Hybrid Simulation of a Dynamic System

2018 AIChE Spring Meeting
Process Research & Innovation

James C Cross III & Wendy C Young

April 24, 2018

Overview

To objective of this work is to demonstrate real-time data exchange between CHEMCAD and MATLAB.

- Enabling functionality
- Case study
- Process flowsheet
- Custom UnitOp
- Steady-state validation
- Dynamic simulations

→ Integration of the platforms affords process engineers unlimited customization capabilities.

MATLAB API for COM Automation Server

MATLAB offers an Application Programming Interface (API) to enable connectivity to other common software platforms.

Common use cases:

- VBA .NET Client
- C# Client

MATLAB API for COM Automation Server

R2018a

Write COM applications to work with MATLAB®

Automation is a COM protocol that allows one application (the *controller* or *client*) to control objects exported by another application (the *server*). MATLAB supports COM Automation server capabilities on Microsoft® Windows® operating systems. Any Windows program that can be configured as an Automation controller can control MATLAB. Some examples are Microsoft Excel® and Microsoft Access™, and many Microsoft Visual Basic® and Microsoft Visual C++® programs.

If you build client applications using C/C++, or Fortran, then use MATLAB Engine Applications instead of an Automation server.

Functions

`actxGetRunningServer`	Handle to running instance of Automation server
`enableservice`	Enable, disable, or report status of MATLAB Automation server
`Execute`	Execute MATLAB command in Automation server
`Feval`	Evaluate MATLAB function in Automation server
`GetCharArray`	Character array from Automation server
`GetFullMatrix`	Matrix from Automation server workspace
`GetVariable`	Data from variable in Automation server workspace
`GetWorkspaceData`	Data from Automation server workspace
`MaximizeCommandWindow`	Open Automation server window
`MinimizeCommandWindow`	Minimize size of Automation server window
`PutCharArray`	Store character array in Automation server
`PutFullMatrix`	Matrix in Automation server workspace
`PutWorkspaceData`	Data in Automation server workspace
`Quit`	Terminate MATLAB Automation server
`regmatlabserver`	Register current MATLAB as Automation server

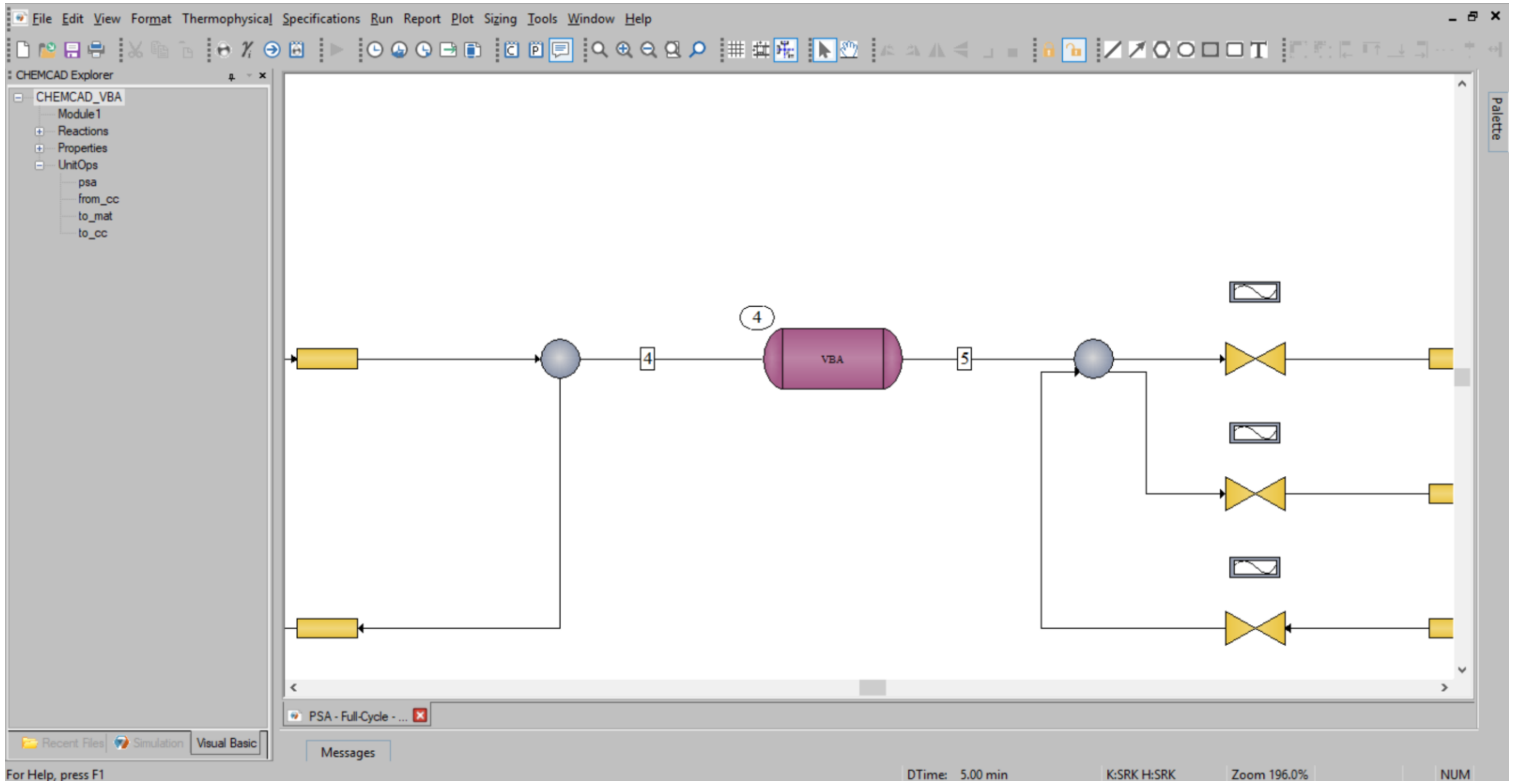
Since CHEMCAD and MATLAB can both communicate with VBA, information exchange is possible!

CHEMCAD VBA Unit Ops

CHEMCAD offers customizable unit operations configurable via Visual Basic, C++, or Excel.

Custom UnitOp examples:

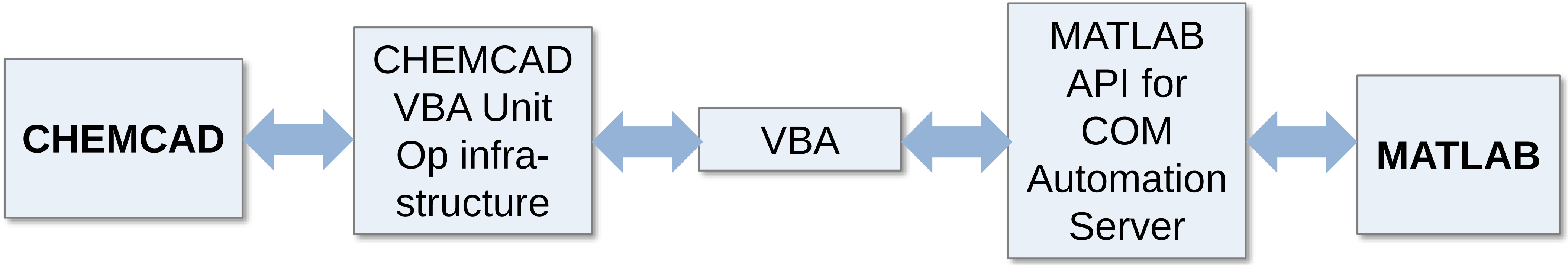
- Proprietary membrane separation units
- Fuel cells
- Specialized solids handling



In this work, we engage the embedded VBA interface to call the MATLAB API for COM Automation Server.

Platform Integration Overview

VBA is the "middleware" communicating with both CHEMCAD and MATLAB.



Manages overall simulation

When VBA UnitOp is involved, BCs are passed → BCs are packaged → Func eval call is made to API → API manages function call → Model takes inputs, calculates outputs

VBA UnitOp state is returned ← Outputs are passed into the simulation ← Outputs are assigned to CC variables ← API packages outputs ← Model takes inputs, calculates outputs

CHEMCAD manages both the VBA and MATLAB instances.

Dynamic System – Case Study

Process context: **Pressure Swing Adsorption (PSA)**

Case study: **Dynamic flow through an adsorbent bed**

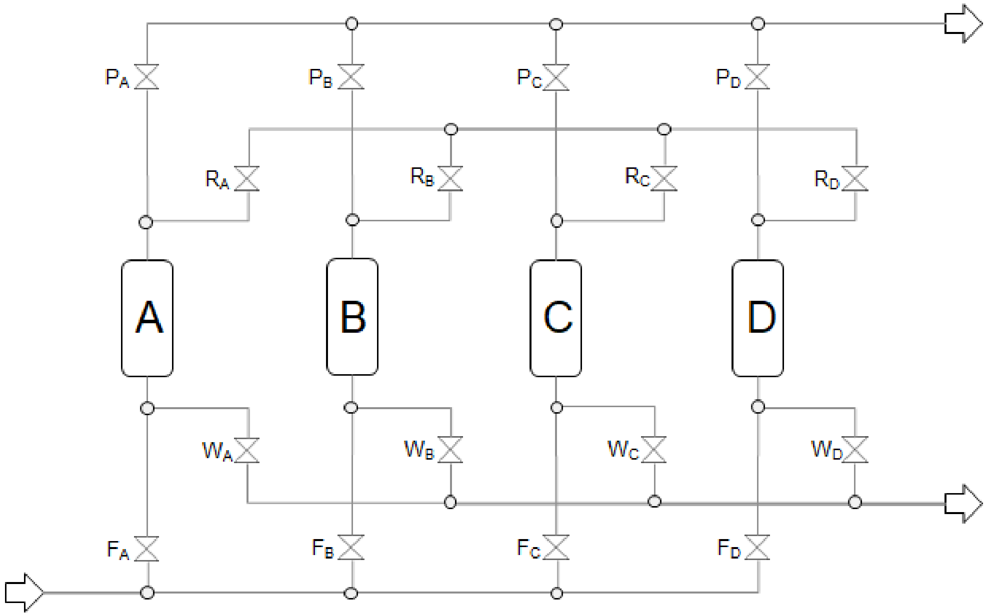
Select uses:

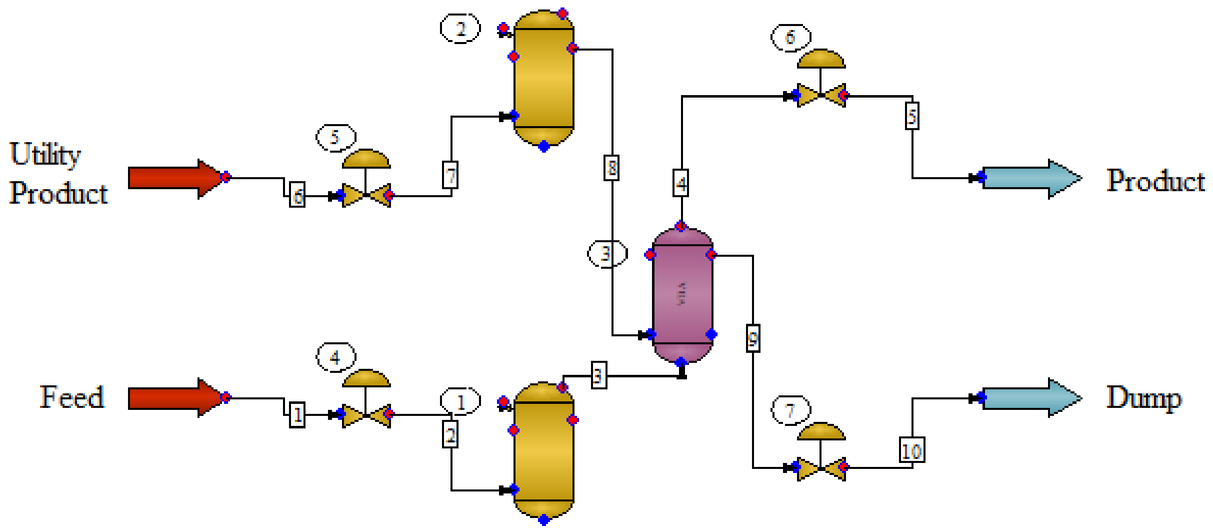
- Air driers
- CO2 removal
- CH4/H2
- H2S removal

http://www.xebecinc.com/biogas-purification-overview.php

Process Context

There are 6 bed states: ads, depress, purge supply, blowdown, purge receive, repress – the valve sequence can be a little confusing.



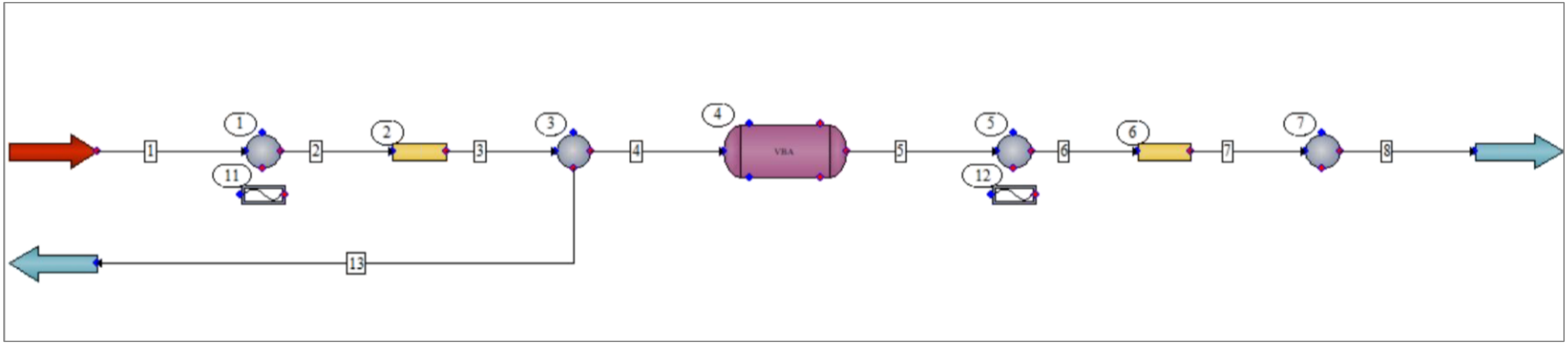


Basic valve schedule

VALVES	start	0-60s	60-90s	90-120s	120-150s	150-180s	180-210s	210-225s	225-240s	240-300s
CV-4	O	O	X	X	X	X	X	X	O	O
CV-6	X	O	X	X	X	X	X	X	X/O	O
CV-5	X	X	O	O	X	O	O	O	X	X
CV-7	X	X	X	X	O	O	X	X	X	X

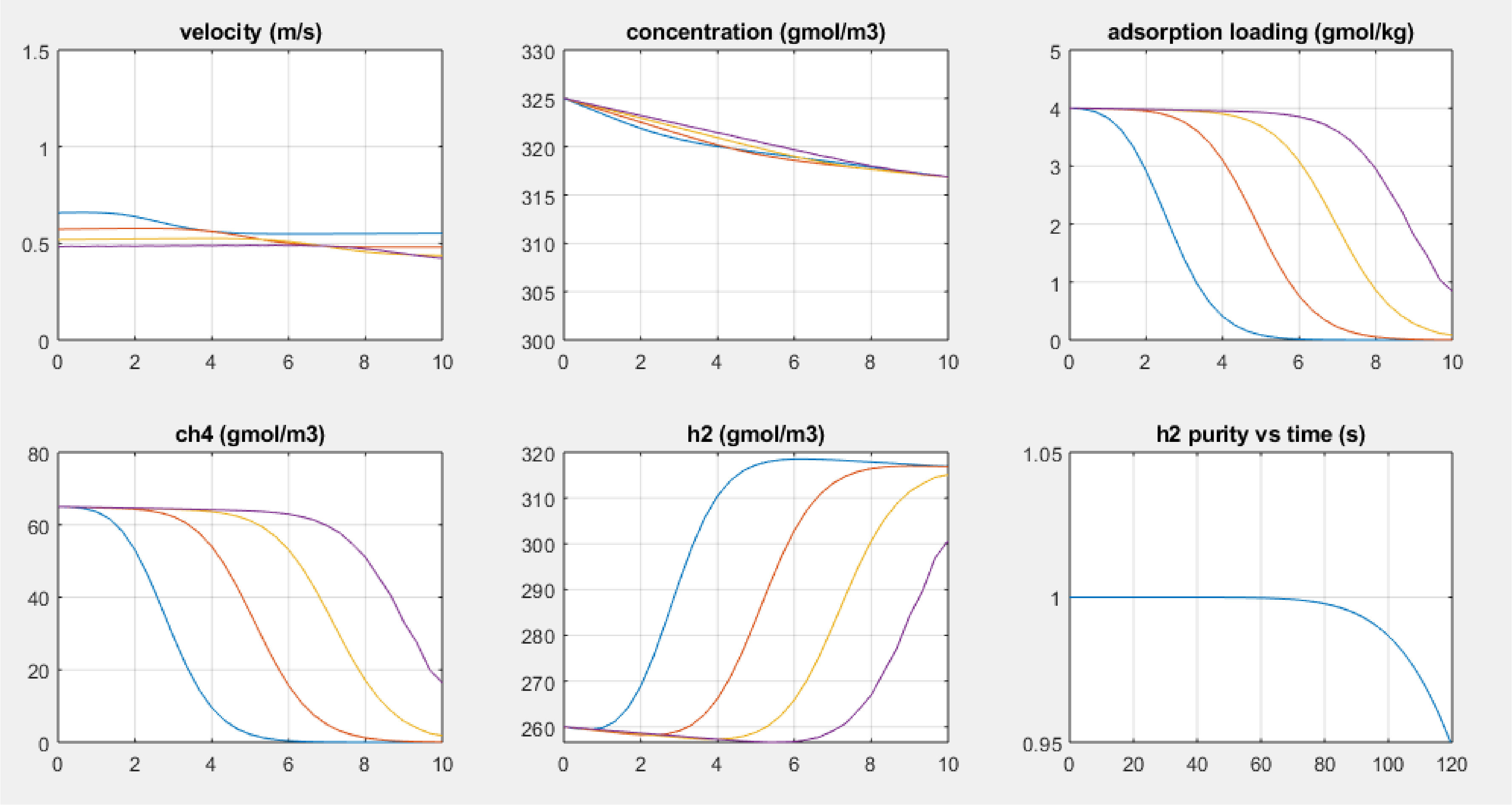
Case Study – Initial Flowsheet

A model of a single bed can be used to represent the whole system.



Custom Model: Adsorbent Bed

An isothermal adsorption model for a binary mixture was written in MATLAB.



For our proof of concept work, we created a CSTR simplification from this high-fidelity model.

MATLAB Model Function

Rendering the model as a function allows variable passing capability to be exploited.

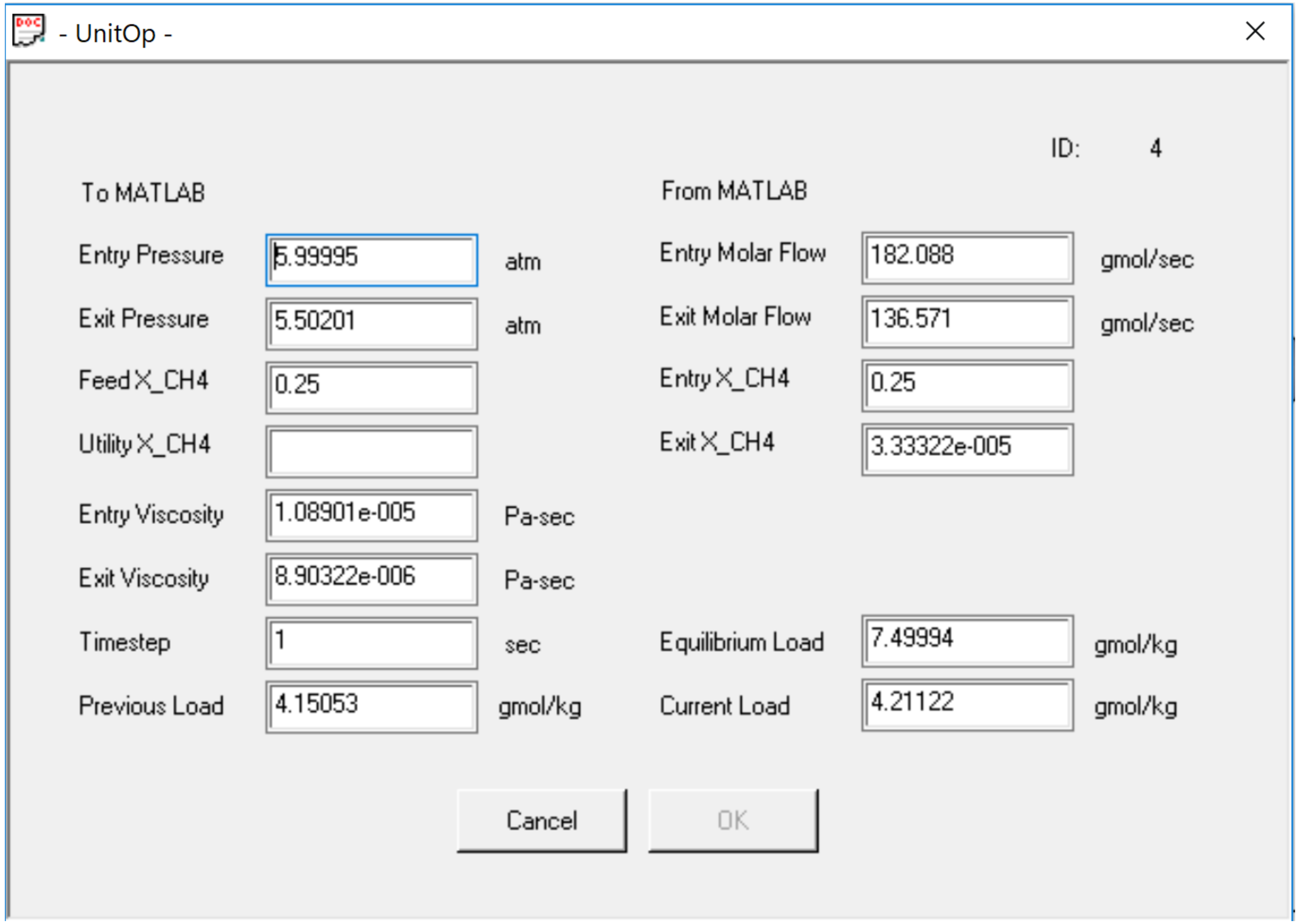

```
% Pressure Swing Adsorption for binary separation -- BATCH model

% INPUTS:
% model_in(1) = BC mode (1 = p_entry, p_exit), 2 = ndot_entry, p_exit)
% model_in(2) = bed entry pressure (Pa), or molar flowrate (gmol/s)
% model_in(3) = bed exit pressure (Pa)
% model_in(4) = bed entry mole frac adsorbate
% model_in(5) = bed entry viscosity of mixture (Pa-s)
% model_in(6) = timestep (seconds)
% model_in(7) = bed adsorbent loading state (gmol/kg) -- PERSISTENT

% NOTE >>> Flow direction is assumed to be from bed entry to exit.

% OUTPUTS:
% model_out(1) = error status code
% model_out(2) = bed entry molar flowrate (gmol/s), or pressure(Pa)
% model_out(3) = bed exit molar flowrate (gmol/s)
% model_out(4) = bed exit mole frac adsorbate
% model_out(5) = equilibrium adsorbent loading (gmol/kg)
% model_out(6) = bed adsorbent loading state (gmol/kg) -- PERSISTENT
```

Value of UnitOp “state” (adsorbent loading) must be preserved between function calls.

VBA Unit Op: Adsorption Bed

The CHEMCAD custom unit operation block was configured according to the MATLAB model.

- Build CHEMCAD flowsheet
- Write VBA routine to connect with MATLAB
- Customize VBA UnitOp parameters & dialog

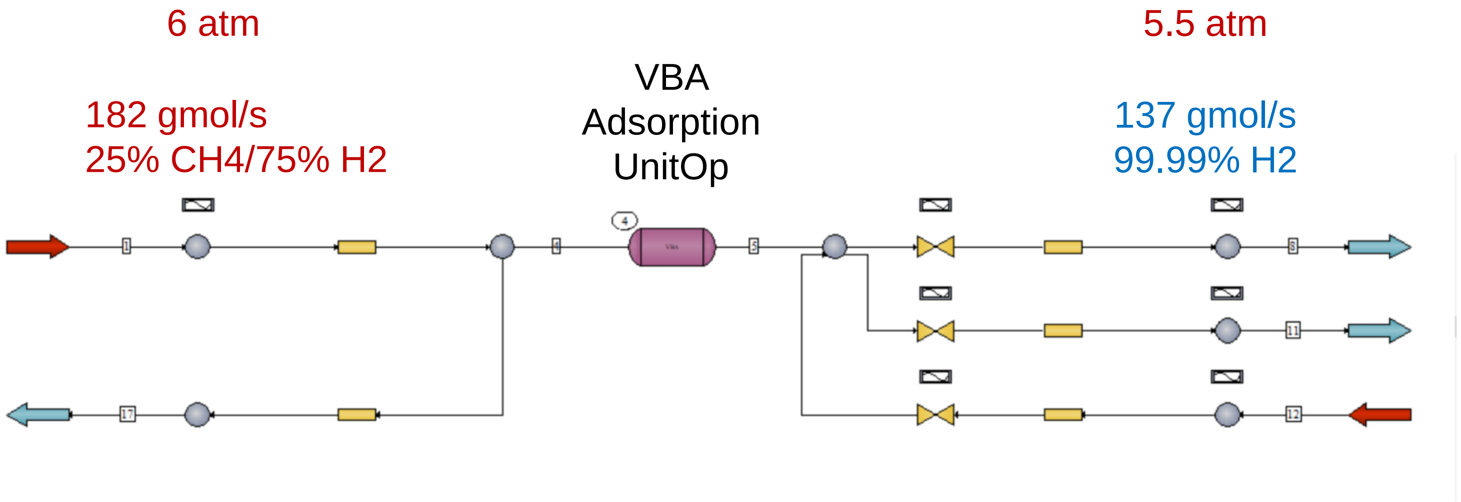


When running simulation, MATLAB opens in background. CHEMCAD records stream & UnitOp data for reporting results.

Steady State Simulation

To illustrate connectivity between the platforms, we first ran a steady-state case.

- Inlet BC = pressure, feed composition
- Outlet BC = pressure
- CHEMCAD correctly displays the exit flow and composition calculated by MATLAB.



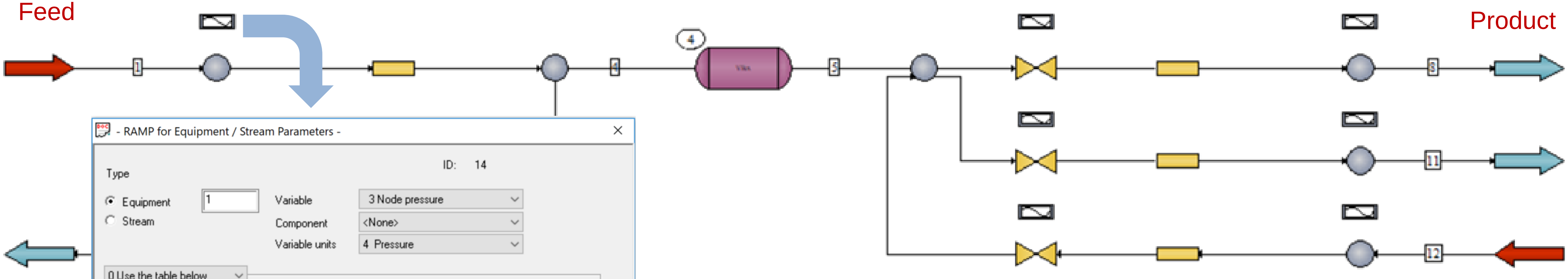
What about a dynamic simulation?

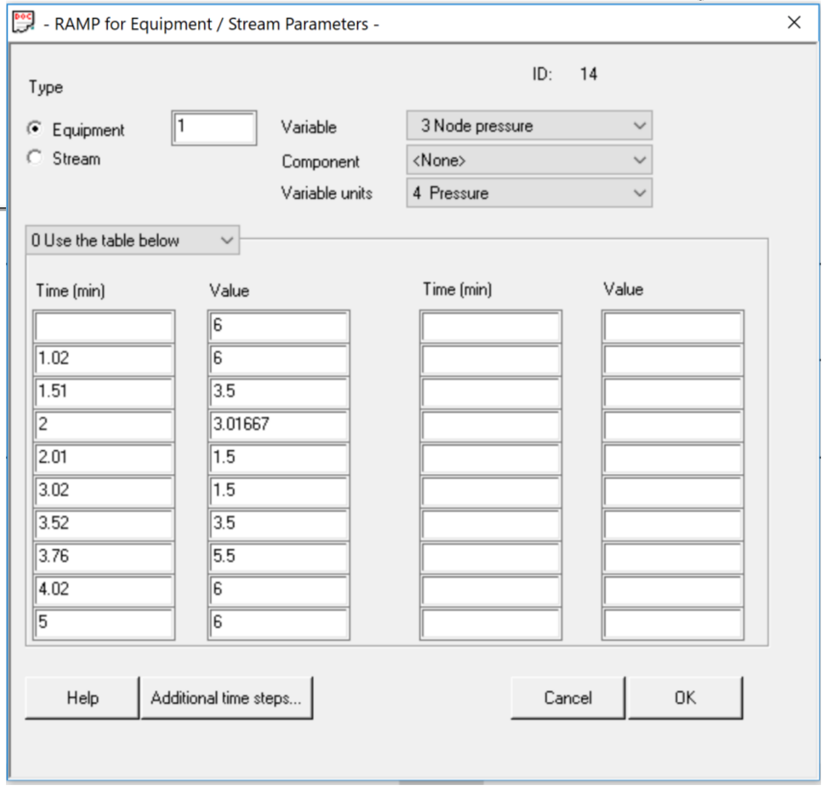
Dynamic Simulation – Schedules

The first dynamic run involved prescription of a time-dependent boundary condition schedule, to ensure that CHEMCAD and MATLAB results agreed over a multiple of time values.

Feed

Product





- RAMP for Equipment / Stream Parameters -

Type ID: 14

- (•) Equipment: 1
- ( ) Stream

Variable: 3 Node pressure

Component: <None>

Variable units: 4 Pressure

0 Use the table below

Time (min)	Value	Time (min)	Value
	6		
1.02	6		
1.51	3.5		
2	3.01667		
2.01	1.5		
3.02	1.5		
3.52	3.5		
3.76	5.5		
4.02	6		
5	6		

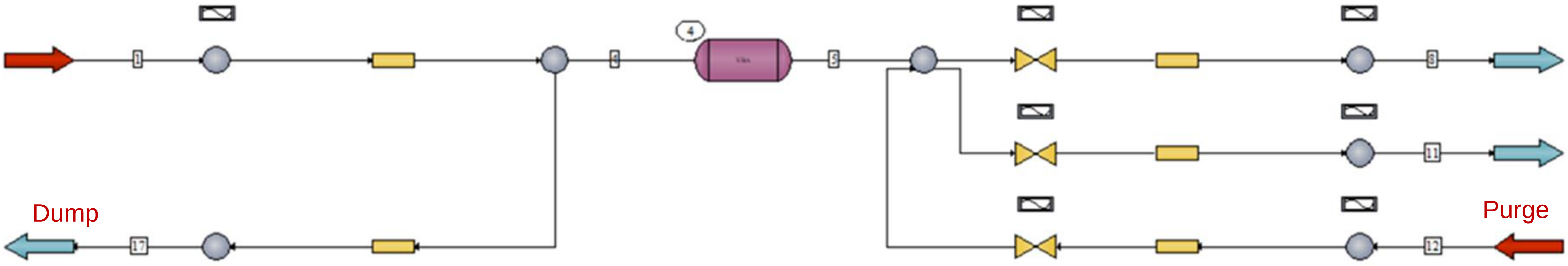
Help | Additional time steps... | Cancel | OK

RAMP controlling pressures

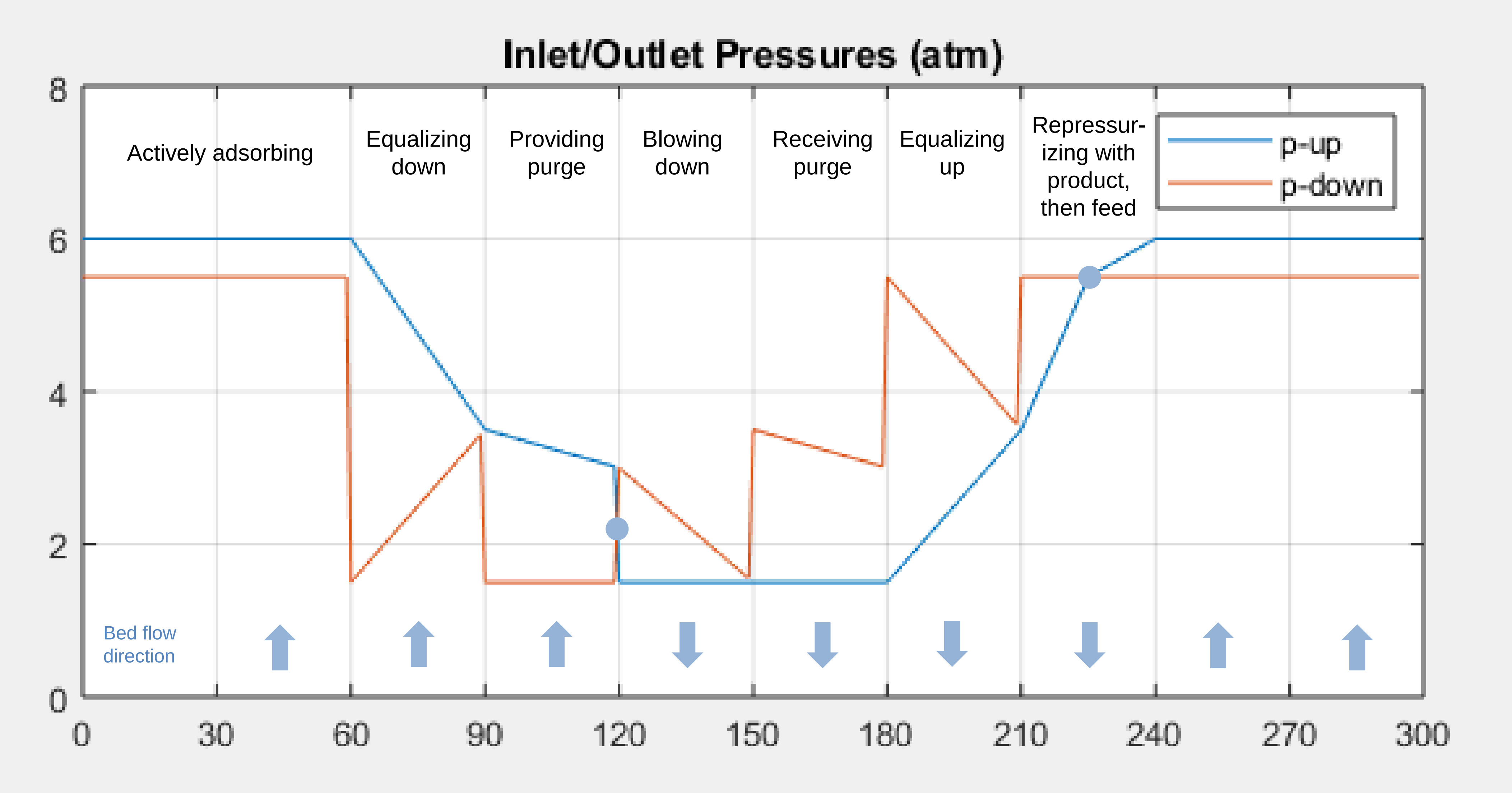
Now what about reverse flow - the PSA purge cycle?

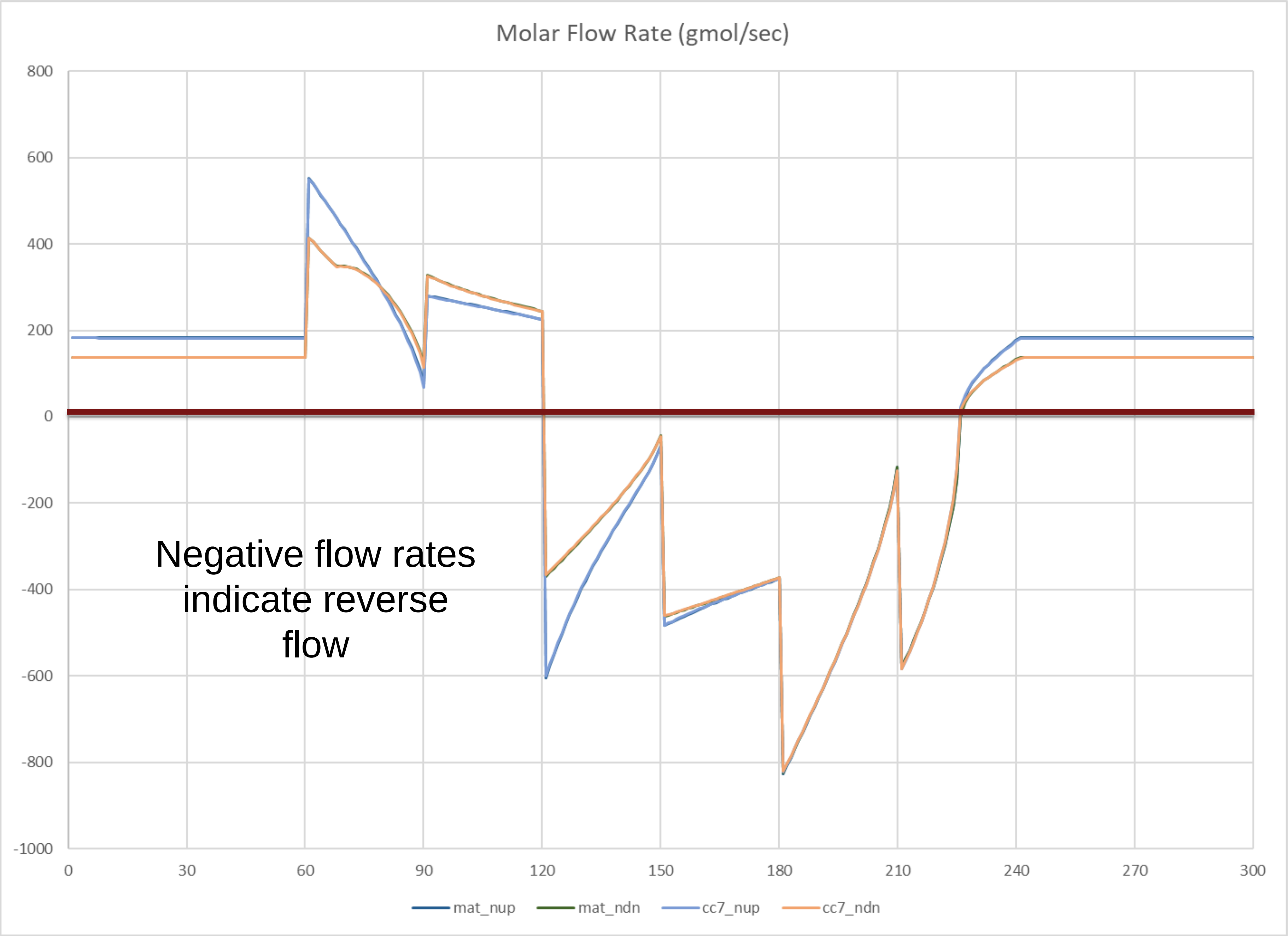
Dynamic Simulation – Reverse Flow

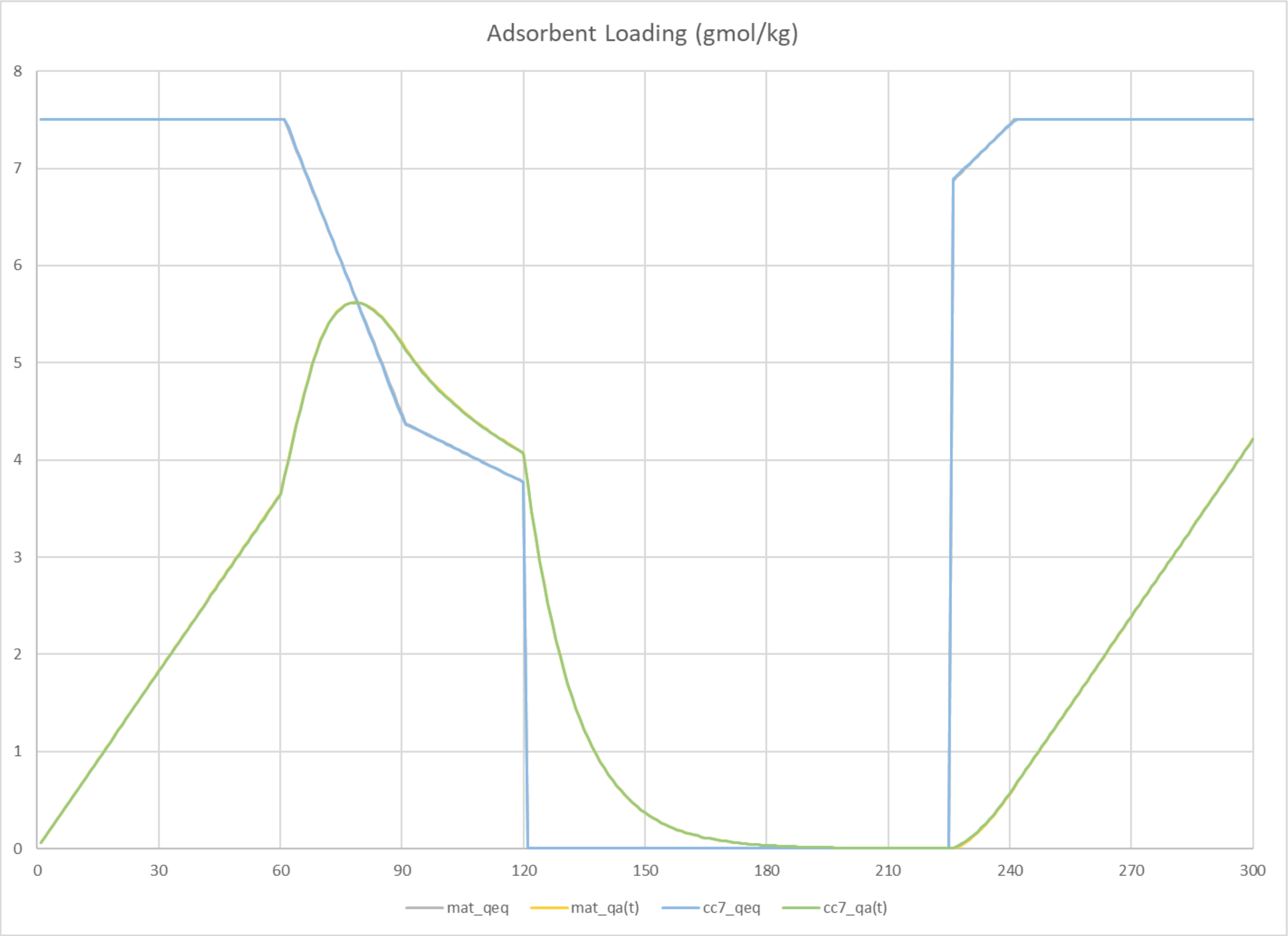
The PSA cycle requires both forward (adsorption) and reverse (desorption) flow through the bed. Accordingly we studied a reverse flow scenario akin to the purge stage.



Pressure Schedule



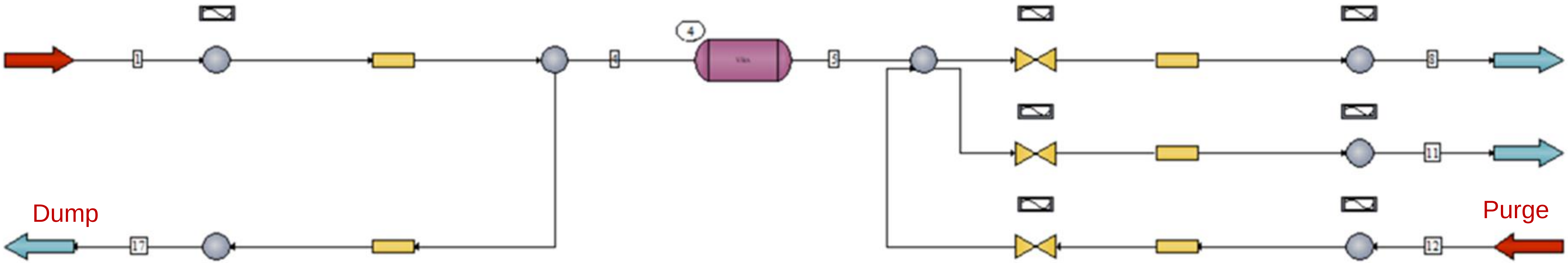

Molar Flow Rate (gmol/sec)
800
600
400
200
0
-200
-400
-600
-800
-1000
0
30
60
90
120
150
180
210
240
270
300
mat_nup
mat_ndn
cc7_nup
cc7_ndn
Negative flow rates indicate reverse flow



Adsorbent Loading (gmol/kg)
8
7
6
5
4
3
2
1
0
0
30
60
90
120
150
180
210
240
270
300
mat_qeq
mat_qa(t)
cc7_qeq
cc7_qa(t)

Dynamic Simulation – Reverse Flow

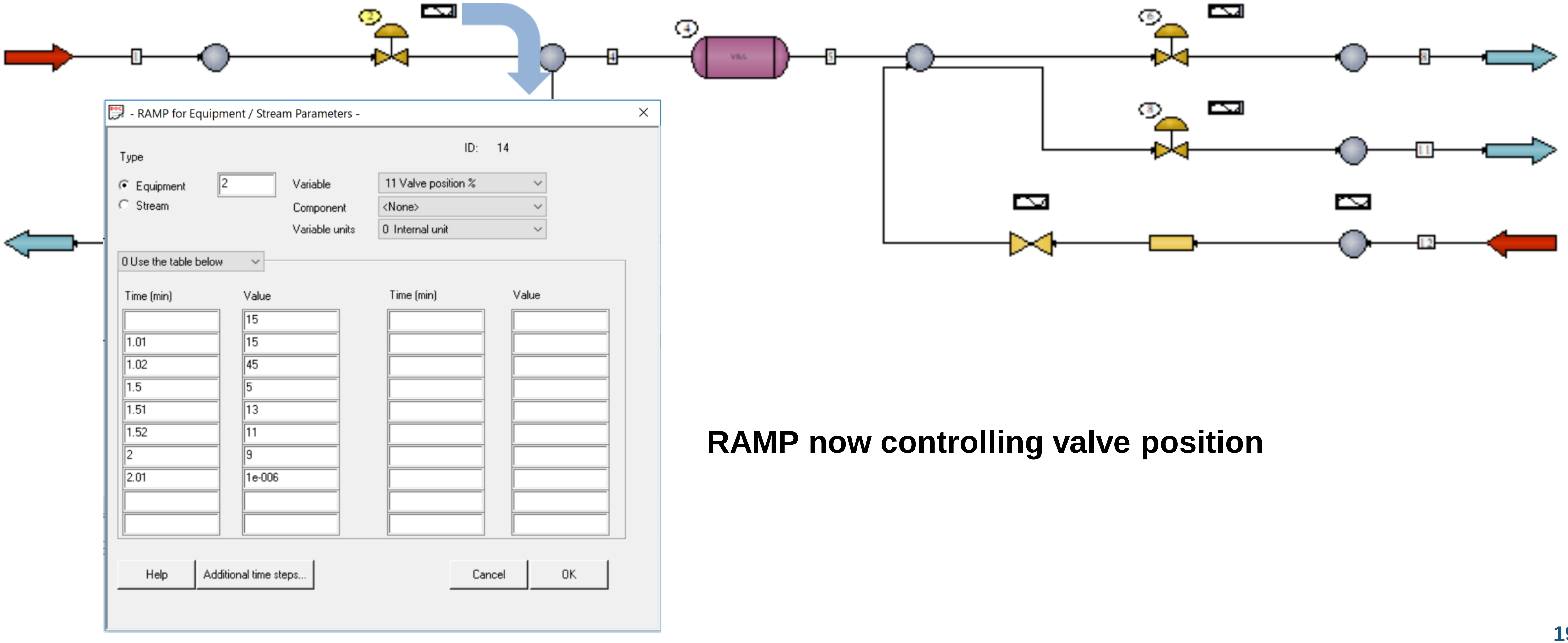
The PSA cycle requires both forward (adsorption) and reverse (desorption) flow through the bed. Accordingly we studied a reverse flow scenario akin to the purge stage.



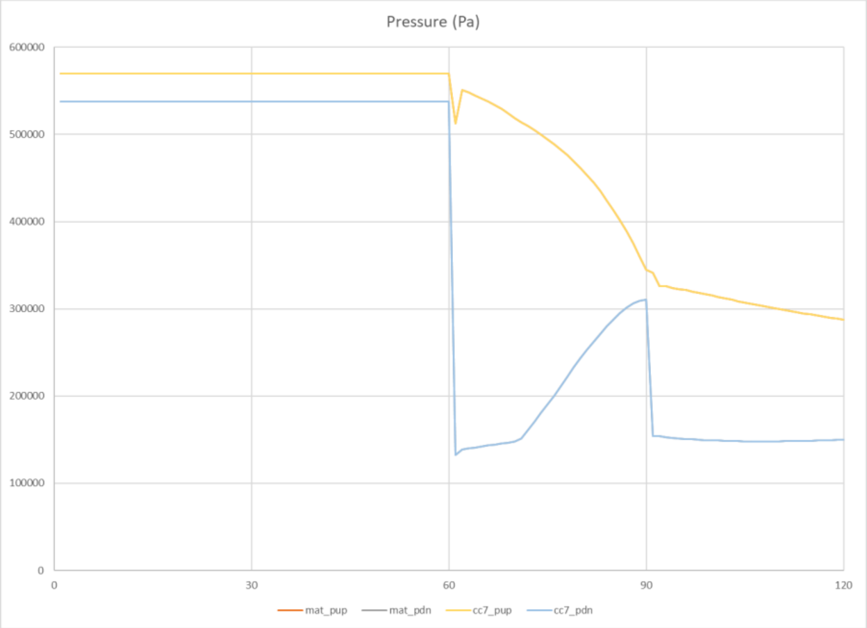
The basic dynamics of a PSA cycle have now been demonstrated, using a hybrid simulation.

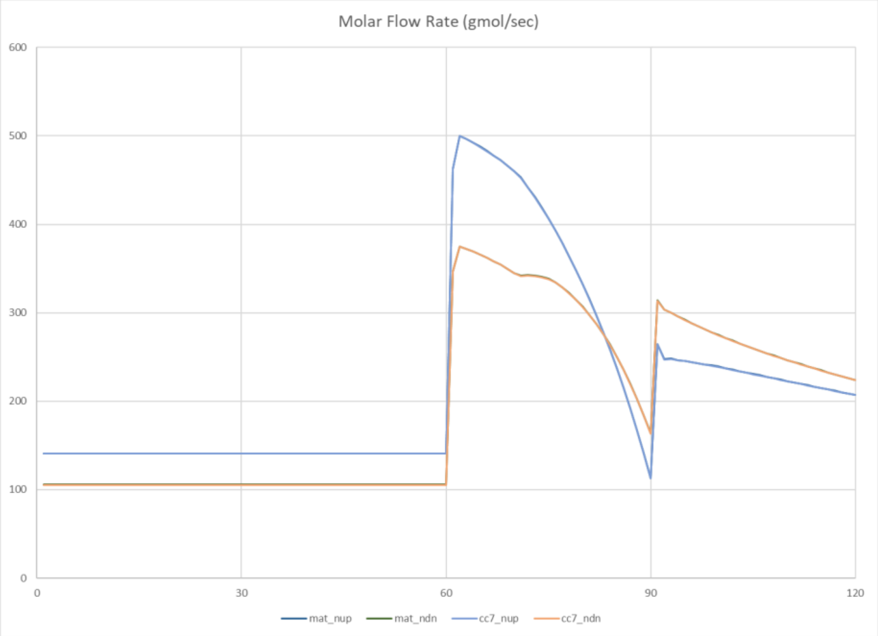
Dynamic Simulation – Valve Dynamics

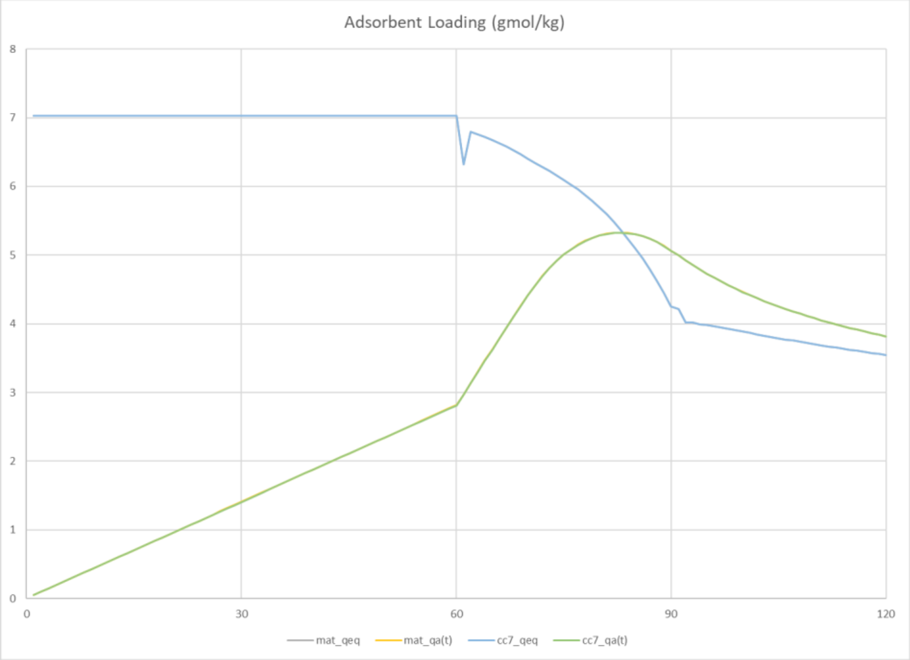
Control valves upstream and downstream of the bed were then scheduled to control the flow dynamics for forward flow.



RAMP now controlling valve position







See effect of valve action in plots

Curves closer to reality

Summary

Fully-featured process simulation

+

Broad-based technical computing

=

Possible future work:

- High-fidelity model
- Energy balance
- Blowdown & pressurization w/control valves
- PSA cycle scheduling w/multiple beds

Acknowledgments

Chemstations:

- Carla Lara
- David Hill
- Aaron Herrick

Xebec Adsorption:

- Nima Mohammadi