



IDAES
Institute for the Design of
Advanced Energy Systems

Progress on IDAES Model Libraries

Andrew Lee

National Energy Technology Laboratory

May 16, 2019



**Sandia
National
Laboratories**



West Virginia University

Carnegie Mellon



UNIVERSITY OF
NOTRE DAME



U.S. DEPARTMENT OF
ENERGY

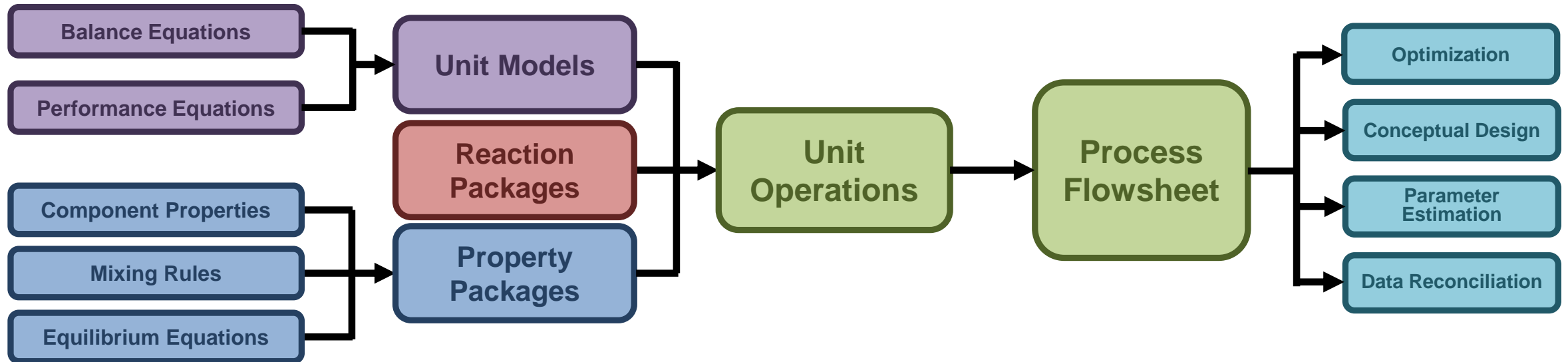
IDAES Model Libraries

Model libraries are a core component of IDAES

- Libraries of models for unit operations and thermophysical properties
- Open, modular structure
- Models are all optimization-ready and extensible
- Models can easily be used directly from libraries

IDAES Modeling Framework

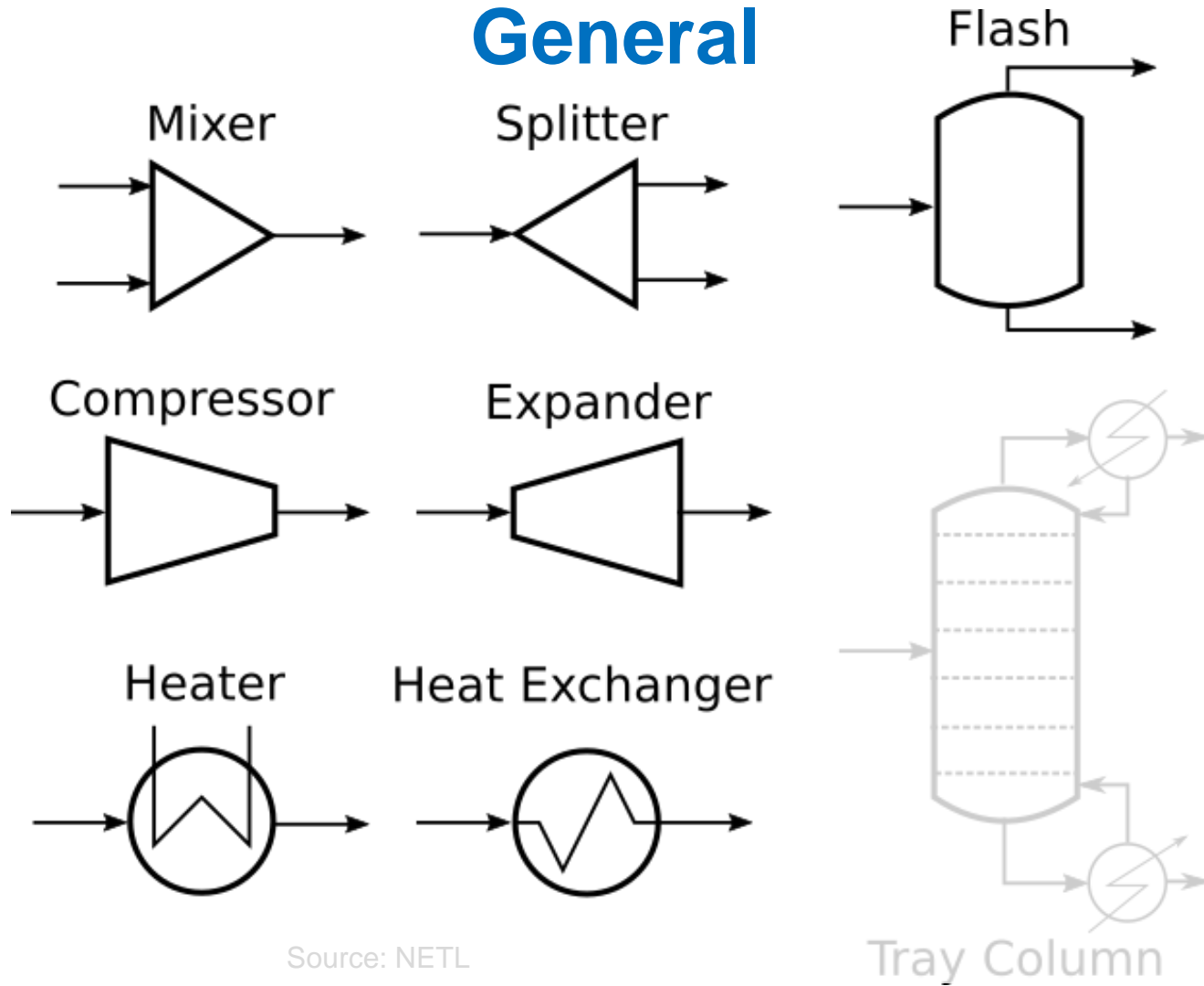
- Combines features of process simulators and equation-oriented packages



- Flexible, modular framework allows for models with different levels of rigor and complexity
- Can mix models of different rigor for individual units and physical properties

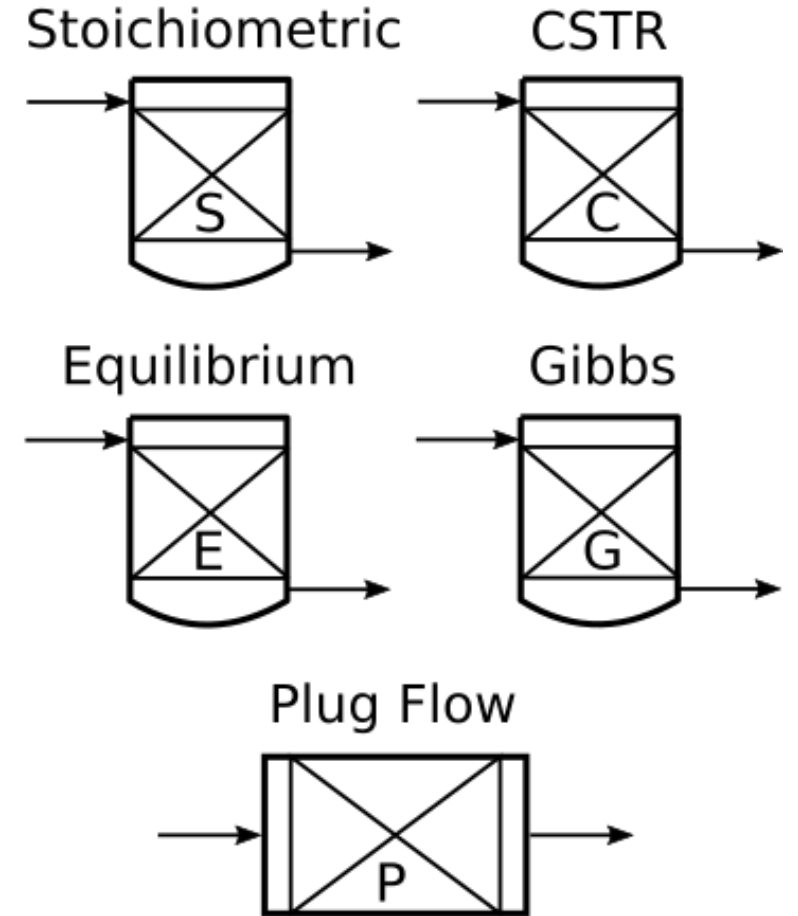
Optimization-Ready Model Library

General





Source: NETL

Reactors



Libraries are Fully Documented



latest

- Installation
- IDAES Modeling Standards
- Core Library
- Unit Model Library
 - Continuous Stirred Tank Reactor
 - Degrees of Freedom
 - Model Structure
 - Additional Constraints
 - Variables
 - CSTR Class
 - CSTRData Class
 - Equilibrium Reactor
 - Feed Block

[Docs](#) » [Unit Model Library](#) » Continuous Stirred Tank Reactor

[Edit on GitHub](#)

Continuous Stirred Tank Reactor

The IDAES CSTR model represents a unit operation where a material stream undergoes some chemical reaction(s) in a well-mixed vessel.

Degrees of Freedom

CSTRs generally have one degree of freedom. Typically, the fixed variable is reactor volume.

Model Structure

The core CSTR unit model consists of a single `ControlVolume0D` (named `control_volume`) with one Inlet Port (named `inlet`) and one Outlet Port (named `outlet`).



Additional Constraints

CSTR units write the following additional Constraints beyond those written by the ControlVolume Block.

$$X_{t,r} = V_t \times r_{t,r}$$

where $X_{t,r}$ is the extent of reaction of reaction r at time t , V_t is the volume of the reacting material at time t (allows for varying reactor volume with time) and $r_{t,r}$ is the volumetric rate of reaction of reaction r at time t (from the outlet property package).

Libraries are Fully Documented



latest

- Installation
- IDAES Modeling Standards
- Core Library
- Unit Model Library
 - Continuous Stirred Tank Reactor
 - Degrees of Freedom
 - Model Structure
 - Additional Constraints
 - Variables
 - CSTR Class
 - CSTRData Class
 - Equilibrium Reactor
 - Feed Block
 - ...

[Docs](#) » [Unit Model Library](#) » Continuous Stirred Tank Reactor

[Edit on GitHub](#)

Continuous Stirred Tank Reactor

The IDAES CSTR model represents a unit operation where a material stream undergoes some chemical reaction(s) in a well-mixed vessel.

Degrees of Freedom

CSTRs generally have one degree of freedom. Typically, the fixed variable is reactor volume.

Model Structure

The core CSTR unit model consists of a single `ControlVolume0D` (named `control_volume`) with one Inlet Port (named `inlet`) and one Outlet Port (named `outlet`).

Additional Constraints

CSTR units write the following additional Constraints beyond those written by the ControlVolume Block.

$$X_{t,r} = V_t \times r_{t,r}$$

where $X_{t,r}$ is the extent of reaction of reaction r at time t , V_t is the volume of the reacting material at time t (allows for varying reactor volume with time) and $r_{t,r}$ is the volumetric rate of reaction of reaction r at time t (from the outlet property package).

Libraries are Fully Documented

Additional Constraints

Variables

CSTR Class

CSTRData Class

Equilibrium Reactor

Feed Block

Feed Block with Flash

Flash Unit

Gibbs Reactor

Heater

HeatExchanger (0D)

Heat Exchangers (1D)

Mixer

Plug Flow Reactor

Pressure Changer

Product Block

Separator

StateJunction Block

Stoichiometric (Yield) Reactor

Translator Block

Power Generation Models

Property Model Library

Additional Constraints

CSTR units write the following additional Constraints beyond those written by the ControlVolume Block.

$$X_{t,r} = V_t \times r_{t,r}$$

where $X_{t,r}$ is the extent of reaction of reaction r at time t , V_t is the volume of the reacting material at time t (allows for varying reactor volume with time) and $r_{t,r}$ is the volumetric rate of reaction of reaction r at time t (from the outlet property package).

Variables

CSTR units add the following additional Variables beyond those created by the ControlVolume Block.

Variable	Name	Notes
V_t	volume	If <code>has_holdup = True</code> this is a reference to <code>control_volume.volume</code> , otherwise a Var attached to the Unit Model
Q_t	heat	Only if <code>has_heat_transfer = True</code> , reference to <code>control_volume.heat</code>

CSTR Class

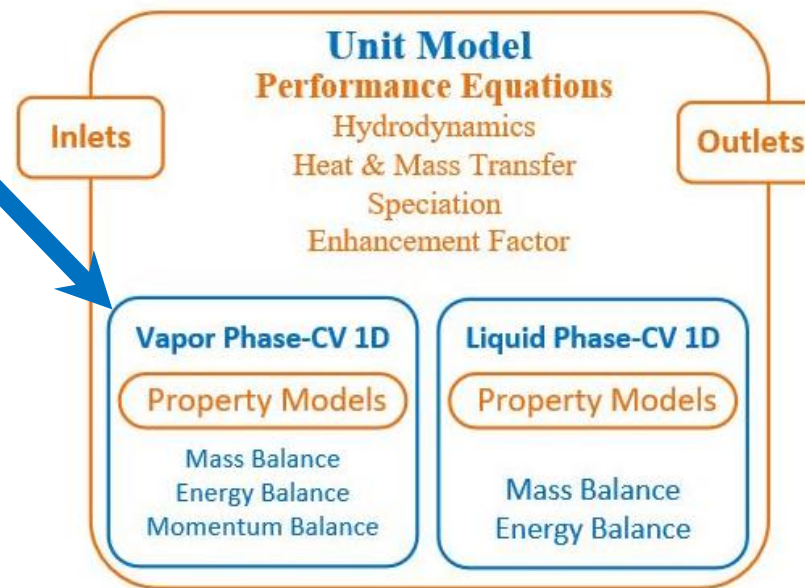
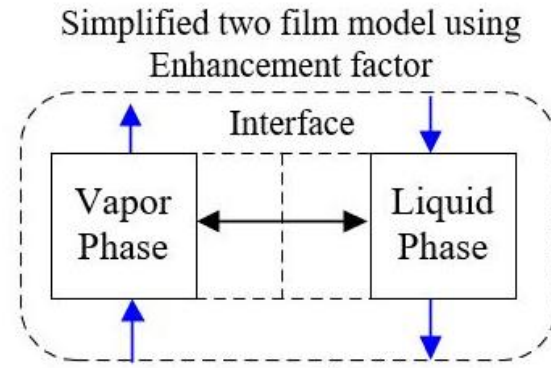
```
class idaes.unit_models.cstr.CSTR(*args, **kwargs)
```

- Parameters:
- `rule` (*function*) – A rule function or None. Default rule calls `build()`.
 - `concrete` (*bool*) – If True, make this a toplevel model. **Default** - False.
 - `ctype` (*str*) – Pyomo ctype of the block. **Default** - "Block"
 - `default` (*dict*) –

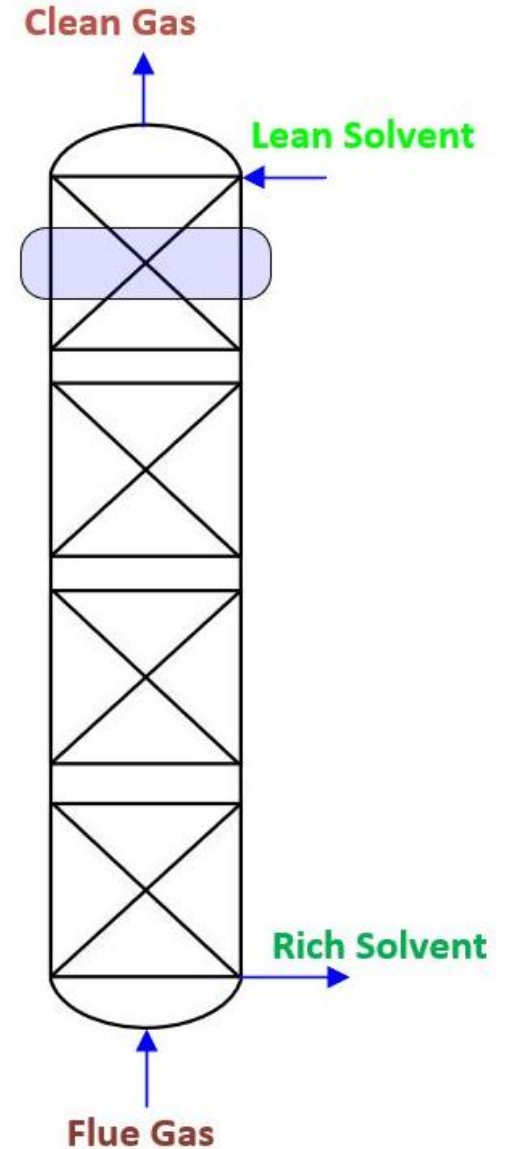
Modular Framework for Complex Processes

Control Volumes

- Automatically links to property packages
- Tools for automating:
 - Material balances
 - Energy balances
 - Momentum balances
- 0- and 1-D forms
- Steady-state and dynamic
- Methods to assist with initialization



Continuous Differential
Contactor Model



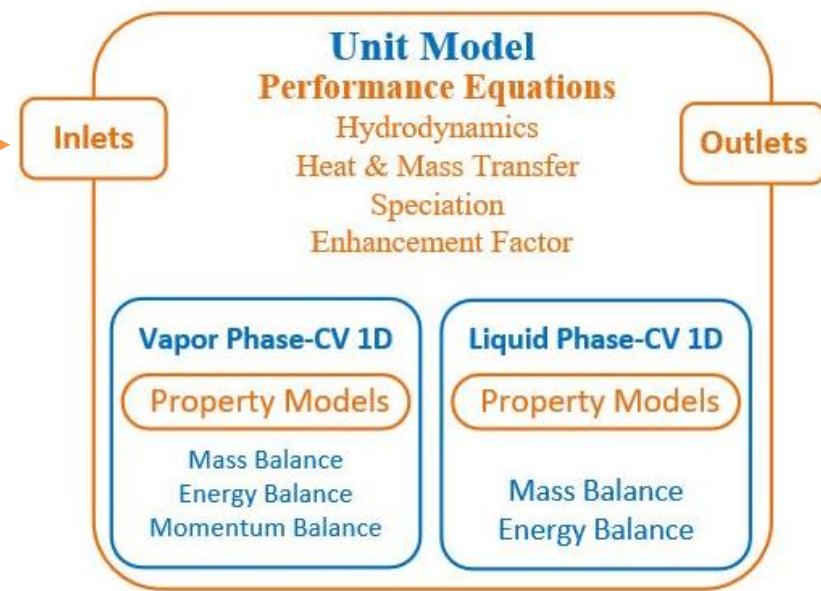
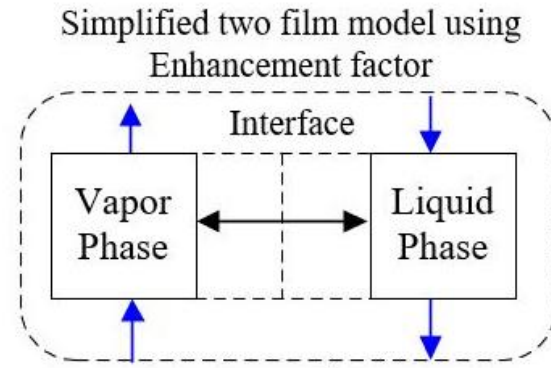
Packed Column

Modular Framework for Complex Processes

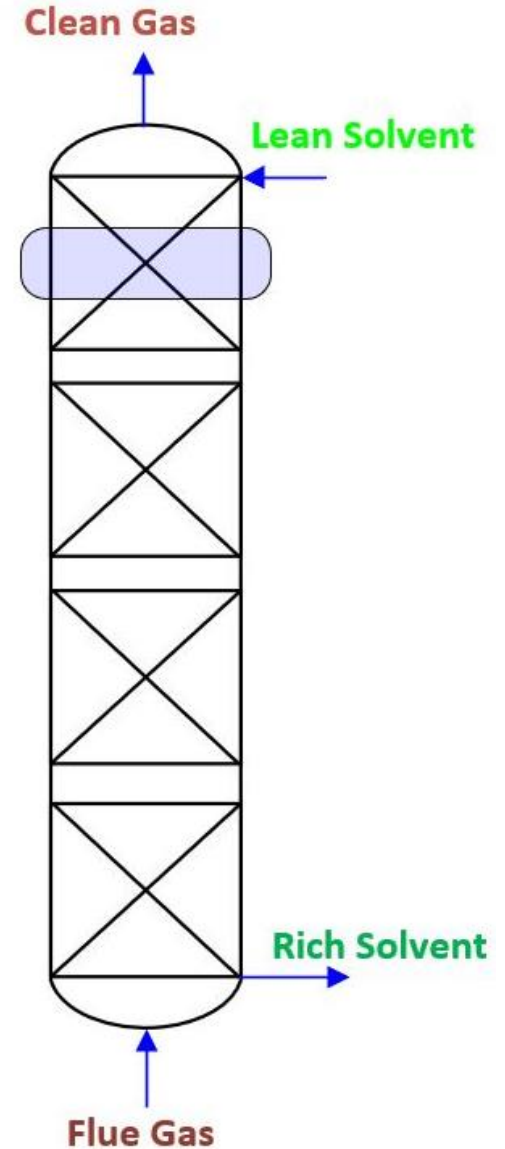
Control Volumes

Ports

- Points to link between units
- Coupled with control volumes
- Automatically determines variables to link



Continuous Differential
 Contactor Model



Packed Column

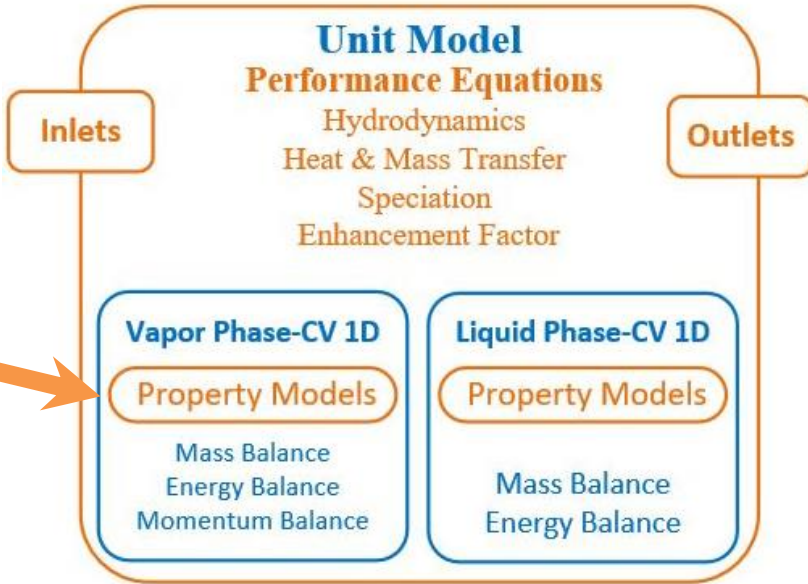
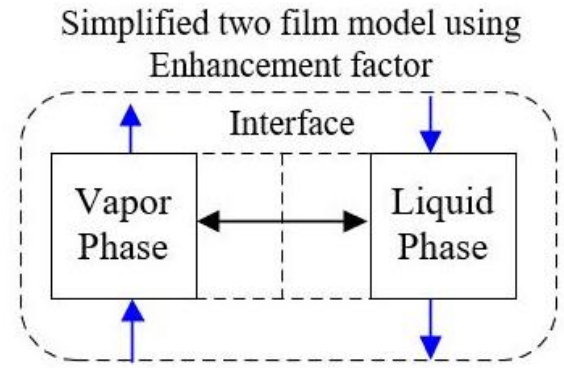
Modular Framework for Complex Processes

Control Volumes

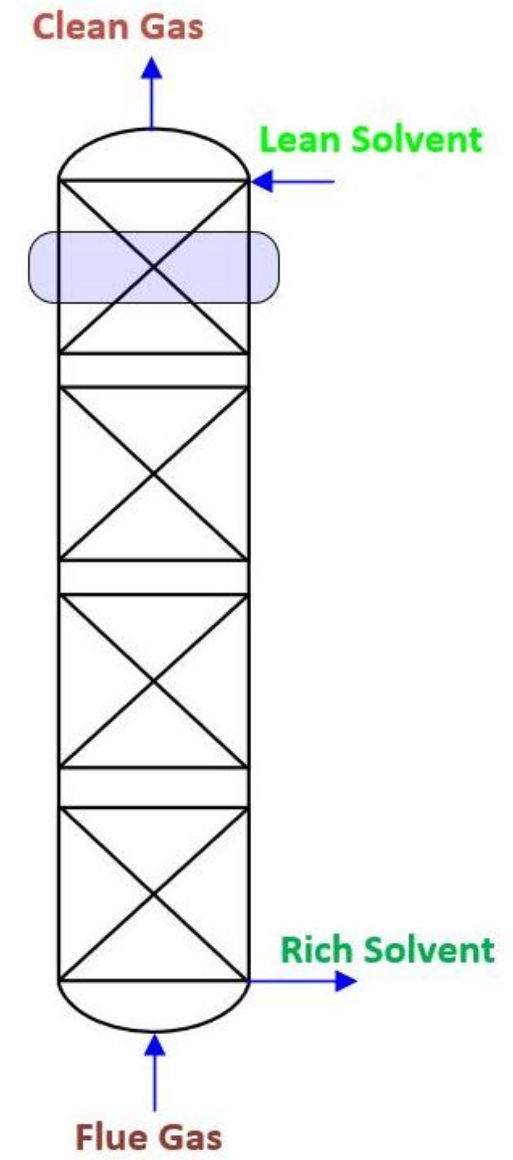
Ports

Property Models

- Instances of property models



Continuous Differential Contactor Model

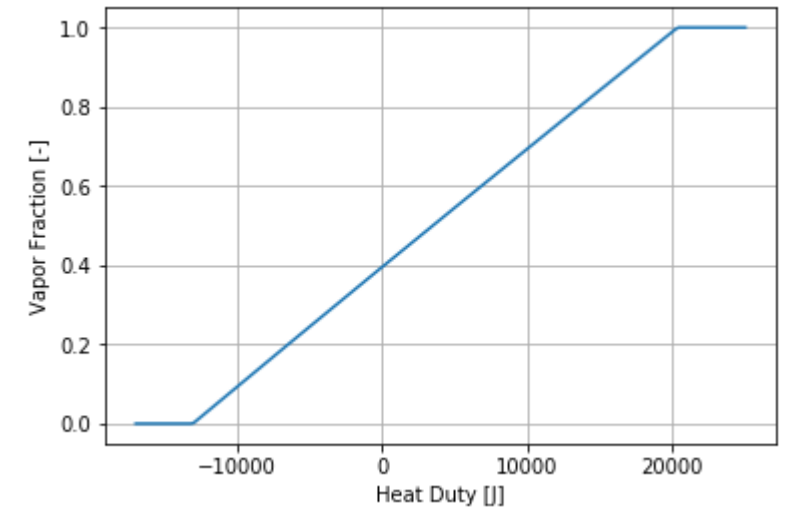


Packed Column

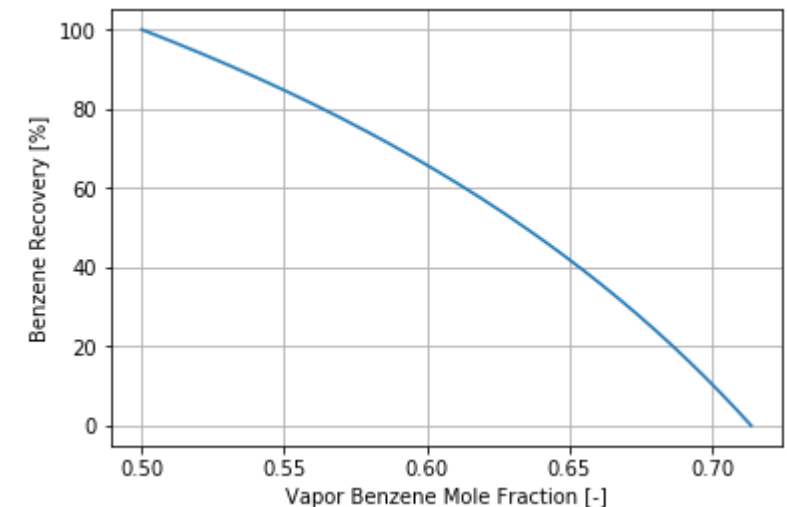
Property Models

- Optimization-ready unit models need optimization ready property packages
- Developing a limited library of standard property model forms
- Ideal-Ideal model for vapor-liquid equilibrium
 - Prototype for Ideal-NRTL and Ideal-Wilson
- Modular form allows different combinations of vapor and liquid phase properties

Vapor Fraction versus Heat Duty



Benzene Recovery versus Purity



Integrating New Property Models

- Modular form allows for development of new property models
 - Add new component and parameters to existing models
 - Develop new correlations
- Models for each physical quantity are modular
 - Can model each property separately
 - Mix-and-match models for different properties
- IDAES tools for creating new property models

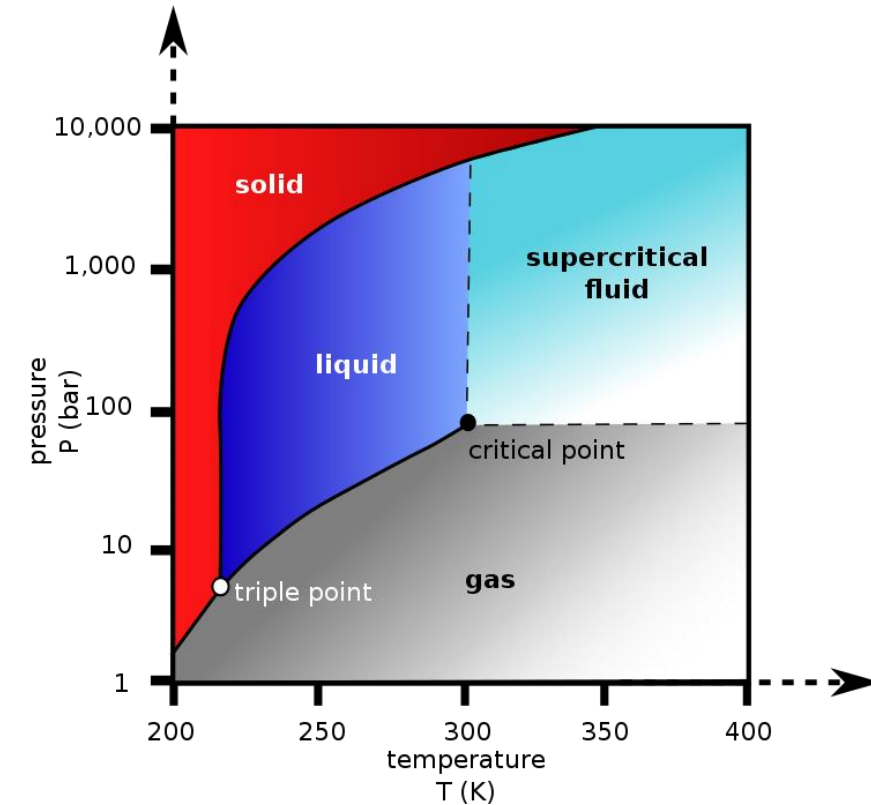


Image from
https://en.wikipedia.org/wiki/Supercritical_carbon_dioxide

Using Model Libraries

Import desired models

```
from idaes.unit_models import CSTR  
from idaes.properties import Ideal
```

Create a flowsheet

```
fs = Flowsheet(default={"dynamic":False})
```

Create a property instance

```
fs.ideal_props = Ideal.Parameters()
```

Create a model instance

```
fs.reactor = CSTR(default={  
    "property_package": fs.ideal_props})
```

Connect units

```
fs.stream = Arc(source=fs.reactor.outlet,  
                destination=fs.unit2.inlet)
```

Fix conditions

```
fs.reactor.volume.fix(10)
```

Initialization of Models

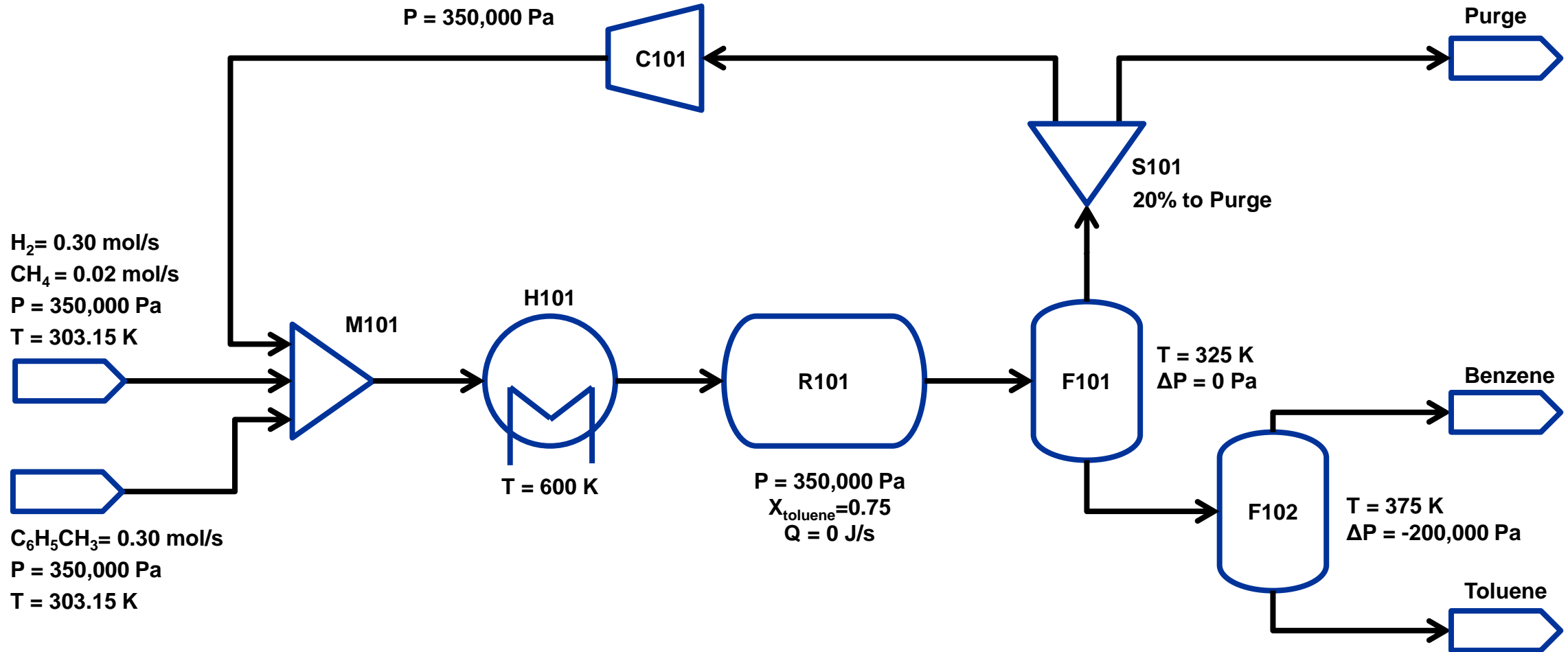
- All IDAES models contain initialization methods
 - unit models
 - control volumes
 - property packages

```
fs.reactor.initialize(initial_guesses)
```

- Facilitates a hierarchical, modular approach to initialization
- Developing tools to automate much of the process



Workshop Flowsheet



Summary

- Core libraries contain models for most common unit operations
- Library of basic property methods and tools for generating new models
- All models are optimization ready
- Modular structure designed for extensibility





IDAES
Institute for the Design of
Advanced Energy Systems

***Disclaimer** This presentation was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.*

***Attribution:** KeyLogic Systems, Inc.'s contributions to this work were funded by the National Energy Technology Laboratory under the Mission Execution and Strategic Analysis contract (DE-FE0025912) for support services.*