Ensemble Modeling

RAVEN Workshop

www.inl.gov

Idaho National Laboratory

Outline

- RAVEN models: brief overview
- Ensemble Modeling
  - Overview
  - Characteristics and limitations
- Application examples of Ensemble Modeling
- Hands-on:
  - Example using 2 external models
  - Example using the Code Interface we previously created

RAVEN models: overview

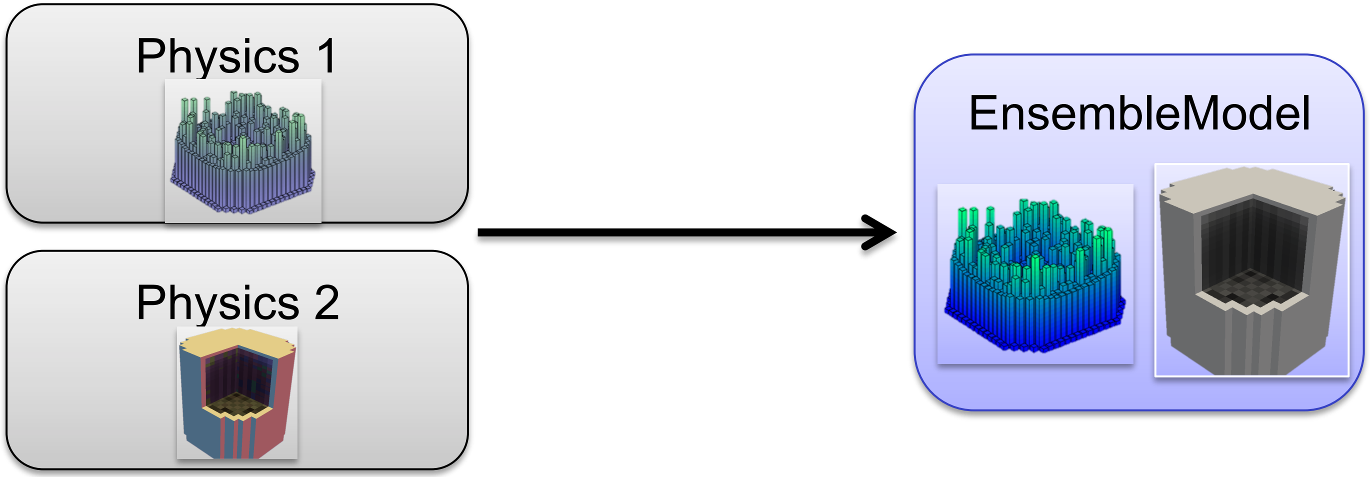
RAVEN models: a quick introduction

- RAVEN categorizes in its Models entity the following sub-entities:
  - Codes:
    - Aimed to interface with physical codes (e.g. RELAP5-3D, etc.)
  - ROMs:
    - Aimed to emulate the response of a system based on a simplified mathematical representation
  - External Models:
    - Aimed to provide to the user an easy way to implement sets of equations directly in RAVEN
  - Post-Processors:
    - Aimed to analyze the generated datasets (e.g. Statistical moments, Data Mining, etc.)
  - Ensemble Models:
    - Aimed to assemble multiple models

RAVEN ensemble modeling

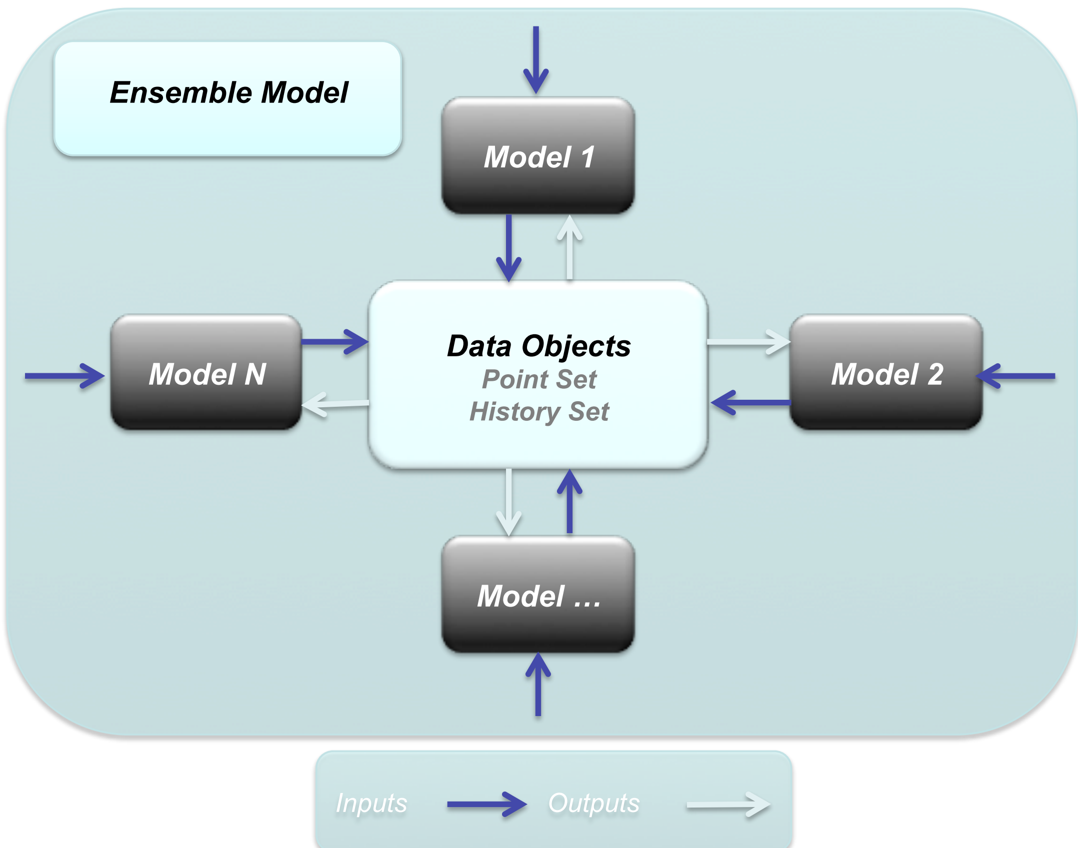
Ensemble Modeling Motivations

- In several cases multiple models need to interface with each other since the initial conditions of some are dependent on the outcomes of others
- In order to face this “problem” in the RAVEN framework, a new model category (e.g. class), named *EnsambleModel*, has been designed
- This class is able to assemble multiple models of other categories (i.e. Code, External Model, ROM), identifying:
  - the input/output connections
  - the order of execution
  - the parallel execution strategy for each sub-model



Ensemble Model

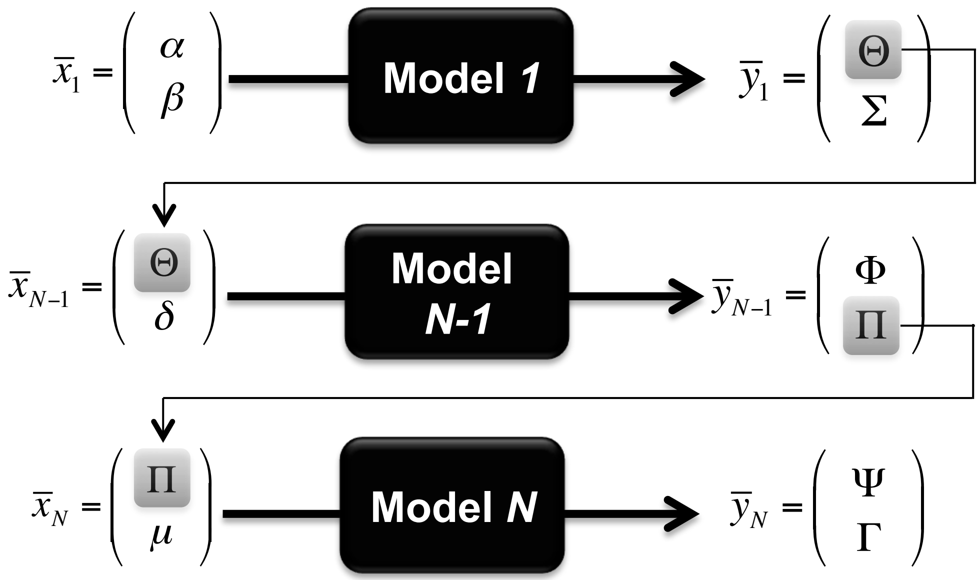
- A new model entity (e.g., class), named *EnsembleModel*, has been developed:
  - Assemble multiple models of other categories, identifying the input/ output connections and the order of execution

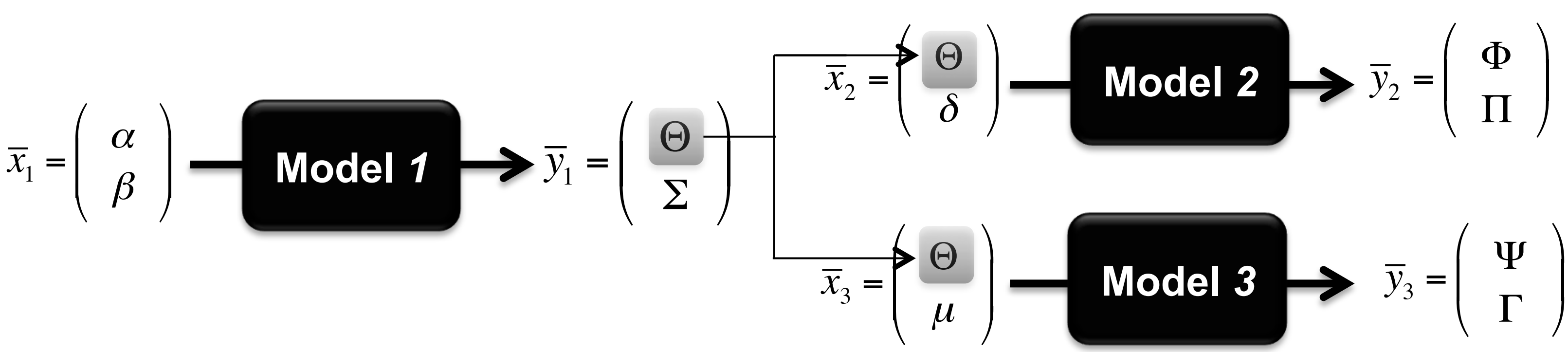


Ensemble Model: Main Characteristics

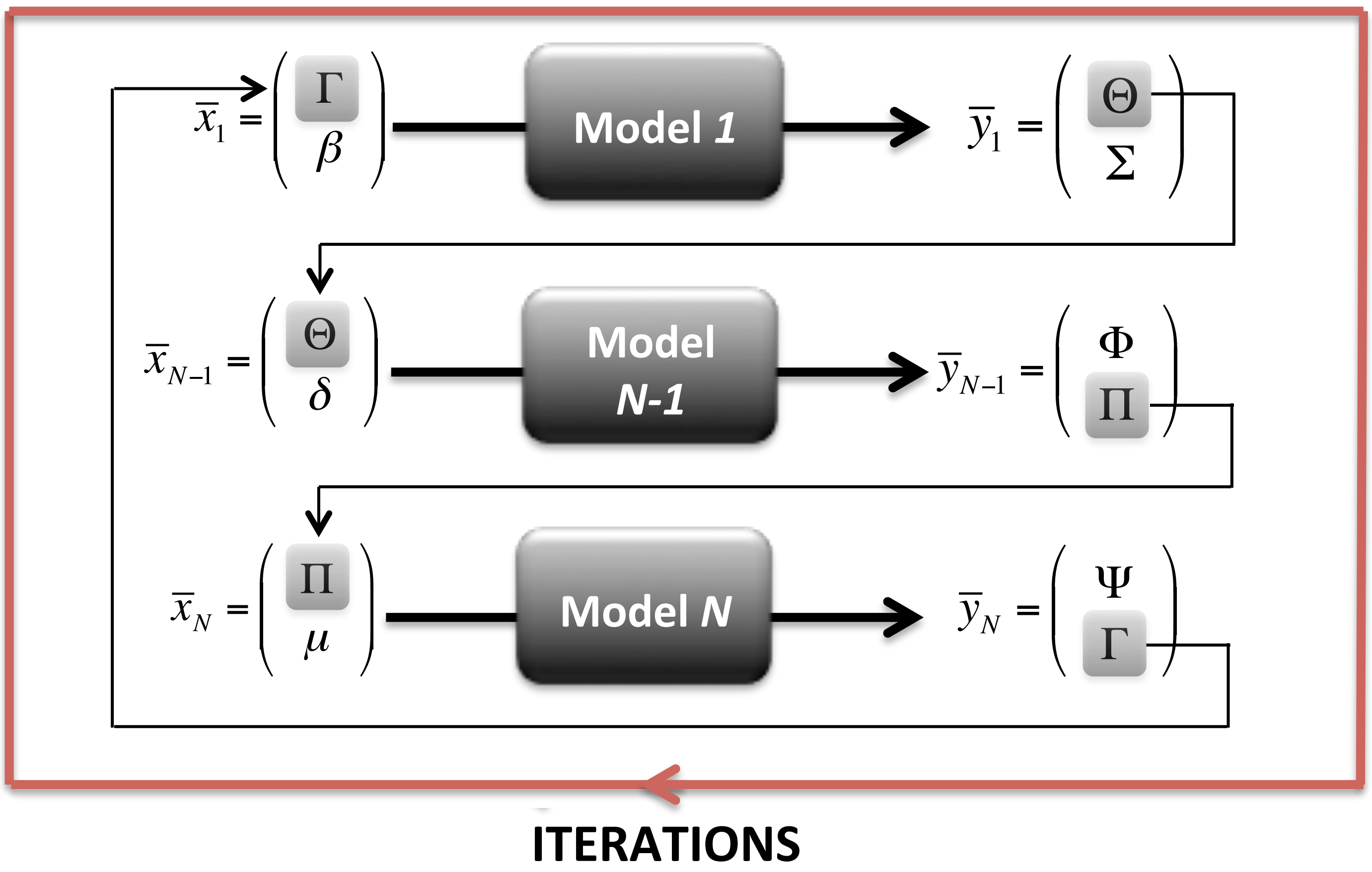
- The *EnsembleModel* entity has the following main characteristics:
  - Ability to link all the RAVEN Models:
    - Codes, ROMs, ExternalModels
  - Practical no limit on the number of Models in the Ensemble configuration
  - Capability to link the different Models through both scalar and vector variables (e.g. Max Cladding Temperature (scalar) or Power history (vector))
  - Capability to transfer meta-data from the different models (e.g. restart files, etc.)
- The current *EnsembleModel* entity is not indicated to handle high-density field data

Ensemble Model: Chain of Models





Ensemble Model: Non Linear

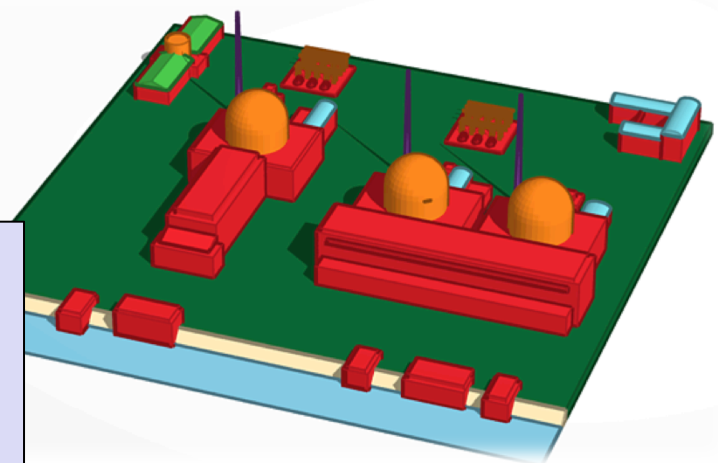


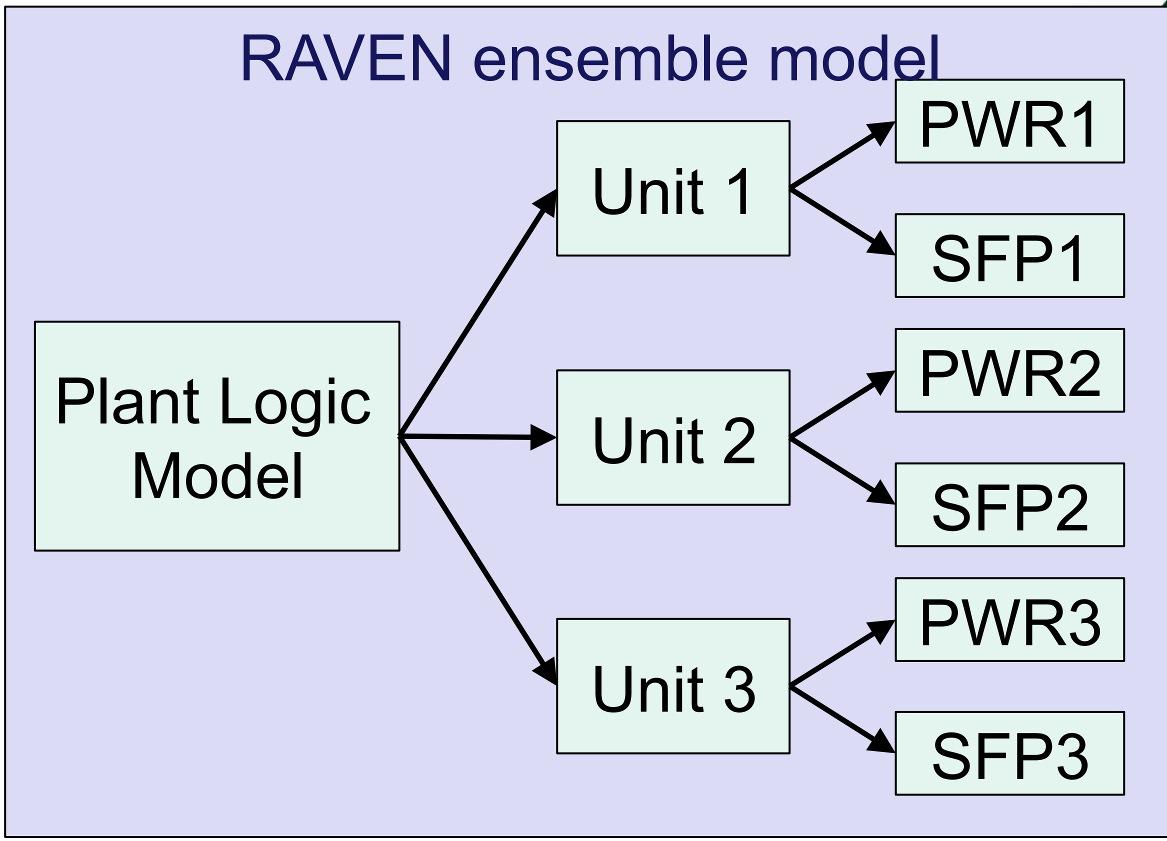
Idaho National Laboratory

Employing Ensemble Modeling in real applications

Ensemble model for Multi-Unit Power Plant: 1st Configuration

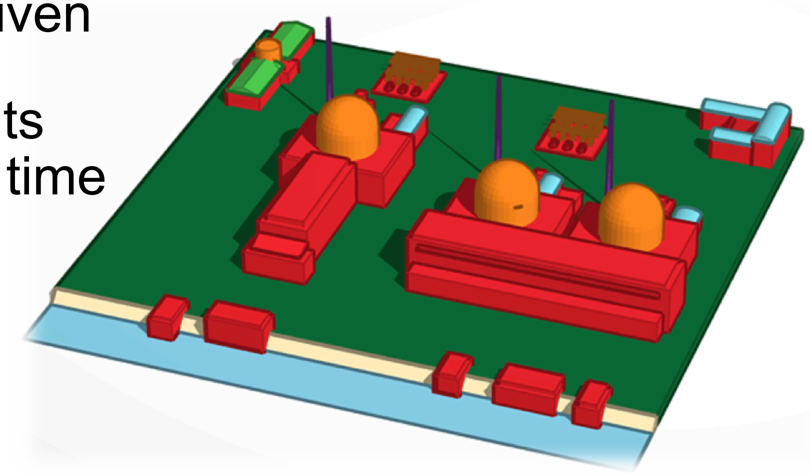
- Dynamic PRA for a Station Black Out Multi-Unit scenario



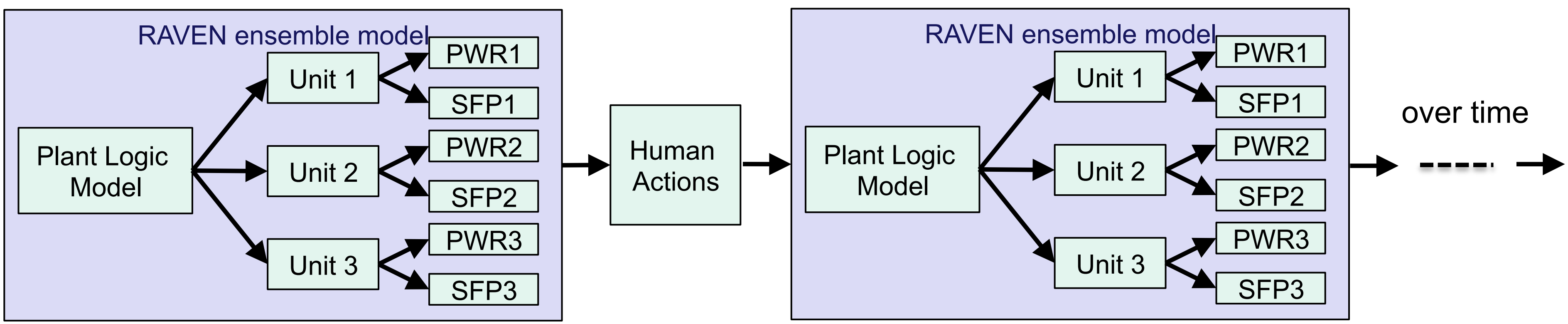


Ensemble model for Multi-Unit Power Plant: 2nd Configuration

- Exploiting the restart capability of the driven code, the *EnsembleModel* can be constructed through a chain of basic units that can be repeated, for example, over time

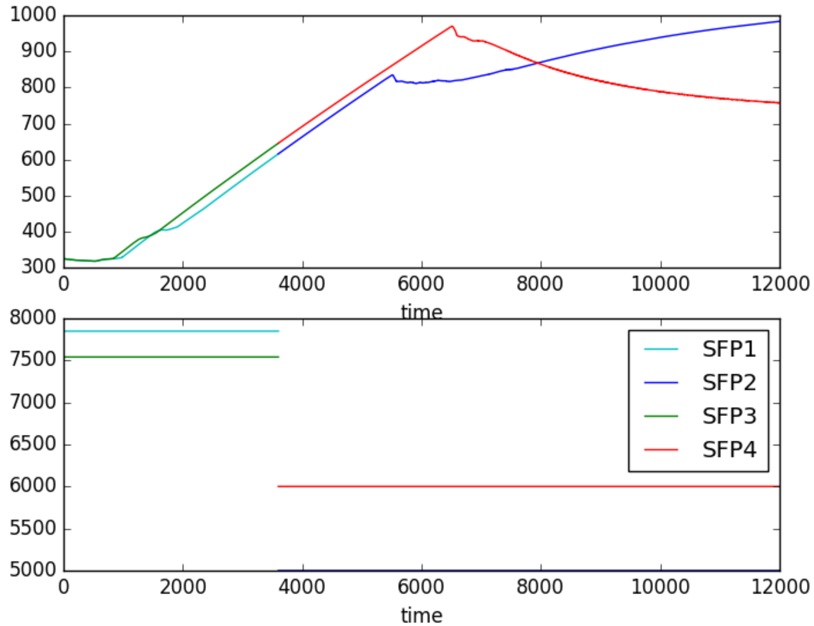


10 minutes

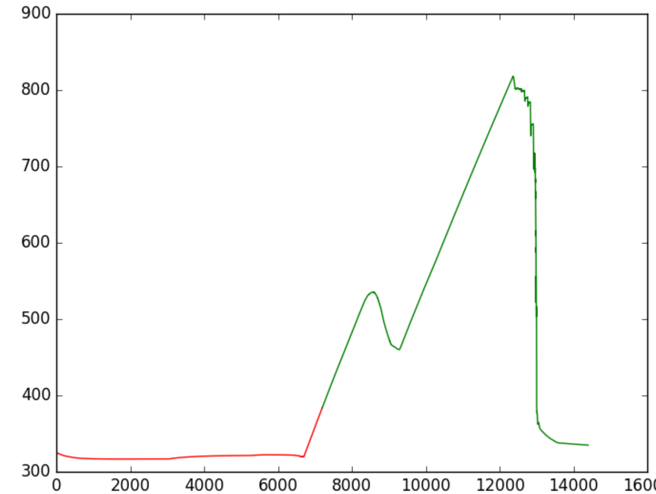


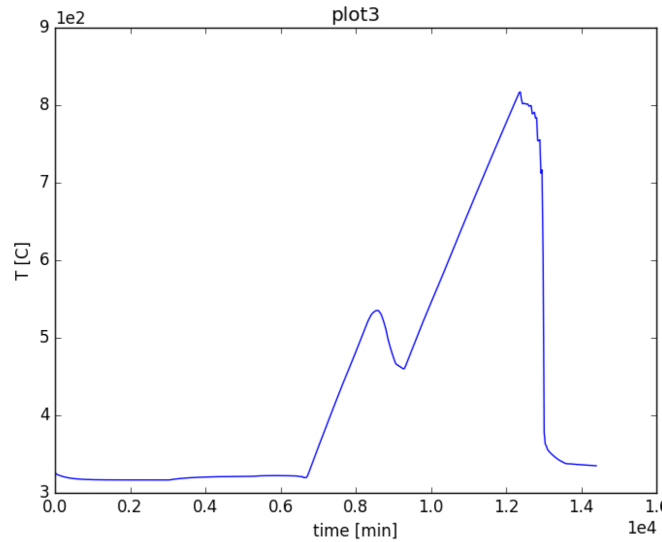
Ensemble model for Multi-Unit Power Plant: Preliminary results

4 Spent Fuel Pools



Transferring Restarts





Employing Ensemble modeling in RAVEN: 2 Examples

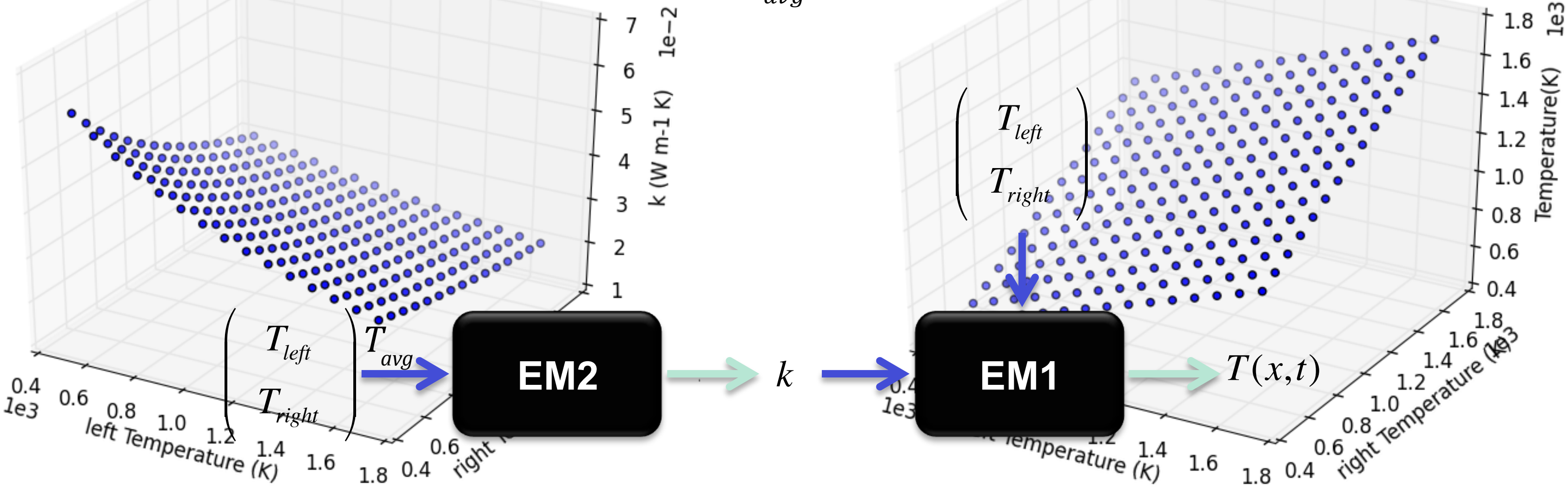
Ensemble Model: Example 1 specifications

- 1-Dimensional heat conduction transient (in a slab of thickness L=1 m):
  - *EM1*, heat conduction partial differential equation:

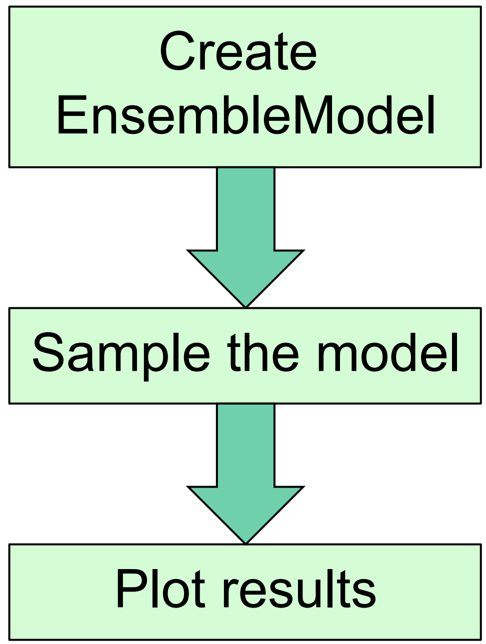
$$\begin{cases} \dfrac{dT(x,t)}{dt} = k\dfrac{d^2T(x,t)}{dx^2} \\ T(0,t) = T_{left} \\ T(L,t) = T_{right} \end{cases}$$

  - *EM2*, thermal conductivity (input of *EM1*) as function of the average temperature in the slab boundary conditions:

$$k = \frac{38.23}{129.2 + T_{avg}} + 6.077E^{-13} * T_{avg}$$



Workflow



Input File name: ensemble_model_ext_models.xml

Create an Ensemble model of a code and an ExternalModel

Distributions | Models | Samplers | Databases | DataObjects | OutStreams

Steps

```
<Distributions>
    <Uniform name=’leftTemperatureDist'>
        <upperBound>500</upperBound>
        <lowerBound>1700</lowerBound>
    </Uniform>
    <Uniform name=’rightTemperatureDist'>
        <upperBound>500</upperBound>
        <lowerBound>1700</lowerBound>
    </Uniform>
</Distributions>
```

Distribution specifications

Create an Ensemble model of a code and an ExternalModel

Distributions	Models	Samplers	Databases	DataObjects	OutStreams
		Steps			

```
<Models>
    ...
    <ExternalModel name='heatTransfer' subType='' ModuleToLoad='EM1linear'>
       <variables>leftTemperature,rightTemperature,k,solution</variables>
    </ExternalModel>
  <ExternalModel name='thermalConductivityCompuation' subType='' ModuleToLoad='EM2linear'>
       <variables>leftTemperature,rightTemperature,k,averageTemperature</variables>
    </ExternalModel>
    ...
</Models>
```

List of Models we are going to use

Create an Ensemble model of a code and an ExternalModel

Distributions	Models	Samplers	Databases	DataObjects	OutStreams

Steps

```
<Models>
   ...
   <EnsembleModel name='codeAndExtModel' subType=''>
       <Model class='Models' type='ExternalModel'>
         thermalConductivityComputation
         <Input class="Files" type="">inputHolder</Input >
         <TargetEvaluation class='DataObjects'
             type='PointSet'>thermalConductivityComputationContainer
         </TargetEvaluation>
       </Model>
       <Model class='Models' type='Code'>
         heatTransfer
         <Input class="Files" type="">inputHolder</Input >
         <TargetEvaluation class='DataObjects' type='PointSet'>
            heatTransferContainer</TargetEvaluation>
       </Model>
    </EnsembleModel>
    ...
</Models>
```

Inputs of this model

Exclusive output

Create an Ensemble model of a code and an ExternalModel

Distributions | Models | Samplers | Databases | **DataObjects** | OutStreams

Steps

Link

```
<DataObjects>
    <PointSet name=’heatTransferContainer'>
      <Input>leftTemperature,rightTemperature,k</Input>
      <Output>solution</Output>
    </PointSet>
    <PointSet
         name=’thermalConductivityComputationContainer'>
        <Input>leftTemperature,rightTemperature</Input>
        <Output>k</Output>
    </PointSet>
    <PointSet name=’metaModelOutputTest'>
        <Input>leftTemperature,rightTemperature</Input>
        <Output>k,solution</Output>
    </PointSet>
</DataObjects>
```

Create an Ensemble model of two model

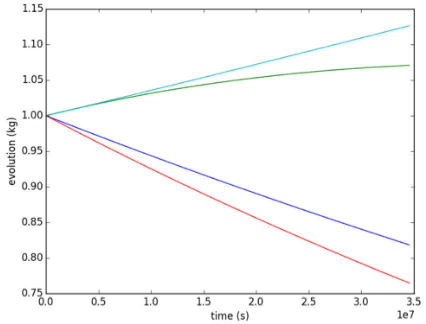
Let’s run the code…

Exercise 2: Create an EnsembleModel of a Code and an ExternalModel

Ensemble Model: Example 2 specifications

- Also codes can be used in the Ensemble modeling.
  - *EM1: Code, Analytical Bateman*
    - Transmutation

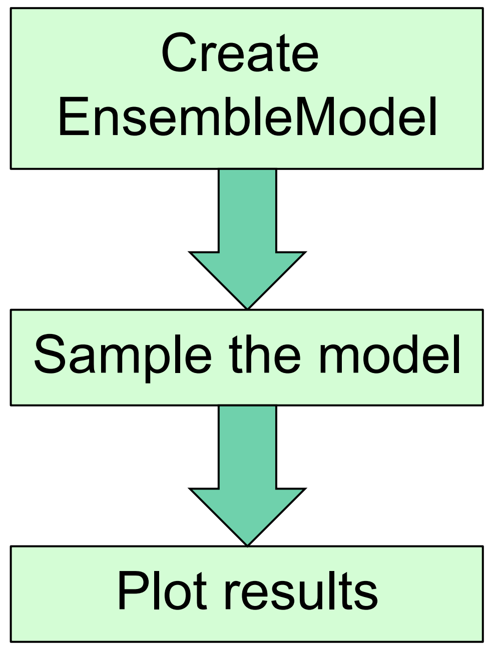
$$\begin{cases} \dfrac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{S} - \mathbf{L} \\ \mathbf{X}(t=0) = \mathbf{X_0} \end{cases}$$



  - *EM2: External Model, convert final outcomes of EM1 into atom densities*

$$\begin{pmatrix} \sigma_A \\ \sigma_B \\ \lambda_A \\ \lambda_A \end{pmatrix} \rightarrow \boxed{\textbf{EM1}} \rightarrow \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \rightarrow \boxed{\textbf{EM2}} \rightarrow \begin{pmatrix} densA \\ densB \\ densC \\ densD \end{pmatrix}$$

Workflow



Input File name: ensamble_model_example.xml

Create an Ensemble model of a code and an ExternalModel

Distributions	Models	Samplers	Databases	DataObjects	OutStreams

Steps

```
<Distributions>
    <Uniform name='sigma'>
        <upperBound>1000</upperBound>
        <lowerBound>0.0</lowerBound>
    </Uniform>
    <Uniform name='decayConstant'>
        <upperBound>1.e-7</upperBound>
        <lowerBound>1.e-8</lowerBound>
    </Uniform>
</Distributions>
```

Distribution specifications

Create an Ensemble model of a code and an ExternalModel

Distributions	**Models**	Samplers	Databases	DataObjects	OutStreams
		Steps			

```
<Models>
    ...
    <ExternalModel name=’convertToAtomDensity' subType=’’ ModuleToLoad=’toAtomDens’>
       <variables>A,B,C,D,densA,densB,densC,densD</variables>
    </ExternalModel>
    <Code name=’testModel' subType=’GenericCode’>
       <executables>ensembleModelWithCode/AnalyticalDplMain.py</executables>
       <clargs arg=”” extension=’.xml’ type=’input’ />
       <clargs arg=”” extension=’.csv’ type=’output’ />
       <prepend>python</prepend>
    </Code>
    ...
</Models>
```

List of Models we are going to use

Create an Ensemble model of a code and an ExternalModel

Distributions	**Models**	Samplers	Databases	DataObjects	OutStreams
		Steps			

```
<Models>
   ...
   <EnsembleModel name='codeAndExtModel' subType=''>
       <Model class='Models' type='ExternalModel' inputNames='inPlaceHolder'>
         convertToAtomDensity
         <Input class="Files" type="">referenceInput.xml</Input >
         <TargetEvaluation class='DataObjects' type='PointSet'>convertedData
         </TargetEvaluation>
       </Model>
       <Model class='Models' type='Code'>
         testModel
         <Input class="Files" type="">inputHolder</Input >
         <TargetEvaluation class='DataObjects' type='PointSet'>sampleMC</TargetEvaluation>
       </Model>
    </EnsembleModel>
    ...
</Models>
```

Inputs of this model

Exclusive output

Create an Ensemble model of a code and an ExternalModel

Distributions	Models	Samplers	Databases	DataObjects	OutStreams

Steps

Link

```xml
<DataObjects>
    <PointSet name='convertData'>
      <Input>A,B,C,D</Input>
      <Output>densA,densB,densC,densD</Output>
    </PointSet>
    <PointSet name='sampleMC'>
        <Input>sigma-A,sigma-B,decay-A,decay-B</Input>
        <Output>A,B,C,D</Output>
    </PointSet>
    <PointSet name='finalResponses'>
        <Input>sigma-A,sigma-B,decay-A,decay-B</Input>
        <Output>A,B,C,D,densA,densB,densC,densD</Output>
    </PointSet>
</DataObjects>
```

Create an Ensemble model of two model

Let’s run the code…

Thank you

Questions?