

**If at first you don't succeed...**

## **Tips for more effective problem solving with *ChemSep***

**Ross Taylor and Harry Kooijman**

**August 20, 2013**

**Updated for Version 8.11: January 3, 2019**

The models in *ChemSep* include material balance equations, energy balances, phase equilibrium equations and, in the case of the nonequilibrium model, mass and energy transfer rate equations. The model equations are non-linear and so must be solved by an iterative method.

Iterative methods require an initial estimate of all of the unknown variables. These are the flow rates, composition, and temperatures of all of the interstage and product streams. An algorithm then tries to “improve” these guesses until all of the equations are satisfied. Some simulation programs ask their users to provide initial estimates of some important variables (e.g. top and bottom stage temperatures, bottom and top stage flows), and, while that is possible with *ChemSep*, most users probably don't even know that they can provide initial estimates of anything and simply allow the program to generate its own estimates of all of the unknown variables.

As is the case with most iterative methods, convergence is not assured using this method and whether or not a solution is found depends largely on the quality of the initial guess of the unknown variables. Indeed, we sometimes say that while *ChemSep*'s algorithms will immediately solve 90% of your simulations, it is the other 10% that demand 90% of your time to find a solution.

### ***Methods in ChemSep***

*ChemSep* now includes several different approaches to solving the model equations, all based on Newton's method, but now automatically incorporating various “tricks” that we have found helpful in our own use of the program.

ChemSep includes the following methods for solving the model equations:

1. Newton's method
2. 2-pass constant H first
3. 2-pass ideal H first
4. 2-pass ideal K + H first

5. 2-pass ideal K + constant H first
6. 2-pass ideal K + constant H,k first
7. 2-pass constant k first
8. Equilibrium stage model first

These methods are discussed briefly below and shown in the partial screen shot (from **Solve Option**) shown below.

Solve options	
Numerics Options	
Initialization	Automatic
Method	Newton's method
Accuracy	Newton's method
Number of Iterations	2-pass constant H first
Trace threshold	2-pass ideal H first
	2-pass ideal K + H first
	2-pass ideal K + constant H first
	2-pass ideal K, constant H, k first
	2-pass constant k first
	Equilibrium model first
Newton step limits:	
Flow	0.200000 (-)
Temperature	10.0000 (K)
Composition	0.0500000 (-)
Flux	1.00000 (-)

#### *Method 1: Newton's method*

Newton's method is the default method and will be used unless an alternative is selected from the drop-down list shown above.

Experienced users of simulation programs will know that problems involving mixtures that are thermodynamically ideal usually are very easy to solve. Thus, methods 2 to 6 in the above list exploit this fact in different ways.

#### *Method 2: 2-pass constant H first*

Distillation is a process in which one often assumes that the molar flows are constant from stage to stage. The practical consequence of this assumption is that the energy balances for each stage can be ignored. The calculations are much simpler and convergence often is very rapid.

Thus, in this option *ChemSep* first attempts to solve a problem with constant enthalpies (for each phase). This is done by resetting the enthalpy model to **None**; once that problem has converged the program automatically resets the enthalpy model to be whatever was initially selected and the simulation re-run using the solution to the constant H-model as the starting point.

#### *Method 3: 2-pass ideal-H first*

The approach taken here is very similar to that of method 2. Rather than select **None** as the enthalpy model, in this version of our algorithm, the enthalpy model used in the first pass is **Ideal**.

#### *Method 4: 2-pass ideal K+H first*

In this version of the 2-pass Newton method, the first problem is solved using ideal enthalpies *and* ideal K-values (in practice this means the Raoult's law-like **Wilson K-value model**).

#### *Method 5: 2-pass ideal K+ constant H first*

In this version of the 2-pass Newton method, the first problem is solved using constant enthalpies *and* ideal K-values (in practice this means the Raoult's law-like **Wilson K-value model**).

#### *Method 6: 2-pass ideal K+ constant H,k first*

In this version of the 2-pass Newton method, the first problem is solved using constant enthalpies *and* ideal K-values *and* constant mass transfer coefficients.

#### *Method 7: 2-pass constant k first*

In this version of the 2-pass Newton method, the first problem is solved using constant mass transfer coefficients. The selected K-value and enthalpy models are used from the start.

#### *Method 8: Equilibrium stage model first*

This method is designed for rate-based problems but uses an equilibrium stage model in the first pass.

### **Newton Step Limits**

The upper right hand side of the **Solve Options** panel shows the step limits that our program applies in order to stop the variables from changing too much between iterations.

The **Flow** limit represents the maximum allowed percentage change in the vapor and liquid flow rates between iterations. We prefer to keep this to 0.5 or less to prevent overly large changes to the flow rates between iterations. A smaller number can be useful in nonequilibrium simulations because the column flow dynamics calculations are somewhat sensitive to the actual flow rates in the column (this is not an issue in equilibrium stage calculations because there are no equipment calculations carried out during the simulation and so it is only the relative flows that matter in such cases).

The **Temperature** limit is the number of degrees by which the temperature is allowed to change between iterations. The default setting is 10K, but sometimes we will set this lower (it is most likely that we would do this in column simulations at “high” pressure).

The **Mole fraction** limit is the maximum amount by which mole fractions are allowed to change between iterations. Mole fractions are not allowed to become negative or take on values that are greater than one, however. *ChemSep* uses a strategy whereby any step large enough to make a mole fraction negative (or greater than one) is automatically reduced so as to take half the step from the current value to zero (or one if the calculated step would take the mole fraction out of range in the other direction, so to speak).

*ChemSep* users might observe that there is a new option in this section; the 4<sup>th</sup> line shows **Flux** which allows us to limit changes to the mass transfer rates between iterations. This option is only used in the nonequilibrium model, of course, but our experience is that it can make all the difference between success and failure. In particular, we find that setting the flux limit becomes more and more important as the column pressure increases. At very high pressure (relative to the critical point of the mixture) we find that success is more likely when the flux limit is set to 0.1 (the default is 1.0 and that value is usually sufficient at low and moderate pressures).

## Recommendations

Newton's method should be the first choice (it is the default setting) until it has been shown not to work. In the event of failure then one of the alternative options can be tried (but not all of these methods will allow every problem to be solved).

Method 2 can be useful for hard-to-converge distillation-type models. It is less useful for hard-to-converge reboiled absorbers and strippers. One of the alternatives might work better for those applications. Our experience is that Method 4 is perhaps the best for these applications. Methods 2-4 should not be used for liquid-liquid extraction operations where the enthalpy model plays little or no role and where the concept of ideal K-values would mean that two liquids phases would not be able to co-exist.

Methods 5-8 should not be selected for equilibrium stage simulations (because the model does not include any mass transfer coefficients). (An inadvertent selection of one of those methods need not be a cause for alarm, the simulation program will automatically switch to the nearest equivalent method that is appropriate for an equilibrium stage model.)

## Difficult Cases

Here we discuss what makes simulations hard to converge and suggests some strategies to help find solutions to these “difficult” cases.

Computer-based methods for solving distillation (and related) column simulation problems now are reasonably reliable. Nevertheless, there remain times that such methods fail to converge. Below we discuss some of the reasons that a simulation might be difficult to converge, along with suggestions on what might make the problem more amenable to solution. The key idea is to modify the problem that is difficult to solve as posed into one that is easier to solve. It is important to note that most simulators allow a calculation to be restarted from an older converged solution. The solution to the “easy” problem may then be used as a starting point for the more difficult one whose solution is desired.

So what circumstances lead to convergence difficulties? Here are some possibilities:

1. “Difficult” or non-standard specifications
2. An inappropriate numbers of stages (too few or far too many)
3. Highly nonideal solutions
4. Large heat effects

5. High pressures
6. Nonequilibrium (rate-based) models

We discuss these circumstances in more detail below.

### *Non-standard Specifications*

Non-standard specifications are very likely to be source of convergence difficulties. It is all too easy to specify a desired product purity or component flow rate that simply cannot be attained with the specified column configuration. There is always (at least one) solution to distillation simulations if feasible values of the reflux ratio and bottoms flow rate are specified (the so-called standard specifications), and to which it is likely to be reasonably easy to converge. Other specifications that, in effect, determine the internal flow rates also tend to lead to simulations that are easy to converge: reflux and reboil ratios, reflux and boilup flow rates, and product flow rates fall into this category. Other specifications that can cause difficulties include specifying temperatures and compositions anywhere in the column, and specifying condenser and/or reboiler heat duties.

A way to circumvent the difficulties that these non-standard specifications might pose is first to obtain a converged solution for a case involving standard specifications. Once the behavior of the column is understood it will be possible to make sensible non-standard specifications, again using an old converged result as a starting point. **Our recommended practice is to carry out the initial studies using standard specifications until one has gained an understanding of column behavior** and, preferably, have found a set of standard specifications that gives us a solution close to that desired. Then, and only then, will we consider specifying product (im)purities, or recoveries, or temperatures, or other non-standard specifications. When switching to use non-standard specifications we would use the previously converged solution as a starting point.

### *Too Few or Too Many Stages*

We add the caveat that specifying product (im)purities is a more reasonable option for mixtures that contain only two components (or have more components but all but two of them are “small”) **and** a number of stages sufficient to the task at hand has been specified (it is all too easy to create a simulation that can never converge simply because too few stages were specified).

Very large numbers of stages can pose their own kind of convergence difficulty. A possible remedy is to reduce the number of stages until a converged solution can be obtained. This solution can then be used as the starting point for a problem with more stages. Interpolation will have to be used to estimate values of the flows, temperatures, and mole fractions for any added stages, something not available in all programs (but is in *ChemSep*).

## *Thermodynamics*

As a rule, we can expect the degree of difficulty to increase with increasing liquid phase nonideality. Simultaneous convergence methods are often recommended for simulating strongly nonideal systems (as opposed to tearing or inside-out methods), but even SC methods can experience difficulties with strongly nonideal systems. Such systems often are encountered in azeotropic and extractive distillation. It is likely that an activity coefficient model is part of the model used to describe the thermodynamics of these systems, and the source of the convergence difficulties often encountered with such systems. A possible remedy is to make the system “less” nonideal. By first solving an equivalent ideal system that omits entirely the activity coefficient model (i.e. using Raoult’s law) may provide a converged solution that may be an adequate starting point for the nonideal system of interest. However, since many simulators use ideal solution thermodynamic models in any self initialization method, this technique may not of sufficient help.

A measure of the nonideality of the system is given by the magnitude of the interaction parameters for the activity coefficient model. It is possible, therefore, to lessen the degree of nonideality by reducing the interaction parameters sufficiently to make the problem easy to solve. The parameters may then be increased in size in a series of steps until the desired values are reached, each time using the solution converged using the previous set of parameter values as the starting point. It is essential that the parameters return to their correct value in the final step because intermediate solutions have no meaning, serving merely as an aid to convergence. Using the stage efficiency as a continuation parameter also is useful for such cases, provided that the simulation employs standard specifications (more on this topic below).

The most strongly nonideal systems are those that may exhibit two liquid phases. We avoid further discussion of such systems because special algorithms are needed for these cases; see, for example, Chapter 8 of Doherty and Malone (*Conceptual Design of Distillation Systems*, McGraw-Hill) for entry points to the literature.

Large heat effects can lead to convergence difficulties. For such systems it is the enthalpies that are the source of the nonlinearity that leads to convergence failures. It is generally not straightforward to modify enthalpies in a simulator because there are no adjustable parameters that exert their influence over the enthalpy in a way comparable to that of the interaction parameters in the activity coefficient model. Use of a constant enthalpy model in distillation calculations, if available, will lead to constant molar flows from stage to stage (within each separate section of the column), a condition often approached in many real distillation (but not absorption) columns. Thus, if the simulator includes a constant enthalpy model then this can be used to obtain a converged solution that may provide a good starting point for the problem with a more realistic enthalpy model. Alternatively, it can help to use an ideal enthalpy model for an initial solution. This alternative is likely to be more useful than the constant enthalpy model for cases where there are significant changes in the flows from stage to stage (reboiled absorbers and strippers and some high pressure distillations come to mind).

## *High pressure*

High pressure can add considerably to the difficulty of converging simulation models. It is likely that a cubic equation of state will be used to estimate fugacity coefficients and enthalpy departures in such systems. Mixtures become increasingly nonideal as the pressure is raised.

In some cases the column may operate “close” to the pseudo-critical point at which the densities of both phases approach each other. In other cases the iterations may take the estimates of temperature and composition into regions where the equation of state can provide only one mathematically real root for density or compressibility. Occurrences of this behavior often are a source of convergence difficulties. One possible strategy to alleviate such difficulties is to reduce the pressure until the problem becomes easy to solve. A converged solution obtained in this way may be used as the starting point for subsequent calculations at increasingly higher pressures (up to that desired).

Columns in which temperature and/or compositions change over a wide range in a limited number of stages pose their own particular difficulties. Some highly nonideal systems exhibit this kind of behavior. For cases such as this it is wise to limit per iteration changes to temperature and composition. Most modern computer methods will allow users to set such limits (and *ChemSep* is not an exception).

### *Rate-Based Models*

Nonequilibrium (or rate-based) models can be more difficult to converge than an equivalent equilibrium stage calculation because the former needs mass transfer coefficients and interfacial areas, parameters that depend on the local state variables, (temperature, pressure, and composition) as well as on flow rates, and equipment dimensions (weir height, for example). **If a nonequilibrium model fails to converge the very first question we would ask is: Have you got the corresponding equilibrium stage model to converge?** (By this we mean a simulation that involves the same components, column configuration, and specifications – in so far as far as that is possible). If the equilibrium stage model is hard to converge, it is likely that the nonequilibrium model will be at least as hard to converge (and probably harder for the reasons just given). In any event, a converged equilibrium stage solution often is an excellent starting point for converging a nonequilibrium model. **Our recommended practice is to always first perform an equilibrium stage before switching to a nonequilibrium simulation.**

### *Discussion*

This is by no means an exhaustive list of the reasons that computer-based simulations fail. Indeed, in many cases it is a combination of more than one of the above factors that leads to difficulty. In those cases we may need to combine several of the strategies outlined above in order to solve the simulation problem. Often, however, there is no substitute trial and error. J. Haas (Chapter 4 in Kister, *Distillation Design*, McGraw-Hill, 1992) offers some additional insight on using simulators to solve distillation column models.

## New Convergence Aids in ChemSep 8.11

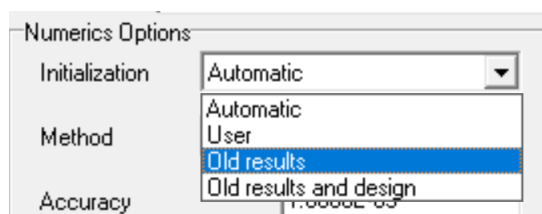
Version 8.11 introduces some new aids to converging what might otherwise be very difficult simulations. These include:

1. A more accurate set of “old results.”
2. An alternative method of handling non-standard specifications
3. Continuation on some non-standard specifications

These topics are addressed in what follows.

### Reusing Old Results

In many cases it is convenient to start a new simulation that is not too different from one already converged using the results from that original simulation. ChemSep will not automatically use old results as the starting point. To ensure that old results are used you need to select them from the drop down list on the *Solve Options Panel*



Convergence from old results can be very fast, especially if the new specifications are not too different from those of the original simulation. However, there are circumstances where the old results lack sufficient precision due to round off error caused by repeated reading and writing of the results. Here is what happened when old results were used for a repeat simulation of a high pressure separation.

```
Run level: Complete model
Solution vector not found

  Iteration  log(Err/Tol)
      0      4.1950 T80= 258.1458
Warning! Feed pressure < column pressure (Feed 2)
      1      4.0346 T80= 256.9878
      2      2.8144 T80= 258.1379
      3      2.3544 T80= 258.1968
      4      2.0972 T80= 258.1537
      5      1.7824 T80= 258.1449
      6      -0.4800 T80= 258.1440

Run level: Report
Convergence obtained in 6 iterations
Time      2413 milliseconds
FixMem driver done
Process ended
```

You can see it takes several iterations to converge to the result that was already available (no specification was changed for this illustration). (Not all cases need this many iterations when restarted from *Old Results*. Most, in fact, will converge in 2 or 3 iterations, but some cases are



especially difficult and these cases may need more iterations even from old results.)

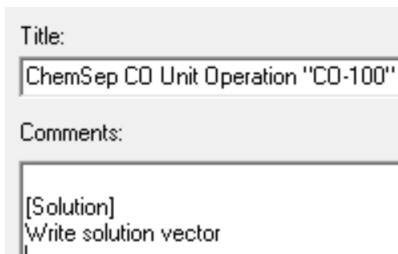
The reason ChemSep needs additional iterations is because of round-off error. Put simply: the solution that is written back to the sep file prior to solution has lost some significant digits so it is less accurate.

Starting with Version 8.11 it is possible to save and then re-use a more accurate version of the existing simulation results. To use the feature type the following into the *Comments* box on the *Title* panel:

```
[Solution]
Write solution vector
```

(You do not have to start typing at the beginning of the line, and you can type other characters on the same lines, but the key words shown above must appear *exactly* as shown and they must be on consecutive lines. If multiple blocks that start with a key word in [ ] are to be used in the same simulation then they should be separated by an empty line.)

Here is an example to show what happens when the solution vector was saved and then was used as the starting point for a repeated simulation. First, an image of the *Comments* box:



The following image shows the iteration history:

```
Iteration  log(Err/Tol)
Warning! Feed pressure < column pressure (Feed 2)
0          1.2937 T80=  258.1476
1          -0.4723 T80=  258.1458
Run level: Report
Convergence obtained in 1 iterations
```

And we see that convergence is extremely fast (as we would, of course, expect). Note that it remains necessary to select *Old Results* from the *Initialization* menu as already discussed.

The disadvantage of writing the solution vector is that it can add considerably to the size of the sep file that records all of the specifications and results.

A second disadvantage is that old solution vectors might lack relevance if used to start a simulation in which the column configuration has changed (the number of stages is different, or the feed stage has moved). The method can still be used but the old results may no longer be sufficiently accurate.

Writing the solution vector should be particularly useful when ChemSep is employed in a

flowsheet simulation program like Aspen HYSYS or UNISIM Design where repeated simulations are the rule, rather than the exception.

## Handling Nonstandard Specifications

We discussed above how use of non-standard specifications can make convergence less likely for some simulations. We also pointed out that for feasible standard specifications a solution always exists (that does not mean the simulator can find it). (We emphasize that the specifications must be feasible because there will not be a solution in cases where the standard specifications do not lead to a feasible design. It is not always easy to know in advance of a simulation if a given set of specifications is feasible.)

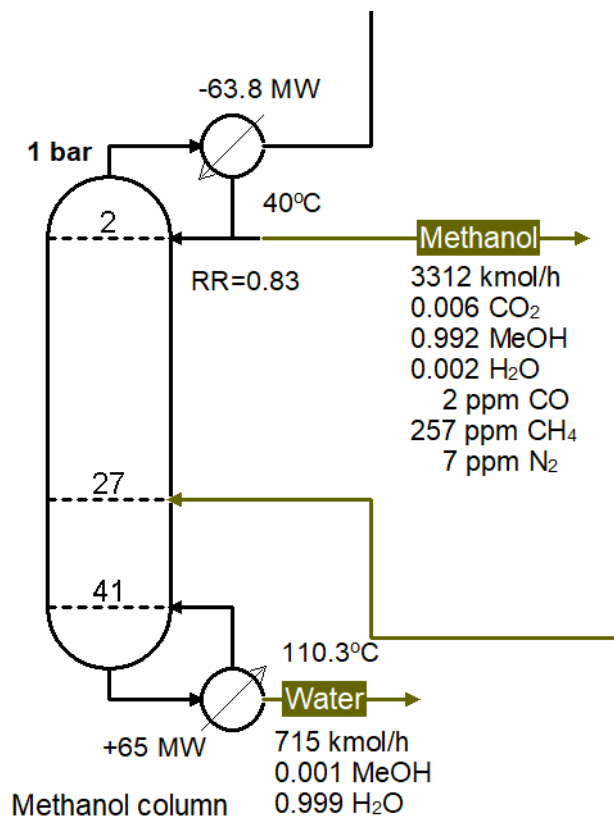
By default, ChemSep uses the actual specifications (which may include product mole fractions or recoveries or stage temperatures or heat duties) following the initialization. This contributes to the degree of difficulty of the simulation. However, it is often the case that a simulation using standard specifications (reflux and reboil ratios, product flow rates, reflux and boil-up flow rates) will converge more easily.

ChemSep 8.11 introduces a mechanism to allow some standard specifications to be used throughout the first (and/or second) passes with Newton's method and then to add an additional pass with Newton's method that uses the actual specifications. To use the feature type the following into the *Comments* box on the *Title* panel:

```
[Algorithm]
Use Fixed Flow Specs
```

(You do not have to start typing at the beginning of the line, and you can type other characters on the same lines, but the key words shown above must appear *exactly* as shown and they must be on consecutive lines. If multiple blocks that start with a key word in [ ] are to be used in the same simulation then they should be separated by an empty line.)

To illustrate the application of this approach consider the following column from a methanol plant design. (Our process simulation can be found here: <http://chemsep.com/downloads/index.html>. The inspiration for this flowsheet comes from a paper by W.L. Luyben <https://pubs.acs.org/doi/abs/10.1021/ie100323d?journalCode=iecred>.)



This column actually has a two-product condenser with one product stream being a sidestream (the vapor recycle stream in this case). The desired specifications are two product mole fractions:

Column Product Specifications			
Top product name	<input type="text" value="Top"/>	Condenser duty name	<input type="text" value="Qcondenser"/>
Top specification	Mole fraction of a component	=	<input type="text" value="0.992000"/> (-)
	<input type="text" value="Methanol"/>		
Bottom product name	<input type="text" value="Bottom"/>	Reboiler duty name	<input type="text" value="Qreboiler"/>
Bottom specification	Mole fraction of a component	=	<input type="text" value="1.0000E-04"/> (-)
	<input type="text" value="Methanol"/>		

Despite the fact that column specifications include two product mole fractions, this is not a particularly hard simulation to converge:

Run level: Complete model

Iteration	log(Err/Tol)	RR=	BR=	RR=	BR=
0	5.8475	RR=	2.0000	BR=	2.0000
1	7.5203	RR=	1.3333	BR=	2.0000
2	6.7014	RR=	0.8889	BR=	3.0000
3	6.4529	RR=	0.8889	BR=	4.5000
4	6.1935	RR=	0.8889	BR=	6.7500
5	5.9992	RR=	0.9603	BR=	9.2163
6	5.8993	RR=	1.0201	BR=	10.2082
...	...	...	...	...	...
26	5.4277	RR=	0.8951	BR=	8.1580
27	5.2097	RR=	0.8951	BR=	8.1580
28	4.9202	RR=	0.8951	BR=	8.1580
29	1.1728	RR=	0.8951	BR=	8.1580
30	-2.0965	RR=	0.8951	BR=	8.1580

Run level: Report  
Convergence obtained in 30 iterations

We see that ChemSep needed 30 iterations for this example. Notice that the reflux ratio (RR) and boilup ratio (BR) are displayed on each iteration. This is not done by default, variables to monitor during the iterations can be selected from the specifications panel as will be shown below. We can also see that the initial estimates of both of these ratios is 2. These are the default values used by ChemSep. Often we can do better and below we will show how.

If we force ChemSep to use fixed flow specs in the first pass with Newton's method:

Title:

ChemSep CO Unit Operation "Column" in COFE Flowsheet

Comments:

[Algorithm]  
Use Fixed Flow Specs

The iteration history now becomes:

Run level: Complete model: Using fixed flows

Iteration	log(Err/Tol)	RR=	BR=	RR=	BR=
0	4.5995	RR=	2.0000	BR=	2.0000
1	2.5221	RR=	2.0000	BR=	2.0000
2	0.0245	RR=	2.0000	BR=	2.0000
3	-3.3659	RR=	2.0000	BR=	2.0000

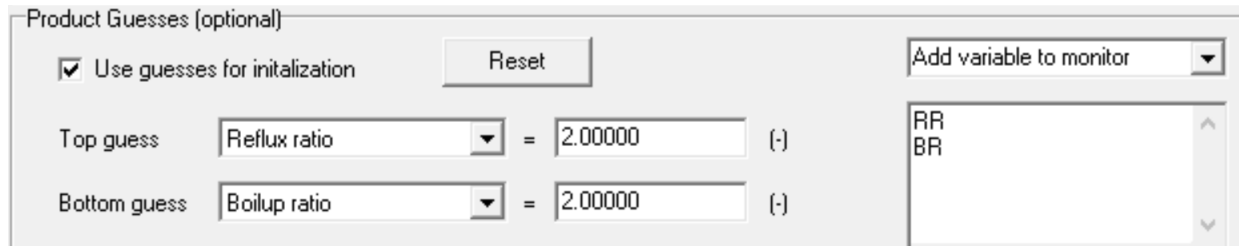
Run level: Actual Specification

Iteration	log(Err/Tol)	RR=	BR=	RR=	BR=
0	5.8519	RR=	2.0000	BR=	2.0000
1	8.2966	RR=	0.6667	BR=	2.0000
2	8.1547	RR=	2.0000	BR=	2.0000
3	8.4565	RR=	3.7340	BR=	2.0000
4	8.3172	RR=	5.6209	BR=	2.4333
5	8.0825	RR=	5.6209	BR=	3.6452
6	8.4555	RR=	1.8736	BR=	10.9355
7	8.2195	RR=	1.3344	BR=	10.9355
8	7.7417	RR=	0.9727	BR=	8.6200
9	7.3584	RR=	1.0313	BR=	8.7994
10	6.7989	RR=	0.9703	BR=	8.4453
11	6.4578	RR=	0.8044	BR=	7.7885
12	4.8517	RR=	0.8827	BR=	8.1075
13	4.4019	RR=	0.8952	BR=	8.1584
14	0.3917	RR=	0.8951	BR=	8.1580
15	-3.5835	RR=	0.8951	BR=	8.1580

where we see that the number of iterations went down from 30 to 18, but only 3 iterations were needed for the first pass where we used fixed reflux and boilup ratios as specifications..

Now, as we said, this isn't a particularly difficult case. None the less, there will be many simulations where forcing the method to use fixed flow specifications in the first pass will prove very useful.

*One important caveat before we leave this topic: ChemSep does not require estimates of the boil-up ratio and/or reflux ratio to be provided by you in order to use this fixed flow method. It can, however, be a definite advantage to do so. Estimates can be provided on the lower part of the specifications panel as illustrated below for this particular case:*



Here we also see the location for selecting the variables to monitor during the iteration process.

As shown, above, failure to provide these estimates will not prevent ChemSep from running, but it can, in some cases, prevent it from converging. In this case the default guesses of 2 for the reflux ratio and 2 for the reboil ratio are good enough for convergence but are not especially close to the correct results. Here is what happens if we use 1 and 8 as the initial estimates for these flow ratios:

```
Run level: Complete model: Using fixed flows
  Iteration  log(Err/Tol)  RR=  BR=
    0         4.7609  RR=  1.0000  BR=  8.0000
    1         4.5439  RR=  1.0000  BR=  8.0000
    2         2.5429  RR=  1.0000  BR=  8.0000
    3        -0.5771  RR=  1.0000  BR=  8.0000
Run level: Actual Specification
  Iteration  log(Err/Tol)  RR=  BR=
    0         4.8854  RR=  1.0000  BR=  8.0000
    1         5.2835  RR=  0.7988  BR=  7.7919
    2         5.4007  RR=  0.8828  BR=  8.0440
    3         5.5622  RR=  0.8951  BR=  8.1685
    4         5.5855  RR=  0.8951  BR=  8.1581
    5         5.2105  RR=  0.8951  BR=  8.1580
    6         3.8894  RR=  0.8951  BR=  8.1580
    7         1.3217  RR=  0.8951  BR=  8.1580
    8        -1.6852  RR=  0.8951  BR=  8.1580
Run level: Report
Convergence obtained in 11 iterations
```

Convergence now is achieved in only 11 iterations!

## Specification Continuation

Version 8.11 introduces one more technique for aiding convergence of difficult cases that involve condenser and/or reboiler specifications of either the heat duty or the temperature. This technique employs the method described in the prior section, whereby we force use of standard specifications in the first (and, if necessary) second pass. Instead of a final attempt with Newton's method that immediately imposes the actual specification, we make use of a "continuation method" in which we "creep up" on the desired specification in a series of steps.

Perhaps this will become more clear with an example involving the specification of the reboiler temperature.

In our example we type the following text into the *Comments* section of the title panel:

```
[Algorithm]
Use Fixed Flow Specs
Reboiler Temperature
10 F 3
```

(These lines should appear as a contiguous block; that is, no empty lines in between the lines shown above. For the first 3 lines of this block it is not necessary to start typing at the beginning of the line, and you can type other characters on the same lines, but the key phrases shown on those lines must appear *exactly* as shown and they must be on consecutive lines. The 4<sup>th</sup> line must appear as if it is to be processed correctly. If multiple blocks that start with a key word in [ ] are to be used in the same simulation then they should be separated by an empty line.)

The first two lines of this block were encountered in the prior section; they force ChemSep to use standard specifications (that fix the flows) in the first (and, possibly second depending on the algorithm of choice) pass with Newton's method. The 3<sup>rd</sup> and 4<sup>th</sup> lines in the block above inform us that it is the reboiler temperature that will be varied in a series of steps. The 4<sup>th</sup> line contains 3 items (all must be present): the final value of the reboiler temperature, the units for that value (ChemSep will recognize F, C, R, and K). The final number on that 4<sup>th</sup> line tells us that we are going to try and creep up on the desired specification (10 F) in 3 steps. Here is the running output from this simulation. (Some of the iterations have been omitted to save space.)

Printed to screen/file	Explanation										
Run level: Simple model: Using fixed flows	Simulation starts using fixed flow specifications.										
Solving ideal K, constant H+k model	Simulation using a two-pass method with ideal K-values, constant enthalpy, and constant mass transfer coefficients in the first pass.										
<table><tr><td>Iteration</td><td>log(Err/Tol)</td></tr><tr><td>0</td><td>4.4253</td></tr><tr><td>1</td><td>4.0412</td></tr><tr><td>...</td><td></td></tr><tr><td>4</td><td>-1.3453</td></tr></table>	Iteration	log(Err/Tol)	0	4.4253	1	4.0412	...		4	-1.3453	Summary of the iterations in this first pass. The size of the number indicates the extent to which the equations are not solved. A negative number indicates this simulation has converged.
Iteration	log(Err/Tol)										
0	4.4253										
1	4.0412										
...											
4	-1.3453										

Run level: Complete model: Using fixed flows	Now switching to a simulation that uses the selected K-value, enthalpy and mass transfer coefficient models. The fixed flow specifications are, however, retained.
<pre> Iteration    log (Err/Tol)   0          4.7895   1          4.4281   2          4.5645   ...   8          1.5998   9          0.5529  10         -0.0500 </pre>	<p>The iterations for this second simulation. Some lines omitted to save space</p> <p>A negative number indicates this simulation has converged.</p>
<pre> Run: Reboiler continuation: Run: Current spec =      214.19 </pre>	<p>The program is telling us that it is about to start a continuation method in which the reboiler specification is varied.</p> <p>The current value of the reboiler specification variable (reboiler temperature in Kelvin in this case) also is reported.</p>
Run: Reboiler spec =      237.18	The new reboiler specification.
<pre> Iteration    log (Err/Tol)   0          6.3615   1          6.1137   2          5.4777   ...  11          0.0288  12         -0.3878 </pre>	Iterations for this simulation using the reboiler specification from the panel above.
Run: Reboiler spec =      260.16	The next value of the reboiler specification.
<pre> Iteration    log (Err/Tol)   0          6.3615   1          6.1135   2          5.4752   ...   6          0.0136   7         -0.2788 </pre>	Iterations for this simulation using the reboiler specification from the panel above.
Run: Reboiler spec =      283.15	The final (in this case) value of the reboiler specification.
<pre> Iteration    log (Err/Tol)   0          6.3615   1          6.1135   ...   4         -0.5743 </pre>	Iterations for this simulation using the reboiler specification from the panel above.
<pre> Run level: Report Convergence obtained in 37 iterations </pre>	<p>These lines tell us that the entire simulation has converged successfully. The report file is being created.</p>

Here is an example from a simulation that did not converge to a final specification value:

Printed to screen/file	Explanation
Run level: Simple model: Using fixed flows	Simulation starts using fixed flow specifications.
Solving CMO model	Simulation using a two-pass method with constant enthalpy in the first pass.
<pre> Iteration    log (Err/Tol)   0           4.2062   1           3.5011   2           1.9741   3          -2.2900 </pre>	Summary of the iterations in this first pass. The size of the number indicates the extent to which the equations are not solved. A negative number indicates this simulation has converged.
Run level: Complete model: Using fixed flows	Now switching to a simulation that uses the selected K-value, enthalpy and mass transfer coefficient models. The fixed flow specifications are, however, retained.
<pre> Iteration    log (Err/Tol)   0           4.6852   1           5.6323   2           5.6286   ...  10           0.2135  11          -0.2335 </pre>	<p>The iterations for this second simulation. Some lines omitted to save space</p> <p>A negative number indicates this simulation has converged.</p>
<pre> Run: Reboiler continuation: Run: Current spec =      381.46 </pre>	<p>The program is telling us that it is about to start a continuation method in which the reboiler specification is varied.</p> <p>The current value of the reboiler specification variable (reboiler temperature in Kelvin in this case) also is reported.</p>
Run: Reboiler spec =      382.49	The new reboiler specification.
<pre> Iteration    log (Err/Tol)   0           5.7092   1           4.7930   2           4.0805   3           1.8047   4          -0.5703 </pre>	Iterations for this simulation using the reboiler specification from the row above.
Run: Reboiler spec =      382.49	The next value of the reboiler specification.
<pre> Iteration    log (Err/Tol)   0           5.7092   1           5.4459   2           5.8268   3           6.0263   4           6.2421   ...  13           9.3706  14           9.7978  15           9.6107  16           9.7543   ...  20           9.7895  21           9.9150 </pre>	<p>Iterations for this simulation using the reboiler specification from the panel above.</p> <p>Note that the numbers to the right of each line show no tendency to decrease.</p> <p>This is an indication that the simulation is unlikely to converge.</p> <p>(In this particular case the simulation crashed after a few dozen additional iterations.)</p>



The behavior illustrated immediately above is typical of a simulation in which the actual specification is infeasible. In this particular case a reboiler temperature was specified and it is impossible for the particular mixture being distilled to reach that temperature (at the specified pressure). Of course, there are many other reasons why a simulation can fail to converge, but a specification continuation such as those shown here can provide insight into what is or is not possible!

A brief reminder, as of ChemSep Version 8.11 the following condenser / reboiler specifications can be used in a specification continuation:

1. Reboiler temperature.
2. Reboiler heat duty
3. Condenser temperature
4. Condenser heat duty

and those are the key words that should appear on line 3 of the text block in the *Comments* box on the *Title* panel.