UK Atomic Energy Authority

UKAEA – Jan 30-31 2024

ReMKiT1D Workshop January 2024
Modelbound data in ReMKiT1D

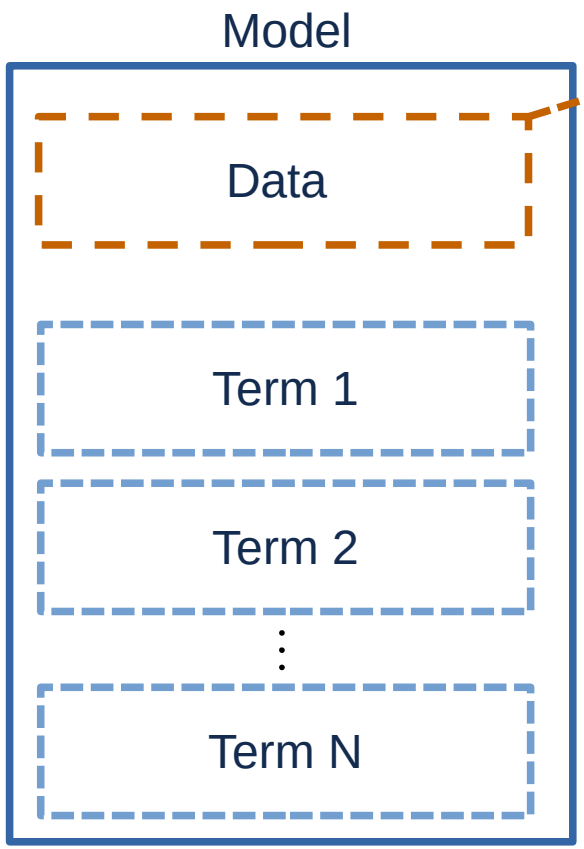
Imperial College London

CCFE
CULHAM CENTRE FOR FUSION ENERGY

Engineering and Physical Sciences Research Council

This work was partly funded by the RCUK Energy Programme
[Grant number EP/W006839/1]

Modelbound data – motivation



Only accessible to terms within

Why is this sometimes desirable?

Avoid unnecessary data copies

For example, integrators tend to make copies of the global variables in order to do intermediate calculations without changing the global data

Enable encapsulation at a high level

Use case: We want to write a model for someone to use. We don't want the user to have to create all the needed derived variables, or we want to make sure some variables are calculated in a particular

Modelbound data types

In an abstract sense, modelbound data can be anything – extensible!

In practice currently

Variable-like data

A collection of derived variables

\+ slightly more flexible data strucure

\- variables aren't communicated

Collisional-Radiative data

Complex data structures

More in next session

Setting modelbound data

Modelbound data is easily attached to models

```
mbData = sc.VarlikeModelboundData()
mbData.addVariable(
    "logLei",
    sc.derivationRule(
        "logLei" + ionSpeciesName, [electronTempVar, electronDensVar]
    ),
)
newModel.setModelboundData(mbData.dict())
```

Modelbound variables become available to terms

$$M_{ij} = cX_i(x)H_i(h)V_i(v)T(t)R_iC_jS_{ij}$$

```
varData = sc.VarData(reqRowVars=[ionDensVar], reqMBRowVars=["logLei"])
```

Generating variable-like modelbound data using calculation trees

It is possible to automatically build the required modelbound data

The model must be “simple” $M_{ij} = R_i S_{ij}$

```
def addNodeMatrixTermModel(
    wrapper: RKWrapper,
    modelTag: str,
    evolvedVar: str,
    termDefs: List[Tuple[ct.Node, str]],
    stencilData: Union[List[dict], None] = None,
):
    """Adds model with additive matrix terms of the form rowVar * implicitVar, where rowVar is a modelbound variable derived from a
    treeDerivation given a node. Optionally gives each matrix terms a different stencil.

    Args:
        wrapper (RKWrapper): Wrapper to add model to
        modelTag (str): Model tag for model to be added
        evolvedVar (str): Evolved variable for all matrix terms
        termDefs (List[Tuple[ct.Node,str]]): Term definitions. A list of (Node,implicitVarName) tuples, such that each matrix term is
        given by the variable calulated using the corresponding tuple's first component (the Node) and with the implicit variable name
        given by the second component
        stencilData (Union[List[dict, optional): Optional list of stencil data for each matrix term. Defaults to None.
    """
```

Note: If two or more terms in this model have the same required MB variables they will be added multiple times! In this case it is better to write your own model!

UK Atomic Energy Authority

Hands-on session