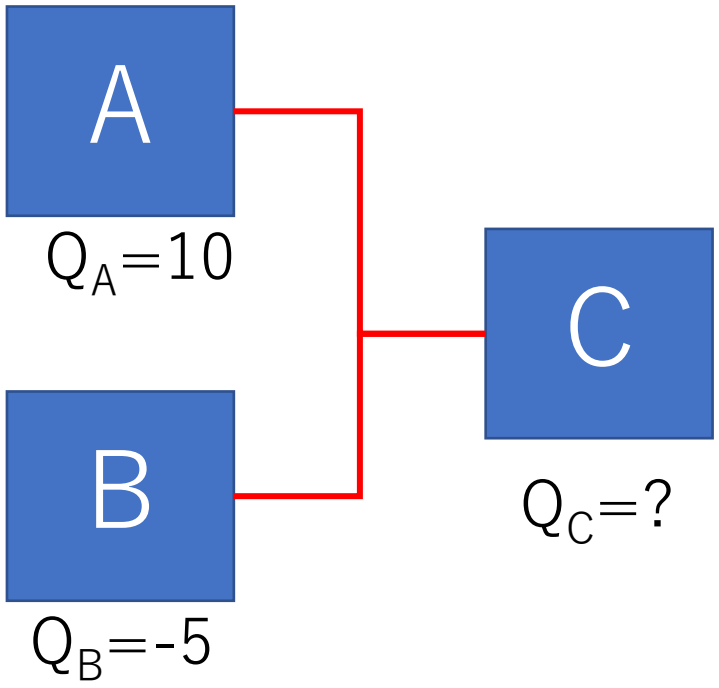
モデルBの熱流量Q_B=Q_Aより-10Wです

Modelica言語ではflow変数についてモデルに流入する場合を正、モデルから流出する場合を負と考えるのが一般的です。上記の場合、Aに熱流量が流入していることになります。

flow変数は接続されたポート間の総量がゼロとなるように値を受け渡します

flow変数の具体例 – 3モデルの接続

以下のように３つのモデルが接続され、モデルA,Bのポート熱流量が10W,-5Wの時はモデルCのポート熱流量はいくらしょうか？
また、どのモデルからどのモデルへ熱流量が流れているのでしょうか？



flow変数の接続の式

$$Q_A + Q_B + Q_C = 0$$

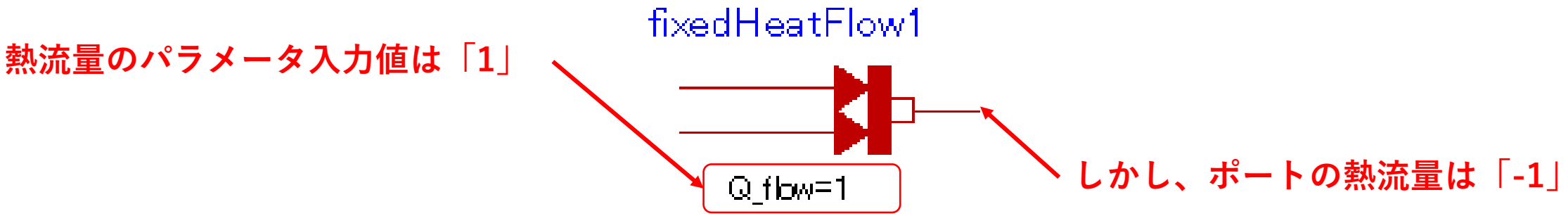
モデルCの熱流量Q_Cは-5Wです
モデルB,CからモデルAに熱量が流れ込んでいます。

どのモデルからどのモデルにflow変数が流れているか
イメージしながら計算しましょう

コラム - flow変数の正負について

ここでモデルの計算結果を確認する際に少し違和感がある実装について解説します。

熱流量を定義するFixedHeatFlowモデルを使用する際、ユーザーは熱流量パラメータに正の値を入力します。
しかし計算を実行しポートの熱流量を確認すると負の値となっています。



flow変数は、モデルに流入する場合が正、流出する場合が負とするのが慣例です。
その慣例にならうと熱流量を1Wを出力する場合、パラメータに「-1」とユーザーは入力しないといけません。
しかし、直感的ではないためほとんどのライブラリではユーザーの入力は正として
モデル内部はマイナスをかけて負としています。

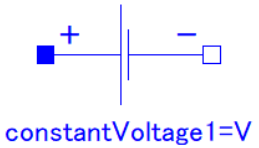
FixedHeatFlowの計算式

```
port.Q_flow = -Q_flow*(1 + alpha*(port.T - T_ref));
```

物理的なモデル

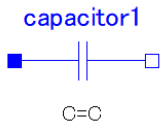
Exercise2

OpenModelicaを使用して、以下のモデルのソースコードを確認しどのように方程式が定義されているか確認してみてください



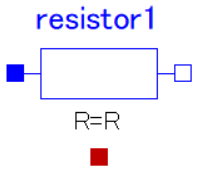
電源

Modelica.Electrical.Analog.Sources.ConstantVoltage



コンデンサ

Modelica.Electrical.Analog.Basic.Capacitor



抵抗

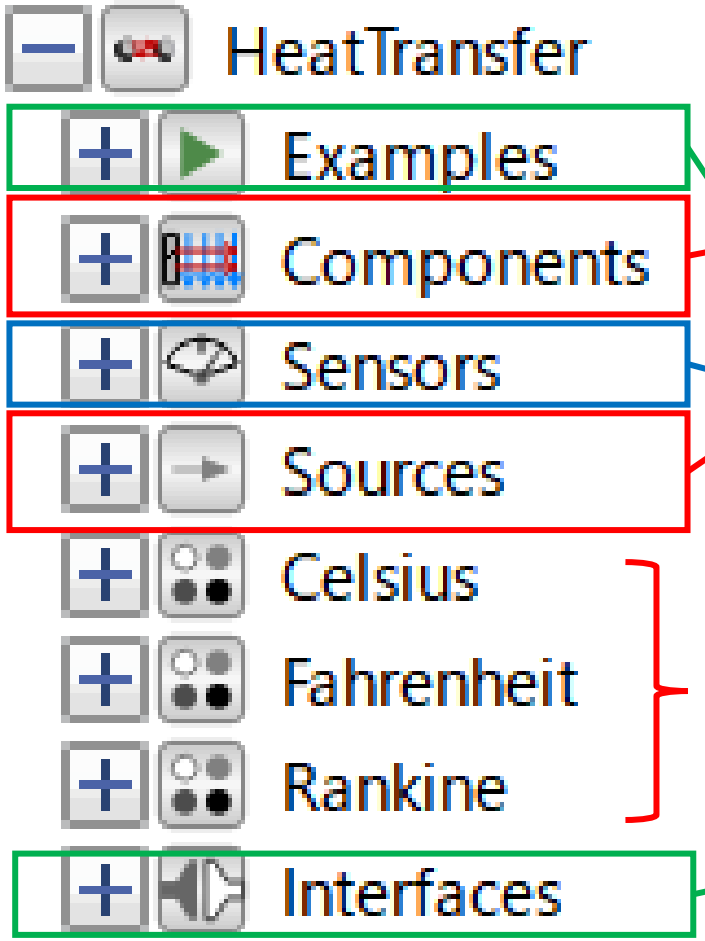
Modelica.Electrical.Analog.Basic.Resistor

既存の物理ライブラリの共通構成

既存の物理ライブラリの共通構成

MSLにおいて、物理ライブラリは基本的に以下の構成となります。

熱ライブラリの例



基本構成

物理モデル(次スライド以降で詳述)

Sources　・・・境界条件の定義

Components・・・抵抗・キャパシタ etc.

特定の物理値を信号として出力するセンサモデル

Sensors　・・・物理値の出力

ライブラリの整理や使用に役立つモデル

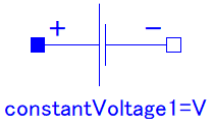
Interfaces　・・・モデル共通のコード

Examples　・・・モデルのサンプル

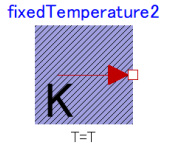
典型的な物理モデル

様々な物理ドメインに共通する挙動を以下のようにグループ化して把握するとどのライブラリも似たような考え方で使いこなすことが出来ます。

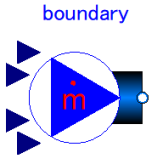
ソース・・・ポテンシャルあるいは
フローを境界条件として与える



電源



温度定義

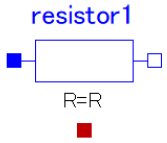


質量流量定義

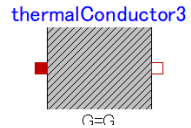
抵抗・・・物理量の通り易さを表す

$$f = \alpha_1 \Delta P$$

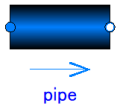
f：フロー　　P：ポテンシャル　　α_n：係数



抵抗



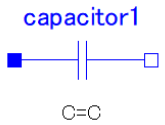
熱抵抗



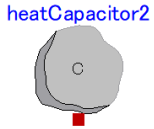
パイプ

キャパシタ・・・物理量をため込む

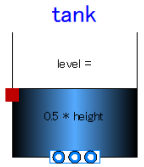
$$f = \alpha_2 \frac{dP}{dt}$$



コンデンサ



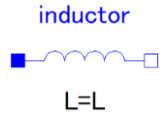
熱容量



タンク

インダクタンス・・・物理量の遅れ(応答性)を表す

$$P = \alpha_3 \frac{df}{dt}$$



コイル

該当無し

流体慣性

典型的な物理モデルの一覧

以下のようにグループ化すると様々な物理現象に対して共通の概念が適用できることが分かります。

モデル	電気	熱	流体	並進運動*1	回転運動*1
ソース　ポテンシャル	電圧源	温度定義	圧力定義	位置定義	回転角度定義
ソース　フロー	電流源	熱流量定義	流量定義	力定義	トルク定義
抵抗 R	電気抵抗	熱抵抗	配管	粘性摩擦	粘性摩擦
キャパシタ C	キャパシタ	熱容量	タンク	バネ	ねじりばね
イナータンス L	インダクター	(無し)	流体慣性	マス	イナーシャ
ポテンシャルとフローの変換	変圧器	(無し)	ノズル	てこ	ギヤボックス

*1 速度/回転速度をポテンシャル、力/トルクをフローとする独自の考え方を載せました。
(次スライドの補足資料を参照)

典型的な物理モデルの一覧 – 補足

どのような物理量をポテンシャルとフローにするかで抵抗やキャパシタは異なります。
MSLやボンドグラフでは並進/回転運動のポテンシャルとフローは異なるため抵抗やキャパシタに相当するモデルが異なることがあります。

前スライドでは速度をポテンシャル、力をフローとする独自の考え方を載せました。
ポテンシャルとフローの取り方は人それぞれなため、ここでは共通の考え方で異なる物理量を表すことができることを確認頂ければと思います。

名称	記号/数式	MSL.Translational ライブラリ	前スライドの考え方
ポテンシャル	P	位置	速度
フロー	f	力	力
抵抗	$f = \alpha_1 \Delta P$	バネ	粘性摩擦
キャパシタ	$f = \alpha_2 \frac{dP}{dt}$	粘性摩擦	バネ
インダクタンス	$P = \alpha_3 \frac{df}{dt}$	？	マス

MSL.Translationライブラリと前スライドの考え方の対比表

典型的な物理モデル

Exercise3

OpenModelicaを使用して、自分の興味ある物理ドメインに対して以下の典型的なコンポーネントを見つけてみてください。

・抵抗
・インダクタンス
・キャパシタ
・ソース　ポテンシャル

(解答資料無し)

典型的な物理モデル

Exercise4

ポテンシャルpとフローfを持つコネクタークラスを作成し、
抵抗、キャパシタ、インダクタンスクラスを作成してみてください。
(解答資料無し　ご要望があれば作成します)

まとめ

- ✓ 1DCAEの主要なモデルは物理現象を表すプラントモデルと、プラントモデルを制御するコントロールモデルに大別できます
- ✓ プラントモデルを作成するには、物理現象の記述方法と物理ライブラリの共通構成を理解すると便利です
- ✓ Modelicaでは物理現象をacross変数、flow変数の関係式で表します
- ✓ 物理ライブラリには境界条件、抵抗、キャパシタ、イナータンスを定義するモデルがある場合が多いです。

参考資料

Michael M. Tiller, Modelica by Example, https://mbe.modelica.university/components/connectors/