Polymers Plus®

POLYMER PROCESS MODELING

Version 3

With Aspen Plus® 10

User Guide

AspenTech®

VOLUME 2

The flowsheet graphics and plot components of Aspen Plus were developed by MY-Tech, Inc.

The License Manager portion of this product is based on:

Use of Aspen Plus and This Manual

This manual is intended as a guide to using Aspen Plus process modeling software. This documentation contains AspenTech proprietary and confidential information and may not be disclosed, used, or copied without the prior consent of AspenTech or as set forth in the applicable license agreement. Users are solely responsible for the proper use of Aspen Plus and the application of the results obtained.

Although AspenTech has tested the software and reviewed the documentation, the sole warranty for Aspen Plus may be found in the applicable license agreement between AspenTech and the user. **ASPENTECH MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENTATION, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.**

CONTENTS

VOLUME 1

Chapter 1 Introduction

Chapter 2 Polymer Structural Characterization

Section 2.1 Component Classification

Section 2.2 Polymer Structural Properties

Section 2.3 Structural Property Distributions

Section 2.4 End-Use Properties

Chapter 3 Thermodynamic Properties

Section 3.1 Van Krevelen Property Models

Section 3.2 Tait Molar Volume Model

Section 3.3 Polymer Viscosity Models

Section 3.4 Flory-Huggins Activity Coefficient Model

Section 3.5 NRTL Activity Coefficient Models

Section 3.6 UNIFAC Activity Coefficient Model

Section 3.7 Polymer UNIFAC Free Volume Model

Section 3.8 Polymer Ideal Gas Property Model

Section 3.9 Sanchez-Lacombe EOS Model

Section 3.10 Polymer SRK EOS Model

Section 3.11 SAFT Equation-of-State Model

Chapter 4 Polymerization Reactions

Section 4.1 Step-Growth Polymerization Model

Section 4.2 Free-Radical Bulk Polymerization

Section 4.3 Emulsion Polymerization Model

Section 4.4 Ziegler-Natta Polymerization Model

Section 4.5 Ionic Polymerization Model

Section 4.6 Segment-Based Reaction Model

VOLUME 2

Chapter 5 Steady-State Flowsheeting

Section 5.1 Steady-State Unit Operation Models

Section 5.2 Plant Data Fitting

Section 5.3 User Models

Section 5.4 Application Tools

Chapter 6 Run-Time Environment

Appendix A Component Databanks

Appendix B Physical Property Methods

Appendix C Van Krevelen Functional Groups

Appendix D Tait Model Coefficients

Appendix E Mass Based Property Parameters

Appendix F Equation-of-State Parameters

Appendix G Kinetic Rate Constant Parameters

Appendix H Fortran Utilities

Input Language Reference

Glossary

Index

5

STEADY-STATE FLOWSHEETING

Polymers Plus allows you to model polymerization processes in both steady-state and dynamic mode. In this chapter, flowsheeting capabilities for modeling processes in steady-state mode are described.

Topics covered include:

- Polymer Manufacturing Flowsheets
- Modeling Polymer Process Flowsheets
- Steady-State Modeling Features

Following this introduction, Polymers Plus flowsheeting capabilities for modeling steady state processes are discussed in several sections.

POLYMER MANUFACTURING FLOWSHEETS

Polymer production processes are usually divided into the following major steps shown in Figure 5.1:

1. Monomer synthesis and purification
2. Polymerization
3. Recovery/separation
4. Polymer processing

The modeling issues of interest in each of these steps were discussed in Chapter 1. The focus here is on the various unit operations required in these processing steps.

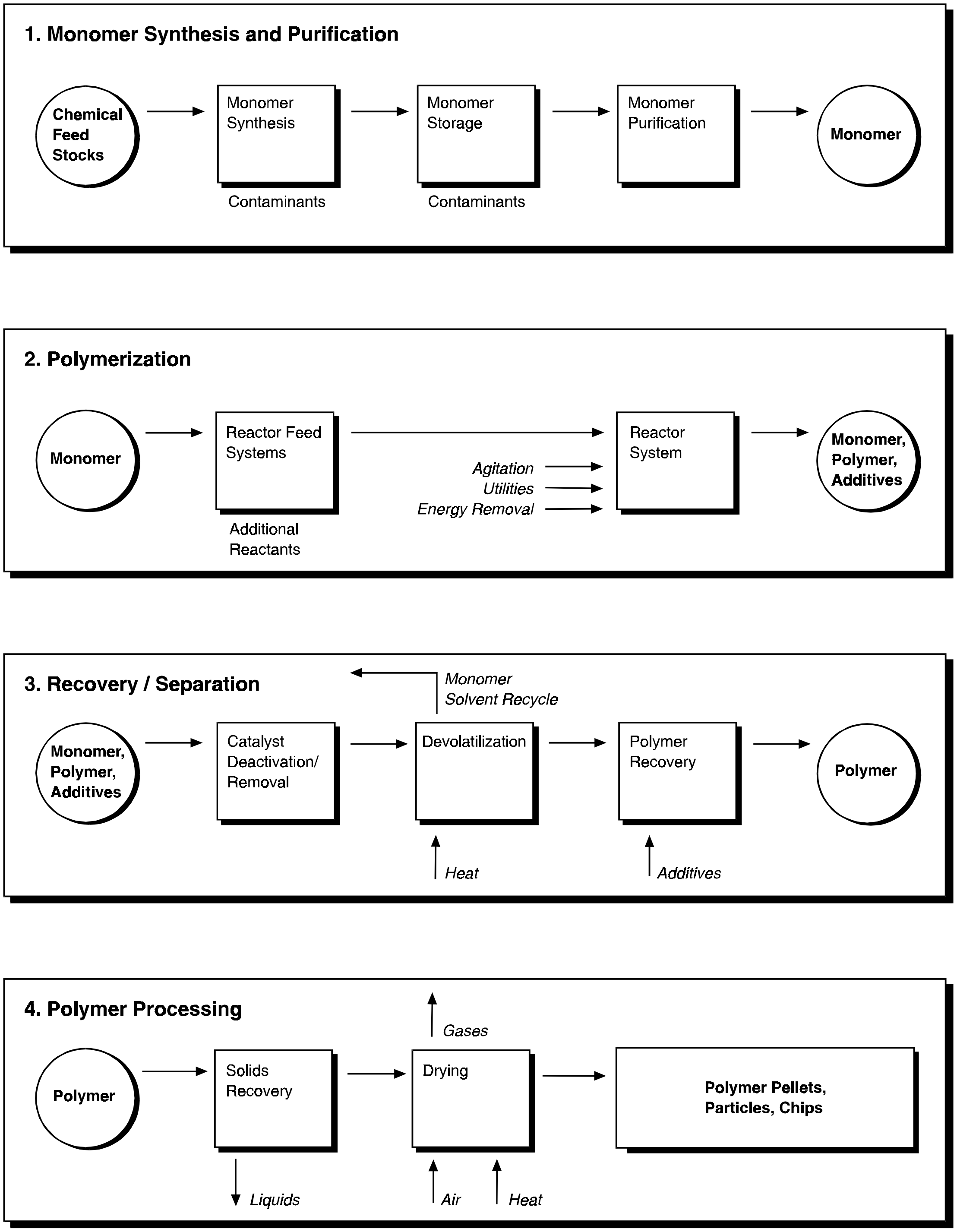


Figure 5.1 *Major Steps in Polymer Production Processes*

Monomer Synthesis

During monomer synthesis and storage the engineer is concerned with purity since the presence of contaminants, such as water or dissolved gases, may adversely affect the subsequent polymerization stage by poisoning catalysts, depleting initiators, causing undesirable chain transfer or branching reactions which would cause less effective heat removal. Another concern is the prevention of monomer degradation through proper handling or the addition of stabilizers. Control of emissions, and waste disposal are also important factors.

Polymer Synthesis

The polymerization step is the most important step in terms of capital and operating costs. The desired outcome for this step is a polymer product with specified properties (e.g. molecular weight distribution, melt index, viscosity, crystallinity) for given operating conditions. The obstacles that must be overcome to reach this goal depend on the type of polymerization process.

Polymerization processes may be batch, semi-batch, or continuous. In addition, they may be carried out in bulk, solution, suspension, or emulsion. Bulk continuous systems provide better temperature and molecular weight control at the expense of conversion; batch systems offer less control over molecular weight. In addition, they may result in a high viscosity product and require high temperatures and pressures. Solution systems also provide good temperature control but have associated with them the cost of solvent removal from the polymer.

In summary, for the polymerization step, the mechanisms that take place during the reaction introduce changes in the reaction media which in turn make kinetics and conversion, residence time, agitation, and heat transfer the most important issues for the majority of process types.

Recovery / Separations

The recovery/separation step is the step where the desired polymer produced is further purified or isolated from by-products or residual reactants. In this step, monomers and solvents are separated and purified for recycle or resale. The important issues for this step are phase equilibrium, heat and mass transfer.

Polymer Processing

The last step, polymer processing, can also be considered a recovery step. In this step, the polymer slurry is turned into solid pellets or chips. Heat of vaporization is an important issue in this step (Grulke, 1994).

MODELING POLYMER PROCESS FLOWSHEETS

The obvious requirement for the simulation of process flowsheets is the availability of unit operation models. Once these unit operation models are configured, they must be adjusted to match the actual process data. Finally, tools must be available to apply the fitted model to gain better process understanding and perform needed process studies. As a result of the application of the process models, engineers are able to achieve goals such as production rate optimization, waste minimization and compliance to environmental constraints. Yield increase and product purity are also important issues in the production of polymers.

STEADY-STATE MODELING FEATURES

Polymers Plus has tools available for addressing the three polymer process modeling aspects.

Unit Operations Modeling Features

A comprehensive suite of unit operations for modeling polymer processes is available in Polymers Plus. These include mixers, splitters, heaters, heat exchangers, single and multistage separation models, reactors, etc. The unit operation models available are discussed in Section 5.1.

Plant Data Fitting Features

Several tools are available for fitting process models to actual plant data. Property parameters may be adjusted to accurately represent separation and phase equilibrium behavior. This can be done through the Data Regression System (DRS). See *the Aspen Plus Properties Reference Manual* for information about DRS.

Another important aspect of fitting models to plant data has to do with the development of an accurate kinetic model within the polymerization reactors. The powerful plant data fitting feature (Data-Fit) can be used for fitting kinetic rate constant parameters. This is explained in Section 5.2.

Process Model Application Tools

The tools available for applying polymer process models include capabilities for performing sensitivity and case studies, for performing optimizations, and for applying design specifications. These features are discussed in Section 5.4.

REFERENCES

Dotson, N. A, R. Galván, R. L. Laurence, M. Tirrell, Polymerization Process Modeling, VCH Publishers, New York (1996).

Grulke, E. A., Polymer Process Engineering, Prentice Hall, Englewood Cliffs, New Jersey (1994).

5.1

STEADY-STATE UNIT OPERATION MODELS

This section summarizes some typical usage of the Aspen Plus unit operation models to represent actual unit operations found in industrial polymerization processes.

Topics covered include:

- Summary of Aspen Plus Unit Operation Models
- Distillation Models
- Reactor Models
- Mass-Balance Reactor Models
- Equilibrium Reactor Models
- Kinetic Reactor Models
- Treatment of Component Attributes in Unit Operation Models

SUMMARY OF ASPEN PLUS UNIT OPERATION MODELS

Aspen Plus includes a number of basic unit operation models that are typically used to represent one or more unit operations found in real processes. These models may be used alone to represent equipment such as pumps, heaters, valves, mixers, etc., or they may be used as generic “tools” to build models of more complex unit operations. Table 5.1 summarizes the models available.

Table 5.1 Summary of Unit Operation Models

Basic Unit Operation Models and Stream Manipulators	
Dupl	Copies inlet stream to any number of outlet streams
Flash2	Performs two-phase (vapor-liquid) or three-phase (vapor-liquid-solid) phase equilibrium calculations
Flash3	Performs three-phase (vapor-liquid-liquid) phase equilibrium calculations
FSplit	Splits inlet stream to any number of outlet streams
Heater	Represents heaters, coolers, or mixers with known heat duty or specified temperature
Mixer	Adiabatic mixing of any number of feed streams
Mult	Multiplies stream flow rates by a constant
Pipe	Calculates pressure drop through pipelines
Pump	Represents pumps or liquid standpipes (pressure must be specified)
Distillation and Fractionation Models	
Sep	Mass-balance model for separation operations with any number of product streams
Sep2	Mass-balance model for separation operations with two product streams
RadFrac	Predictive multistage distillation model
MultiFrac	Predictive model for complex distillation operations with multiple columns
Reactor Models	
RStoic	Mass-balance model based on specified conversion for any number of stoichiometric reactions
RYield	Mass-balance model based on specified product yield for any number of stoichiometric reactions
REquil	Chemical equilibrium calculated from user-specified equilibrium constants
RGibbs	Chemical equilibrium calculated by Gibbs free-energy minimization
RCSTR	Predictive, reaction rate-based model to simulate continuous stirred tank reactors
RPlug	Predictive, reaction rate-based model to simulate continuous plug-flow reactors
RBatch	Predictive, reaction rate-based model to simulate batch and semi-batch stirred tank reactors

Dupl

DUPL

The Dupl block copies one inlet stream to two or more outlet streams. By design, the mass flow rate and attribute rates out of this block will be greater than the flow rates into the block, violating mass and attribute conservation principles.

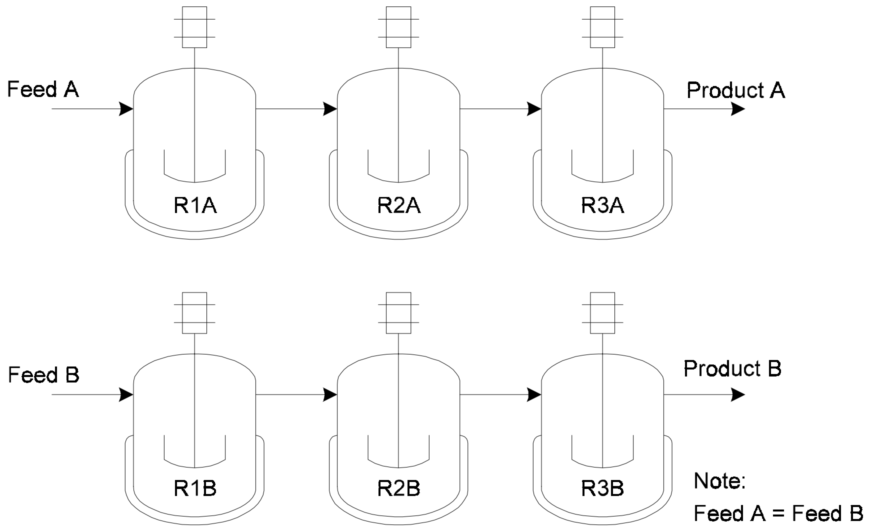
Frequently, the Dupl block is used as a shortcut to reduce the simulation time required to model a process consisting of two or more parallel process lines. For example, consider the process shown in Figure 5.2. The second unit (“R2A” and “R2B”) in the “A” and “B” lines consist of identical unit operations operating at the same conditions. The third unit (“R3A” and “R3B”) operates differently in the two lines. Since the process lines are identical up to the third unit, there is no need to include both process lines in the model. Instead, we can consider one line, such as “A” and duplicate the outlet stream at the point where the process conditions diverge from each other.

Another application of the Dupl model is to carry out simple case studies. For example, assume there are two proposed scenarios for carrying out a given reaction. In the first scenario, the reaction is carried out at a high temperature in a small reactor with a short residence time. In the second scenario, the reaction is carried out at a low temperature in a large reactor with high residence times. The two reactors can be placed in a single flow sheet model. The duplicator block is used to copy one feed stream to both reactors. The two “cases” can be compared by examining the stream summary.

Operating Conditions

	R1A	R1B	R2A	R2B	R3A	R3B
Temperature, °C	250	250	260	260	270	265
Pressure, torr	760	760	1200	1200	1500	1700
Volume, liter	2000	2000	1500	1500	1000	1200

Process Diagram



Model Block Diagram

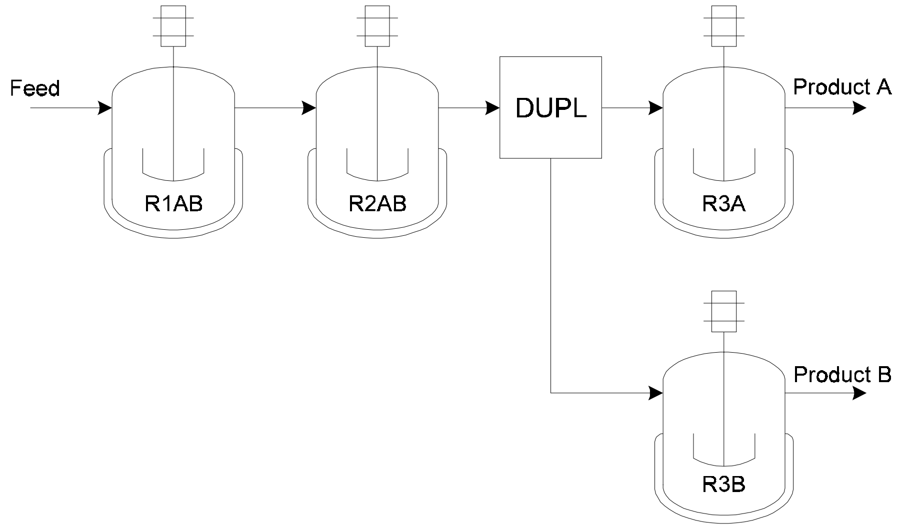
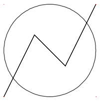


Figure 5.2 Using Dupl Block to Eliminate Redundant Calculations

Flash2



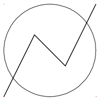
The Flash2 block carries out a phase-equilibrium calculation for a vapor-liquid split. The “chemistry” feature of this block can be used to extend the phase equilibrium to vapor-liquid-solid systems. The free-water option can be used to extend the phase equilibrium calculations to include a free water phase in addition to the organic liquid phase.

The Flash2 model can be used to simulate simple flash drums with any number of feed streams. The model is also a good tool for representing spray condensers, single-stage distillations, knock-back condensers, decanters, and other types of equipment which effectively operate as one ideal stage.

The Flash2 model assumes a perfect phase split, but an entrainment factor can be specified to account for liquid carryover in the vapor stream. The entrainment factor is specified by the user, it is not calculated by the model. If a correlation between the vapor flow rate and the entrainment rate is available, this correlation can be applied to the model using a Fortran block which reads the vapor flow rate calculated by the Flash block, calculates the entrainment rate, and writes the resulting prediction back to the Flash block. Note that this approach creates an information loop in the model which must be converged.

The Flash2 block does not fractionate the polymer molecular weight distribution. Instead, the molecular weight distribution of the polymer in each product stream is assumed to be the same as the feed stream.

Flash3



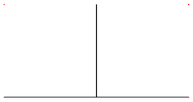
The Flash3 block carries out phase-equilibrium calculations for a vapor-liquid-liquid splits. The liquid phases may be organic-organic (including polymer-monomer) or aqueous-organic. For aqueous-organic systems, the Flash3 model is more rigorous than the Flash2/free water approach described above. The key difference is that the Flash3 model considers dissolved organic compounds in the aqueous phase while the free water approach assumes a pure water phase.

Generally, three-phase flashes are more difficult to converge than two-phase flashes. Three-phase flash failures may indicate bad binary interaction parameters between the components. The problem may also stem from bogus vapor pressures or heats of formation. In general, it is a good idea to study two-phase splits for the system in question before attempting to model a three-phase decanter or reactor.

As with the two-phase flash, the three-phase flash is more stable if temperature and pressure are specified. Other options, such as duty and vapor fraction, are more difficult to converge. Temperature estimates may aid convergence in duty-specified reactors.

The Flash3 block does not fractionate the polymer molecular weight distribution. Instead, the molecular weight distribution of the polymer in each product stream is assumed to be the same as the feed stream.

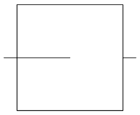
FSplit



The flow splitter block, FSplit, is used to represent valves or tanks with several outlets. The outlet flow rates can be specified on a mass, mole, or volume basis, or they can be specified as a fraction of the feed stream. In general, the fraction specifications are best because they are independent of the feed stream flow rates. This makes the model more flexible and reliable when using tools like SENSITIVITY or DESIGN-SPEC which might directly or indirectly manipulate the stream which is being split. The FSplit block can also be used with reactor models to account for back-mixing.

The FSplit block assumes that the class 2 polymer attributes split according to mass mixing rules. For example, if the outlet stream is split 60:40, then the class 2 attributes, such as the segment flow rates, are also split 60:40. This approach is identical to assuming that the properties of the polymer in each outlet stream are the same as the properties of the polymer in the inlet stream.

Heater



Heater can be used to represent heaters, coolers, mixers, valves, or tanks. The Heater block allows you to specify the temperature or heat duty of the unit, but does not carry out rigorous heat exchange equations. Any number of feed streams can be specified for the Heater block. This block follows the same mixing rules as the Mixer model.

Mixer

The mixer block, Mixer, is used to mix two or more streams to form a single mixed outlet. The mixer block can be used to represent mixing tanks, static mixers, or simply the union of two pipes in a tee. The Mixer model assumes ideal, adiabatic mixing. The pressure of the mixer can be specified as an absolute value or as a drop relative to the lowest feed stream pressure.

The Mixer model is functionally equal to the Heater model, except it only allows adiabatic mixing. For this reason, the Heater model may be a better choice for modeling mixing tanks.

The Mixer block assumes that the class 2 polymer attributes are additive. For example if stream “A” and “B” are mixed to form stream “C”, and the zeroth moments of a polymer in stream “A” and “B” are 12 kmol/sec and 15 kmol/sec, then the polymer in the product stream has a zeroth moment of 12+15=27 kmol/sec.

Mult

MULT

The Mult block is used to multiply the flow rate of a stream. A common application of this block is to collapse two parallel process line models into a single line to avoid unnecessary duplicate calculations. For example, consider the process shown below in Figure 5.3. In this process, the “A” and “B” lines consist of identical equipment with the same operating conditions. The Mult blocks “HALF” and “TWICE” are used to divide the feed stream flow rate by two after R1, representing the split between lines, and to double the product flow rate, representing the junction of the parallel lines into a single line at R3. This technique avoids the duplicate calculations for R2 “A” and “B” reactors, which should give the same results. This technique can save a great deal of simulation time.

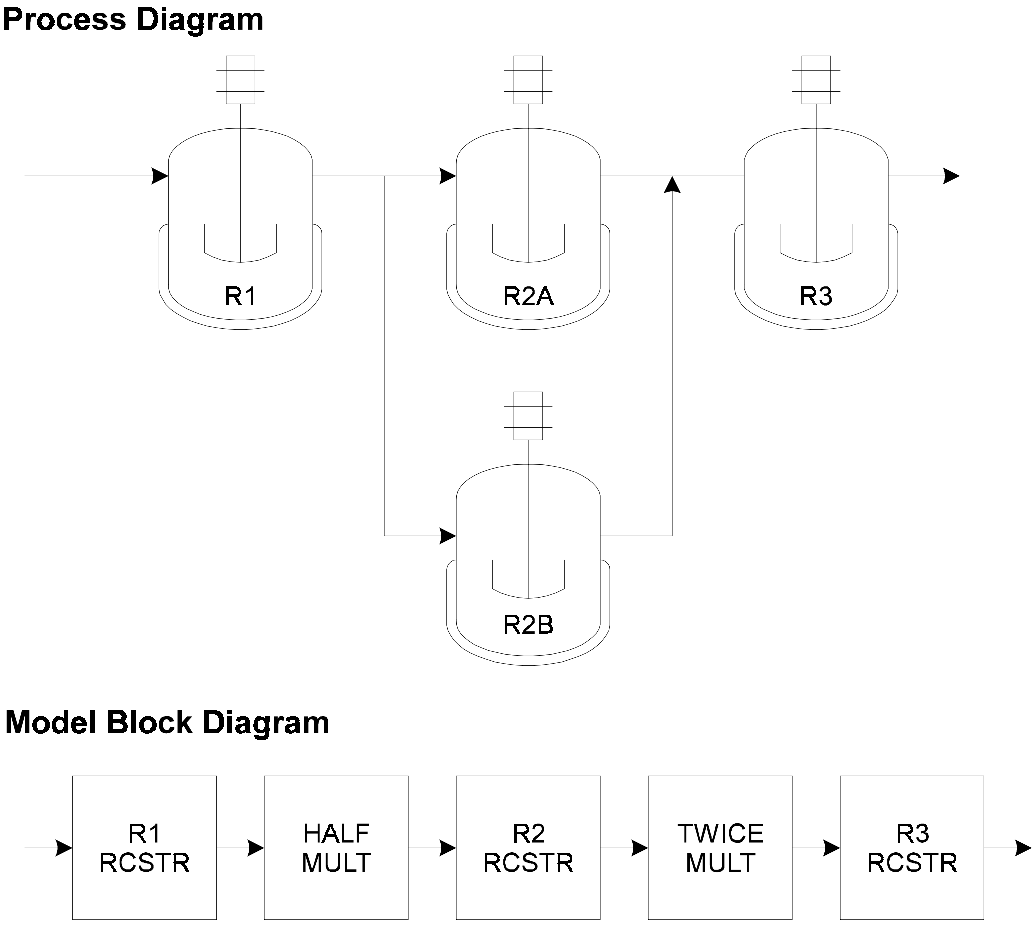
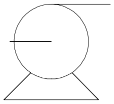


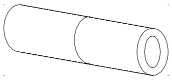
Figure 5.3 *Representation of Redundant Flowsheet Section Using Mult Blocks*

Pump



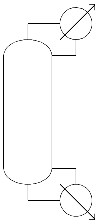
The Pump block changes the pressure of a stream. This block can be used to represent an actual pump, or it can be used to represent pressure increases due to liquid head in standpipes.

Pipe



The Pipe model is used to calculate pressure drops in pipelines. The algorithms in this model are not designed for non-ideal fluids such as polymers, so the pipe model should be used with caution in polymer process models. A better option to calculate pressure drops in polymer pipelines is to use RPlug with a user-written pressure-drop subroutine.

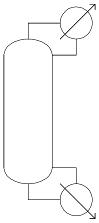
Sep



The Sep block is a generic separation model which allows component fractionation between two or more product streams. The products can be split according to flow rate or fractional specifications. The Sep block is commonly used to represent distillation columns or other separation equipment when the product stream purity is well known and the details of the separation process are not important.

The Sep block does not fractionate the polymer molecular weight distribution. Instead, the molecular weight distribution of the polymer in each product stream is assumed to be the same as the feed stream.

Sep2



The Sep2 block is a generic separation model which allows component fractionation between two product streams. The products can be split according to flow rate or fractional specifications. The Sep2 block is commonly used to represent distillation columns or other separation equipment when the product stream purity is well known and the details of the separation process are not important. Compared to the Sep block, the Sep2 block has more flexible input options, but it only allows two outlet streams.

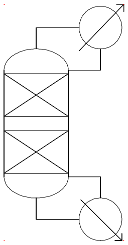
The Sep2 block does not fractionate the polymer molecular weight distribution. Instead, the molecular weight distribution of the polymer in each product stream is assumed to be the same as the feed stream.

DISTILLATION MODELS

Aspen Plus includes several shortcut distillation models (DISTL, SFRAC, etc.) which can be used to represent distillation columns. These blocks do not fractionate the polymer molecular weight distribution. Instead, the molecular weight distribution of the polymer in each product stream is assumed to be the same as the feed stream. The class-2 component attributes in each product stream are set proportional to the mass flow rate of the attributed component in each product stream.

With the exception of the RadFrac model, the rigorous distillation models in Aspen Plus do not account for component attributes.

RadFrac



The RadFrac block is a rigorous multistage distillation model for two- and three-phase systems. RadFrac allows polymer feed streams at any tray, but it does not account for polymerization reaction kinetics. The molecular weight distribution and other polymer properties are not fractionated between the phases. Instead, the class-2 component attributes of the polymer components are split at each stage in proportion to the polymer component mass fractions. For example, if 90% of the polymer fed to a given tray goes to the liquid phase leaving that tray, then 90% of the zeroth moment and other class-2 attributes are assigned to the liquid phase on that tray.

REACTOR MODELS

Aspen Plus includes three classes of reactor models which include various levels of rigor and predictive capability. These classes are: (1) mass-balance models; (2) equilibrium models; and (3) rigorous kinetic models.

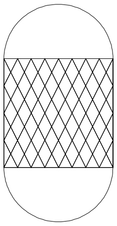
The least predictive models, RStoic and RYield, calculate output flow rates based on user-specified input flow rates. If polymer components are involved in the reactions, then the component attributes associated with the polymer components must be specified for the product stream. These models calculate the mass and energy balances, but they do not perform rigorous kinetic calculations.

The RGibbs and REquil models assume chemical and phase equilibrium. When polymer components are involved in the reactions, then the specified stoichiometry must be consistent with the reference molecular weight of the polymer component. In addition, the component attribute values for the polymer product must be specified by the user. Since the solution algorithms for these models do not consider the influence of the segmental composition of polymer components, they cannot be applied to copolymers.

Rigorous kinetic models include RCSTR (continuous stirred tank reactor), RPlug (plug-flow reactor model), and RBatch (batch stirred tank reactor). Each of these models can consider one, two, or three reacting phases. These reactor models are with the reaction kinetic models to predict product stream composition and flow rates based on calculated reaction rates.

MASS-BALANCE REACTOR MODELS

RStoic



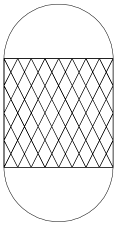
The RStoic reactor model is used to represent reaction equipment when reaction kinetics are unknown or are unimportant, for example when reactions are very fast and proceed until the limiting reagent is exhausted. RStoic requires knowledge of the net reaction stoichiometry, and the extent of reaction or conversion of a key component.

RStoic calculates the product stream flow rates based on user-specified reaction stoichiometries and extent of reaction or conversion of a key component. The reaction stoichiometry statements may include monomers, oligomers, or polymers, but may not include segments. Instead, the segment information (SFLOW or SFRAC) must be specified as component attributes in the COMP-ATTR sentence.

If polymer components are involved in any of the reactions, use the COMP-ATTR form to specify molecular weight values (MWN, MWW or PDI) or degree of polymerization (DPN, DPW or PDI) for the polymer products. Specify the SFRAC attribute for homopolymers or copolymers with a known product polymer composition. For copolymers with product compositions which depend on the feed flow rates of monomers or polymer segments, specify dummy values for the SFLOW attribute and use a user-written Fortran block to predict product segment flow rates which are consistent with the calculated product flow rates. Write the calculated results into the product stream of the RStoic block.

When some of the specified reactions involve polymers, the reaction stoichiometry must be written in a manner consistent with the reference molecular weight of the polymer component. Otherwise, the mass and energy balance calculations will not be consistent.

RYield



The RYield reactor model is used to represent reaction equipment when reaction kinetics are unknown or are unimportant, and the reactions result in a product distribution with a known yield.

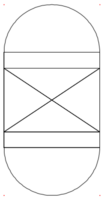
RYield calculates the product stream flow rates based on user-specified reaction stoichiometries and yield distributions. The reaction stoichiometry statements may include monomers, oligomers, or polymers, but may not include segments. Instead, the segment information (SFLOW or SFRAC) must be specified as component attributes in the COMP-ATTR sentence.

If polymer components are involved in any of the reactions, use the COMP-ATTR form to specify molecular weight values (MWN, MWW or PDI) or degree of polymerization (DPN, DPW or PDI) for the polymer products. Specify the SFRAC attribute for homopolymers or copolymers with a known product polymer composition. For copolymers with product compositions which depend on the feed flow rates of monomers or polymer segments, specify dummy values for the SFLOW attribute and use a user-written Fortran block to predict product segment flow rates which are consistent with the calculated yield. Write the calculated results into the product stream of the RYield block.

When some of the specified reactions involve polymers, the reaction stoichiometry must be written in a manner consistent with the reference molecular weight of the polymer component. Otherwise, the mass and energy balance calculations will not be consistent.

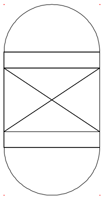
EQUILIBRIUM REACTOR MODELS

REquil



The REquil model calculates product stream flow rates using equilibrium constants determined from Gibbs free energy. The equilibrium constants are based on user-specified reaction stoichiometries and yield distributions. The reaction stoichiometry statements may include monomers or oligomers, but may not include polymers or segments. If the feed stream includes polymer components, the attributes of the polymer components will be copied to the outlet stream.

RGibbs

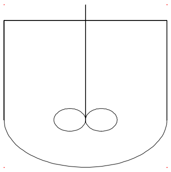


The RGibbs model uses the Gibbs free energy minimization technique to determine the composition of each phase. This algorithm cannot predict the product of equilibrium polymerization reactions. Polymer phase equilibrium, however, can be predicted by the model. The RGibbs phase equilibrium algorithm assumes that the composition and molecular weight distribution of a polymer component is equal in each of the product phases. The class-2 component attributes of the polymer component are set in proportion to the mass flow of the polymer component in each of the product phases. The mass flow rates in the product phases are set by the Gibbs free energy minimization algorithm.

To properly split component attributes among the RGibbs solution phases, use the "Phase equilibrium only" option. With this the model can predict multiple liquid phases such as three liquid phases. Surface tension effects are not considered. If you are certain that there will be no vapor phase, uncheck the "Include vapor phase" box to speed up calculations. Use one outlet stream for each predicted phase, to separate out the component attributes of that phase.

KINETIC REACTOR MODELS

RCSTR



The RCSTR model represents a continuous stirred tank reactor with one or more phases.

The model assumes perfect mixing within and between the phases, phase equilibrium, and isothermal, isobaric operation. Non-ideal mixing can be represented using a network of RCSTR models.

Temperature

The CSTR model allows you to specify duty or temperature. If duty is specified, it is a good idea to provide a temperature estimate, T-EST, to improve the convergence of the model. The maximum temperature step size, T-STEP, may also influence the CSTR convergence. This parameter defaults to 50°C, which results in substantial changes in reaction rates for reactions with typical activation energies. The temperature/duty iteration loop is referred to as the “Energy Balance” or “EB-LOOP” in the CSTR diagnostics.

Pressure

Pressure can be specified as an absolute value or as a pressure drop relative to the feed stream with the lowest pressure. In Aspen Plus, pressure drops are expressed as non-positive pressure specifications given in absolute pressure units.

Residence Time

The RCSTR model allows you to specify the effective hold-up in several different ways. For single-phase reactors, you can specify the total reactor volume or the total residence time. If the residence time is specified, then the estimated reactor volume should be specified to improve the residence-time/volume loop convergence (RT-LOOP).

When two or more condensed phases are present, the RCSTR model assumes that each condensed phase has the same residence time. This assumption is the same as the “no-slip” assumption, and it implies that the volume ratios of the condensed phases in the reactor are equal to the volume flow ratios of the condensed phases exiting the reactor.

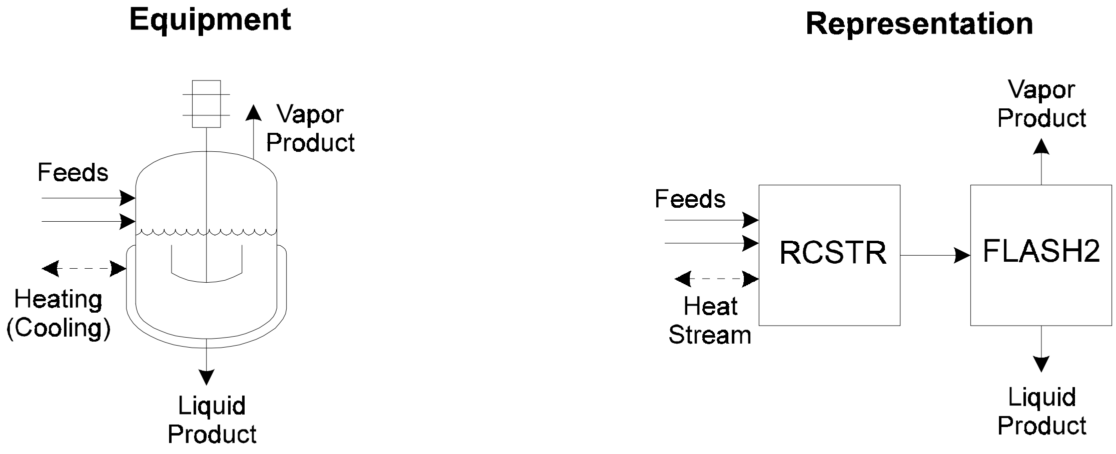
For multiphase reactors, specify the condensed phase volume or residence time in addition to the total reactor volume. Do not specify the total residence time, as this residence time is the average of the vapor and liquid phases. If the reacting phase residence time is specified, provide an estimate for the reacting phase volume. This will improve the reactor convergence. If residence time convergence is troublesome, try adjusting the volume step size.

Multiphase Reactors

The RCSTR model has one process fluid outlet stream which contains all of the phases exiting the reactor. As shown in Figure 5.4, a Flash2 or Flash3 block is used in conjunction with these blocks to split the liquid and vapor phases from the mixed outlet stream of the reactor. In this application, it is good practice to specify PRES=0 (no pressure drop) and DUTY=0 in the flash block to ensure that the phase split occurs at conditions which are consistent with the reactor outlet.

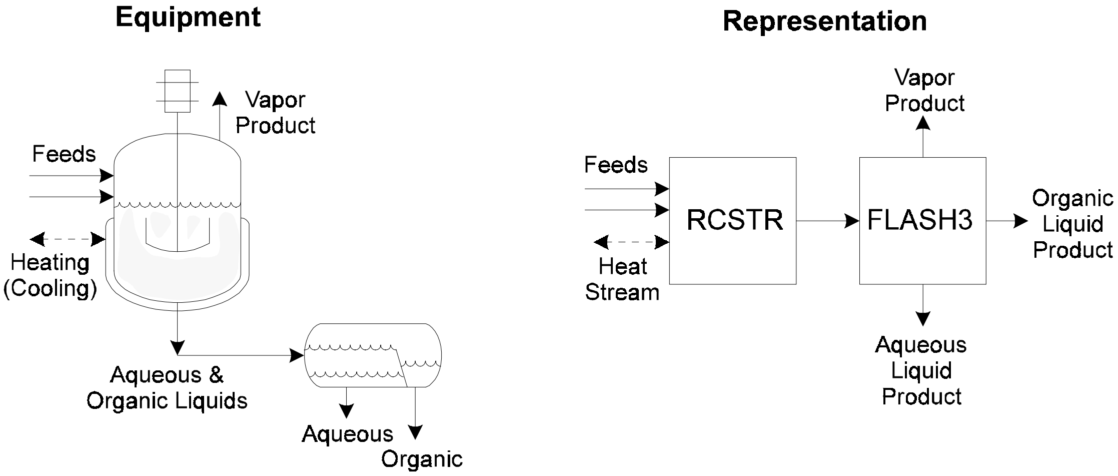
In some cases, the model may not converge if duty is specified instead of temperature. If this occurs, supply a temperature estimate to the flash model. If the problem persists, the number of iterations can be increased to 50. It is generally a bad idea to reduce the flash tolerance to bypass convergence problems - this will only make recycle stream convergence more difficult and it will make the model predictions less trustworthy. Flash convergence failures generally indicate problems in the physical property models or parameters. For example, the vapor pressure of a component may be unreasonably high or the heat of formation of a component may be unrealistic.

If flash convergence problems persist when duty is specified, but do not occur when the temperature is specified, then a TRANSFER block can be used to copy the specified reactor temperature to the specified Flash2/Flash3 temperature. This method keeps the two blocks consistent if the reactor block temperature is manipulated (for example, if the reactor temperature is adjusted in a sensitivity block).



Specify DUTY=0, PRESSURE=0 in the FLASH2 block to keep the flash results consistent with the RCSTR model or use a TRANSFER block to copy the temperature from the RCSTR outlet stream to the FLASH2 block temperature.

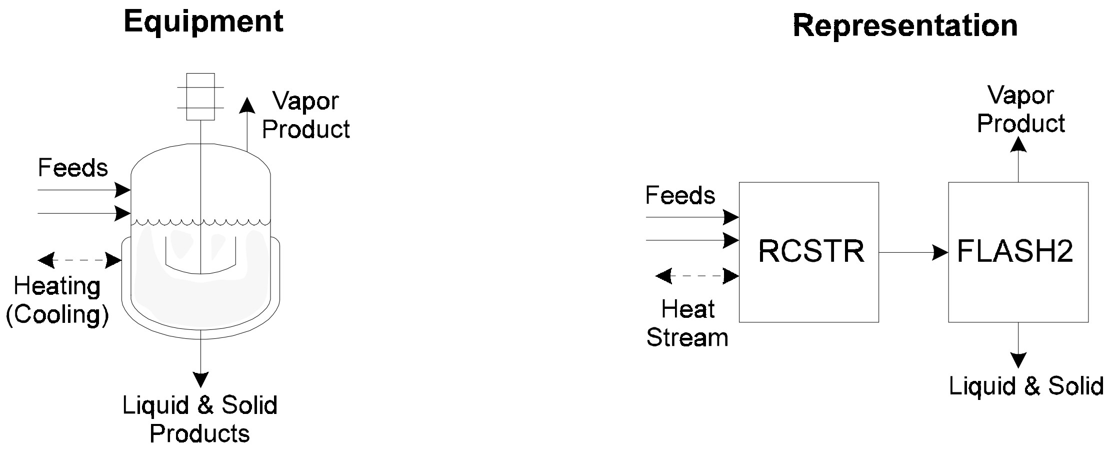
Figure 5.4 *Representation of Two-Phase Reactors*



Specify DUTY=0, PRESSURE=0 in the FLASH3 block to keep the flash results consistent with the RCSTR model or use a TRANSFER block to copy the temperature from the RCSTR outlet stream to the FLASH3 block temperature.

The RCSTR model assumes perfect mixing between and among the liquid phases.
In addition, the model assumes one liquid is dispersed as droplets in the other liquid (it does not make a difference which phase is the dispersed phase). The specified phase volume is the total volume of both liquid phases. These assumptions are consistent with assuming that the two liquid phases have the same residence time in the reactor.

Figure 5.5 *Representation of Three-Phase Reactors (Vapor-Liquid-Liquid)*



Specify DUTY=0, PRESSURE=0 in the FLASH2 block to keep the flash results consistent with the RCSTR model or use a TRANSFER block to copy the temperature from the RCSTR outlet stream to the FLASH2 block temperature.

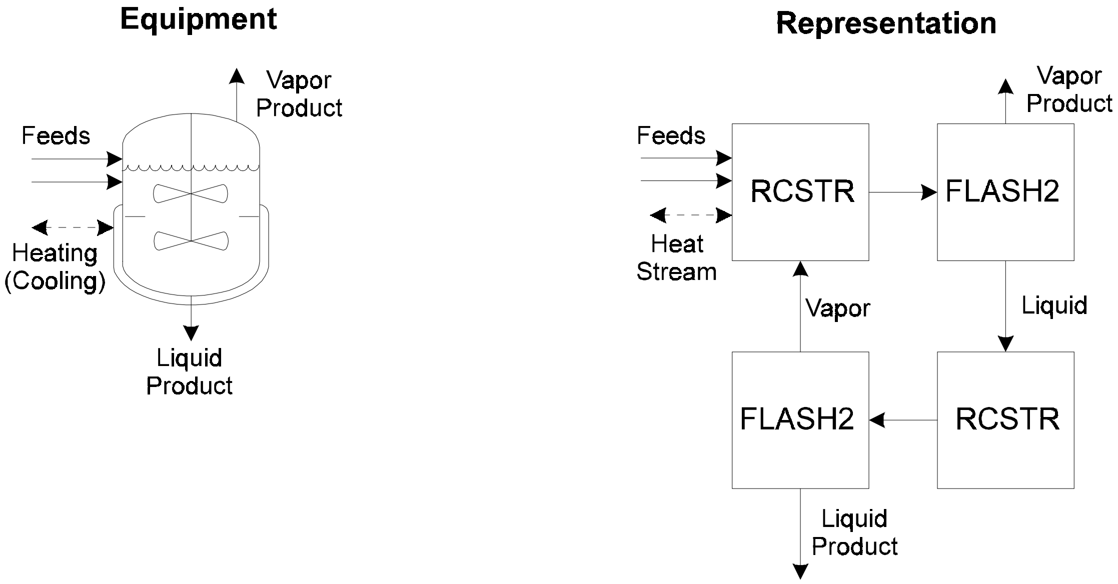
If the solid components participate in phase equilibrium, they must be defined as "salts" in a "CHEMISTRY" form. The chemistry model must be referenced in both the RCSTR and FLASH2 models, and the "TRUE-COMP" option must be set to "YES". Components which participate in chemistry reactions cannot participate in kinetic reactions (e.g., the reactions which are specified in "REACTION" forms).

Figure 5.6 *Representation of Three-Phase Reactors (Vapor-Liquid-Solid)*

Reactors with Non-Ideal Mixing

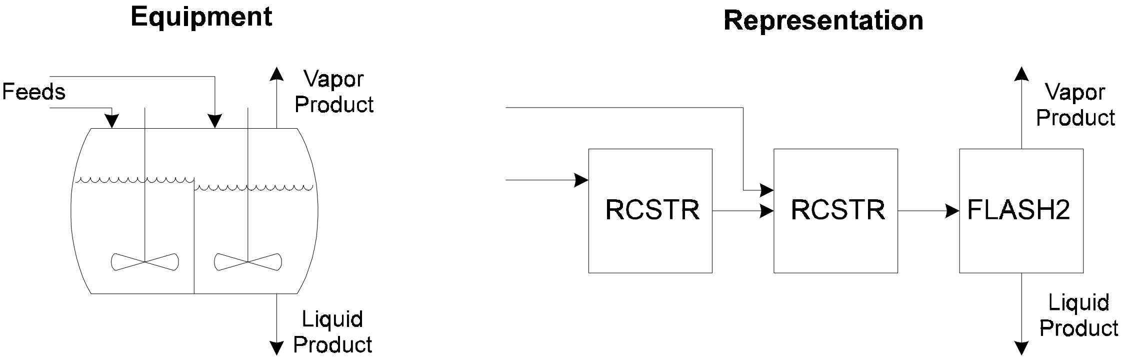
Networks of RCSTR and RPlug blocks can be used to account for non-ideal mixing found in industrial reactors. For example, many industrial reactors are divided into zones by vertical or horizontal baffles. In addition, some reactors have poor mixing characteristics which result in dead zones. The figures below demonstrate ways to model some types of real reactors.

Since many of the “network” models involve recycle loops, they may require substantially more simulation time than a single RCSTR block. In addition, the recycle loop convergence may make the model more difficult to converge. For these reasons, the simplest model which agrees with process data is always the best choice.



A FORTRAN block can be used to account for temperature and pressure difference between the upper and lower stages of the reactor. Boiling in the lower stage causes vapor to rise from the lower stage to the upper stage. This results in a recycle loop which must be converged. An initial estimate of the liquid or vapor streams between the reactor stages may enhance flowsheet convergence.

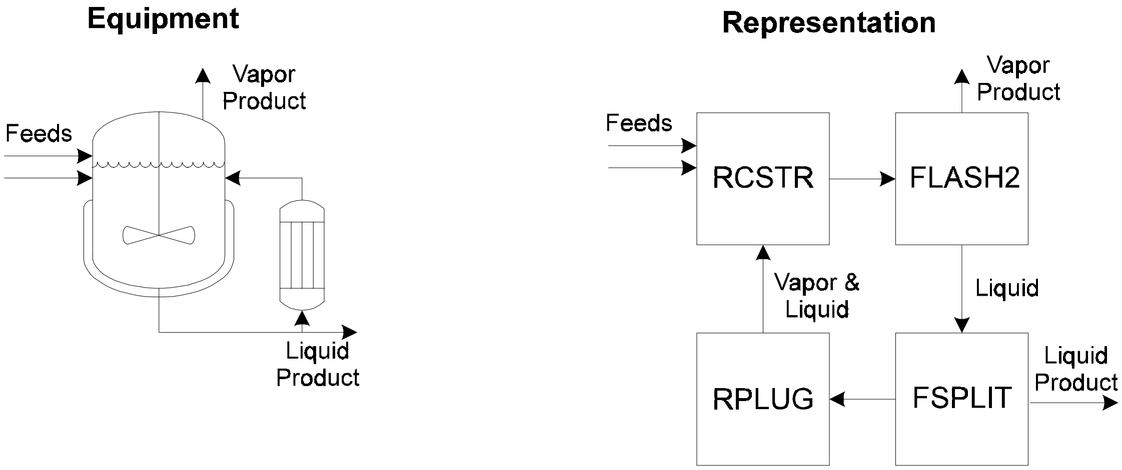
Figure 5.7 *Two-Phase CSTR with Horizontal Partitions*



The model shown above does not consider back-mixing in the liquid or vapor phases. Back-mixing can be considered by splitting the appropriate product stream with an FSPLIT block and recycling a portion back to the first RCSTR block. This approach, however, results in a recycle loop which may be difficult to converge.

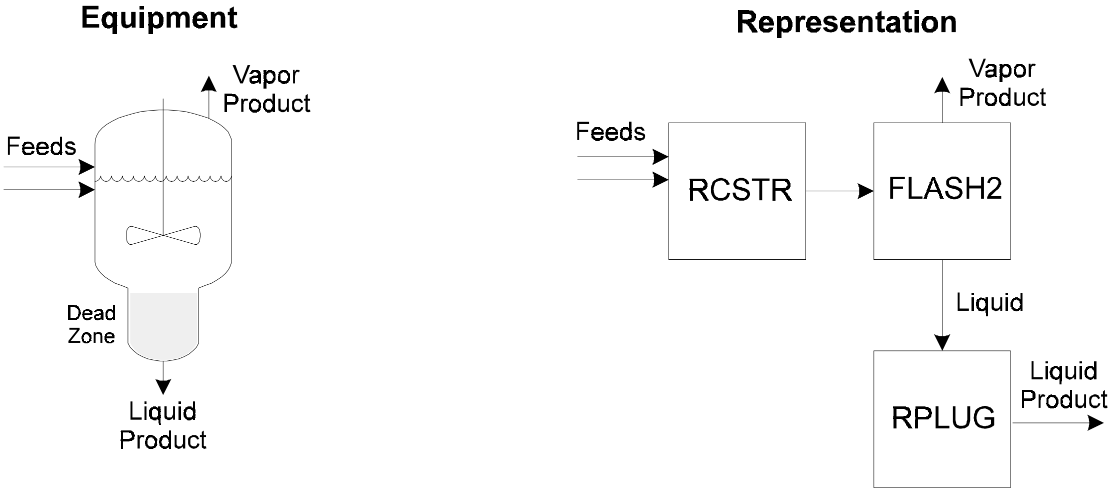
Note that no FLASH block is required between the RCSTR stages because both the liquid and vapor phases flow from the first stage to the second stage.

Figure 5.8 *Two-Phase CSTR with Vertical Partitions*



The heat exchanger is represented using a plug-flow reactor block, RPLUG. The RPLUG model can account for reactions in the heat exchanger and provides an interface to heat-transfer correlations. The circulation flow rate is specified inside the FSPLIT block. If the circulation flow rate is high, the flowsheet is difficult to converge. A good estimate of the recycle stream flow and composition may help. When the circulation flow rate is substantially higher than the product flow rate, the reactor should be modeled using a single RCSTR block.

Figure 5.9 *Two-Phase CSTR with an External Heat Exchanger*



In this example the reactor has a dead zone which falls below the well-mixed zone. The dead zone is at a higher pressure than the reactor due to the liquid head exerted by the reactor contents. The dead zone is represented using a single-phase plug flow reactor. A FORTRAN block can be used to set the dead zone pressure based on the reactor pressure.

Typically, reactors with this type of dead zone exhibit higher yields than ideal reactors.

Figure 5.10 *Two-Phase CSTR with a Dead Zone*

RCSTR Algorithm

The RCSTR model uses a trial-and-error technique to solve the mass and energy balance equations. Trial-and-error solutions are difficult to reach when the reaction rates are high, the variables cover several orders of magnitude, when many equations must be solved simultaneously, and when the variables are strongly related to each other. All of these conditions are found in polymerization reaction kinetics, making reactor convergence especially challenging.

A good understanding of the design of the RCSTR model is required in order to troubleshoot convergence problems. Otherwise, it may be difficult to understand how to apply the various convergence parameters to improve the reliability of the model.

The RCSTR algorithm consists of a series of nested loops, as shown in Figure 5.11. The loops are solved from the inside to the outside using various trial-and-error solver algorithms. Some convergence parameters are associated with each of these loops.

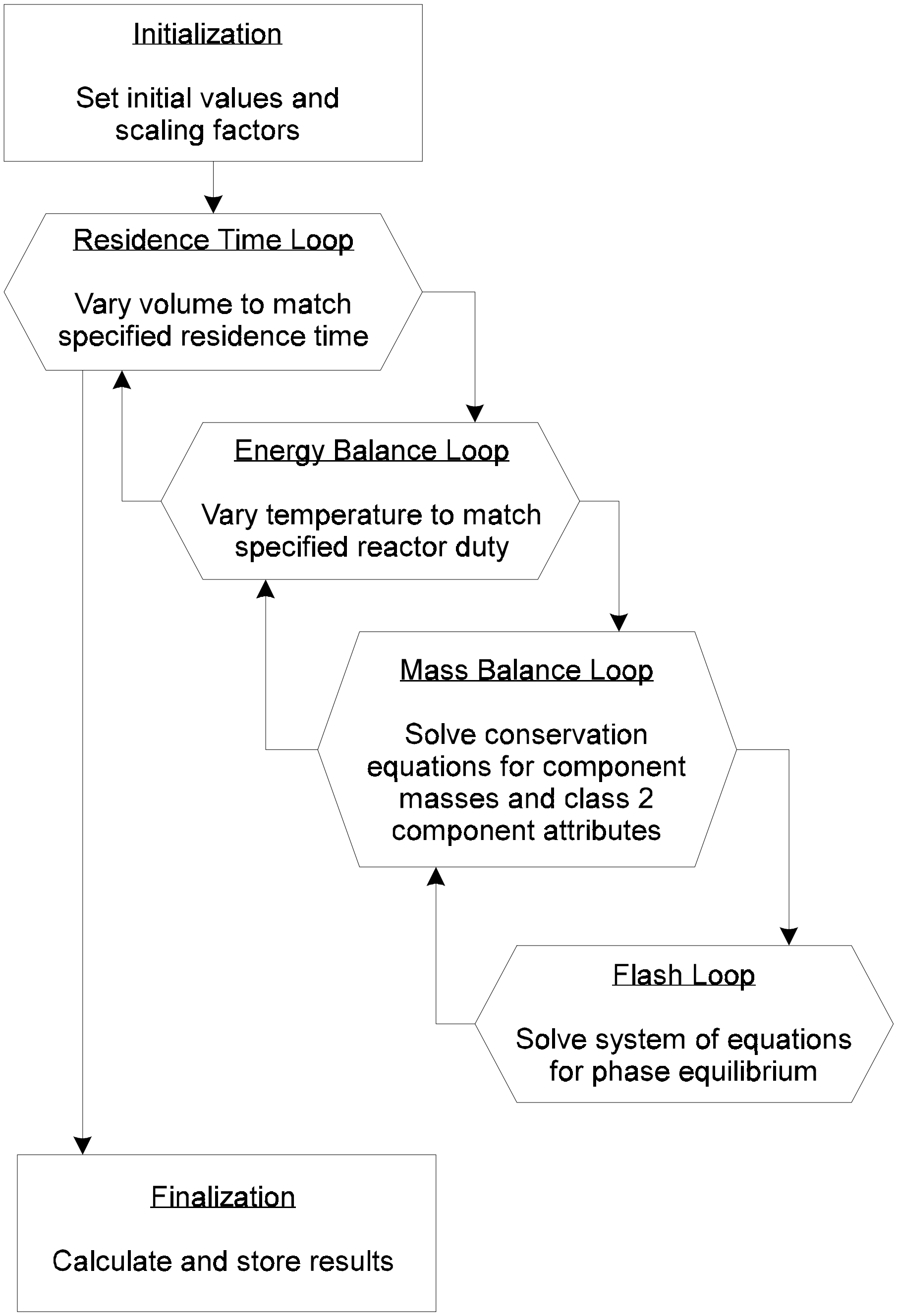


Figure 5.11 *RCSTR Solution Algorithm*

The outer-most loop involves the volume and residence time of the CSTR. Table 5.2 shows that there are many options for specifying the characteristic volume of a multiphase CSTR. When residence time is specified instead of volume, the RCSTR model adjusts the volume to satisfy the residence time specification.

Convergence problems in the residence time loop can be alleviated by providing initial volume estimates in the ESTIMATES form. If convergence problems persist, then the maximum volume step size (Max-Vstep) should be reduced. If the key phase residence time is specified, then the RCSTR model uses the specified reactor volume as an upper limit for the key phase volume.

Table 5.2 Calculation of Volumes and Residence Times in RCSTR

Specifications: Total reactor volume (Vol)		
$\theta_R = \dfrac{V_R}{F\sum_j v_j f_j}$	$V_j = \dfrac{f_j v_j}{\sum_k f_k v_k} V_R$	$\theta_j = \dfrac{V_j}{F f_j v_j}$
Specifications: Total residence time (Res-time)		
$V_R = \theta_R F \sum_j v_j f_j$ **	$V_j = \dfrac{f_j v_j}{\sum_k f_k v_k} V_R$	$\theta_j = \dfrac{V_j}{F f_j v_j}$
Specifications: Total reactor volume (Vol), key phase volume (Ph-vol)		
$\theta_R = \dfrac{V_R}{F\sum_j v_j f_j}$	$V_j = specifed$ *	$\theta_j = \dfrac{V_j}{F f_j v_j}$
Specifications: Total reactor volume (Vol), key phase volume fraction (Ph-vol-frac)		
$\theta_R = \dfrac{V_R}{F\sum_j v_j f_j}$	$V_j = r_j V_R$	$\theta_j = \dfrac{V_j}{F f_j v_j}$
Specifications: Total reactor volume (Vol), key phase residence time (Ph-res-time)		
$\theta_R = \dfrac{V_R}{F\sum_j v_j f_j}$	$V_j = \theta_j F f_j v_j$ **	$\theta_j = specified$
Specifications: Total residence time (Res-Time), key phase volume fraction (Ph-vol-frac)		
$V_R = \theta_R F \sum_j v_j f_j$ **	$V_j = r_j V_R$	$\theta_j = \dfrac{V_j}{F f_j v_j}$

V_R = total reactor volume; V_j = volume of phase "j"; v_j = molar volume of phase "j";

r_j = volume fraction of phase "j"; θ_R = total residence time; θ_j = residence time of phase "j"

F = total molar flow rate at reactor outlet; f_j = molar fraction of phase "j"

* ***If more than one condensed phase is present, and the key phase is liquid, then the specified volume applies to the sum of the condensed phase volumes.***

** ***This equation is solved by trial-and-error technique.***

EB LOOP

The second loop is the energy balance conservation equation (EB-LOOP). In this loop, the reactor temperature is adjusted to match the specified reactor duty. If the temperature is specified instead of the duty, this loop is by-passed.

Since the reaction rates are very sensitive to temperature, large changes in the reactor temperature between energy-balance iterations (EB-ITER) may cause the mass-balance loop (MB-LOOP) to diverge. This problem can be solved by providing a good temperature estimate (T-EST) in the ESTIMATES form. If the problem persists, the maximum temperature step size (Max-Tstep) should be reduced (the default, 50°C, is rather large).

MB-LOOP

The next loop is the mass-balance loop (MB-LOOP). This loop uses a multivariable solver to converge the conservation equations for component mole flow and for the class two component attributes.

Two solvers are available: Broyden and Newton. The Broyden algorithm tends to be relatively fast, but it may be unstable if the number of components and attributes is large and the reaction rates are high. The Newton algorithm tends to be slower, but more stable for many classes of problems. The Newton algorithm calculates the response of each variable to each other variable by perturbing the variables one at a time by a very small amount. These perturbation steps require lots of simulation time, which makes each iteration of the Newton algorithm slow.

The number of mass-balance iterations (MB-Maxit) is defined on the convergence form. By default, the model allows 50 mass-balance iterations. This default is sufficient for the Newton algorithm, but is usually too small for the Broyden algorithm. For polymer reaction kinetics, the number of required mass-balance iterations may be as high as 500.

Using a Damping Factor

The stability of the Broyden algorithm can be adjusted using a damping factor (DAMP-FAC) defined on the “Convergence” form. Decreasing the damping factor decreases the step-size, resulting in a larger number of smaller, more stable steps. Thus, the maximum number of iterations should be increased as the damping factor is decreased.

The damping factor is sensitive on a log scale. If the Broyden algorithm appears unstable, try setting the damping factor to 0.5, 0.3, 0.1, 0.05 etc. Optimum damping factors for polymerization kinetics typically fall between 0.1 and 0.001.

The conservation equations have the form:

$$accumulation = input - output + Generation$$

For the component mole balance equations: $\frac{R_i}{S_i} = \frac{F_i^{in}}{S_i} - \frac{F_i^{out}}{S_i} + \frac{\sum_j G_{i,j} V_j}{S_i}$

For the class-2 component attributes equations: $\frac{R_i}{S_i} = \frac{A_i^{in}}{S_i} - \frac{A_i^{out}}{S_i} + \frac{\sum_j G'_{i,j} V_j}{S_i}$

Where:

R_i = residual value for equation *i*, kmol/sec

F_i^{in} = molar flow rate of component *i* into the reactor, kmol/sec

F_i^{out} = molar flow rate of component *i* out of the reactor, kmol/sec

$G_{i,j}$ = molar generation rate of component *i* in phase *j*, kmol/ m^3 sec

A_i^{in} = flow rate of attribute *i* into the reactor, kmol/sec or particle/sec

A_i^{out} = flow rate of attribute *i* out of the reactor, kmol/sec or particle/sec

$G'_{i,j}$ = generation rate of attribute *i* in phase *j*, kmol/ m^3 sec or particle/ m^3 sec

V_j = volume of phase *j* in the reactor

S_i = scaling factor for equation *i*

The mass-balance loop is converged when the maximum scaled residual of the conservation equations falls below the specified tolerance (MB-TOL):

$$\text{Maximum error} = \text{MAX}_\text{i}\left(\frac{R_i}{S_i}\right) < \text{MB} - \text{TOL}$$

A secondary criteria is the root-mean-square scaled error, or RMS error:

$$\text{RMS Error} = \sqrt{\frac{1}{N_i}\sum_i\left(\frac{R_i}{S_i}\right)^2}$$

The CSTR mass-balance algorithm iterates until the maximum error falls below the specified mass-balance tolerance *or* the maximum number of mass-balance iterations is reached. If the maximum error criteria is reached, and the RMS error is decreasing by a factor of ten on each iteration, the CSTR model continues to iterate until the RMS error reaches the specified function tolerance (FUNC-TOL). This allows the model to reach very tight convergence tolerances when the convergence behavior is good.

Scaling Factors

The scaling factors play an important role in the convergence behavior of the model. If the scaling factors are large, and the variables are small, then the model will be loosely converged. If the scaling factors are small, and the variables are large, the convergence criteria will be unacceptably tight, and the model will not converge. There are two scaling options in the RCSTR model as shown in the Table 5.3.

By default, the component scaling option is used. With this option, the minimum scaling factors depend on the value of the “TRACE” parameter. The trace parameter is effectively a minimum mole fraction. For components with concentrations below the trace level, the scaling factors are set to a minimum value.

The default scaling factors for component attributes are defined as constants in an Aspen Plus Table Building System (TBS) data file, “COMPATTR.DAT”. Although the default scaling factors are set to appropriate values for most classes of reaction kinetics, the optimal values for a particular type of kinetics may be different than the defaults. The default values can be adjusted by editing the attribute TBS table and running the TBS update procedure.

The scaling factors should make the scaled values as close to unity as possible. For this reason, the scale factors are set to the predicted values from previously converged passes through the RCSTR block. On the first pass through the flowsheet, the scaling factors will be set to the estimated value for the variable. Thus, component flow and component attribute estimates can be used to set the initial scale factors.

The scaling factors for component attribute values are normalized with the total mole flow rate of the outlet stream. This keeps the scaling factors reasonable for models of any type of process, from bench scale to production scale units.

Table 5.3 RCSTR Scaling Factors

Variable Type	Component Scaling	Substream Scaling
Enthalpy	Estimated outlet stream enthalpy	10^5
Component Mole Flows	The larger of: • Estimated component mole flow in outlet stream (or retention value if available) • Trace x total mole flow rate of the outlet	Total estimated outlet stream mole flow rate
Class 2 Attributes	The larger of: • Estimated attribute value in outlet stream (or retention value if available) • (Attribute scaling factor from the TBS table) x (Estimated mole flow rate of the attributed component) • (Trace) x (Total estimated outlet mole flow rate) x (Attribute scaling factor from the TBS table)	Attribute scaling factor from the TBS table

The inner-most loop is the phase equilibrium loop, or flash equations. For this reason, it is essential to have accurate physical properties over the entire range of temperatures and pressures found in the process.

The flash calculations start from retention values once the mass-balance error falls below the retention threshold (Ret-Thresh) specified in the convergence form. If the retention threshold is set very high, then the flash may fail, resulting in step-size cuts in the mass balance loop. If the retention threshold is reduced, the reactor calculations may require more time. For most simulation problems, setting the retention threshold to 1×10^{10} results in fast flash convergence without errors. If errors occur, try using the default value, 1×10^{5} . If errors persist, the most likely cause is a physical property problem.

Integration Initialization

The convergence behavior of the RCSTR model depends on how the model is initialized. There are two initialization options for the RCSTR model. By default, the RCSTR model initialized the solver as described below. Optionally, the RCSTR model can be initialized using an integration approach.

In the integration algorithm, the RCSTR is treated as a dynamic stirred-tank reactor. The conservation equations are numerically integrated from an initial condition to the steady-state condition. The initial compositions in the reactor are set equal to those in the feed stream.

If temperature is specified in the reactor, then the temperature profile during initialization can be ramped from the feed stream temperature to the specified temperature over the interval of several residence times. If duty is specified, then the duty can be ramped from adiabatic conditions to the specified duty. Ramping allows the reactor to “cold-start” for improved integration performance.

The numerical integration carries forward until the residual terms (accumulation terms) drop below the specified mass-balance tolerance. At this point, the model enters the solver and continues until the model converges.

Note that initial guesses for component flow rates and component attributes should not be provided when using the integration initialization option.

Solver Initialization

By default, the solver algorithm initializes itself based on previously-saved results (retention). This saves time if the RCSTR block is inside a flowsheet recycle loop, where the block will be run several times in succession. This also saves time if the block is inside a sensitivity, optimization, design-spec, or data-fit study. The retention values are ignored if the “RESTART” switch in the “Block-Options” form is set to “YES”.

If the “RESTART” flag is “YES”, or during the first pass through the model, the model will initialize according to user-specified estimates. Estimates can be provided for the reactor volume, phase volume, reactor temperature, component flow rates, and component attribute values. The component attribute estimates can be specified using class-2 or class-0 attribute values.

If estimates are not provided, then the model assumes that the outlet stream conditions equal the inlet stream conditions.

Troubleshooting Convergence Problems

To diagnose RCSTR convergence problems, set the terminal reporting level to “7” in the Block-Options form. This causes the RCSTR model to report the residence time iterations (RT-ITER), energy balance iterations (EB-ITER), and mass-balance iterations (MB-ITER) to the control panel. In addition, the model reports the maximum and root-mean-square errors for each loop.

The history level controls the diagnostic messages written to the history file (.HIS file). The maximum mass-balance error is reported at level 5. At level 6, the model reports the value of each reacting component flow rate and each component attribute. At level 7, the model reports values and rates of change (reaction rates) for components and attributes.

First, simplify the problem by specifying temperature and volume instead of duty and residence time. If convergence problems persist, they must be related to the mass-balance loop, the reaction kinetic model or rate constants, or the underlying physical property calculations.

Numerical integration is much more reliable than trial-and-error solvers. If the RCSTR mass-balance fails to converge, try running the same kinetics in an RPlug model. If possible, set the phase criteria “liquid-only” to eliminate physical property problems from the list of possible sources of error. If the RPlug model cannot converge with the specified kinetics, then the problem is almost certainly related to reaction kinetics.

Possible sources of error in the reaction kinetics include:

- Errors in the molecular weight of a product or reactant
- Errors in the specified stoichiometry of a reaction (mass balance is violated)
- Unreasonable rate constants, especially activation energies (verify the units)
- Reactions with zeroth-order reactants which are not present
- Unreasonable concentrations of catalysts or inhibitors (put the feed stream in a flash block and verify that the concentrations in the reacting phase make sense).
- Errors in user-written Fortran subroutines.

If these sources of error are eliminated, and convergence problems persist, try simplifying the model by removing unnecessary side reactions or trace components from the model. Convergence is much easier if the number of equations is reduced, the speed of most convergence algorithms varies with the cube of the number of equations (the number of equations equals the number of reacting components plus the number of class-2 component attribute elements).

Table 5.4 summarizes solutions for some common problems encountered when using RCSTR.

Table 5.4 Troubleshooting RCSTR Convergence Problems

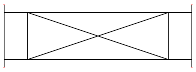
Problem	Solution
Initial flash failure	This is usually a physical property problem. • Check the heat of formation (DHFORM) and ideal gas heat capacity parameters (CPIG) of the polymer and oligomer components. • If supercritical components are present, consider treating them as Henry's law components • Verify that the property method you are using is appropriate for the specified temperature and pressure • Verify the specified phases are consistent with the specified temperature and pressure • Verify the specified local and global flash tolerance
Mass balance not converged in maximum number of iterations, but the error is decreasing from one iteration to the next	Increase the maximum number of iterations. If more than 500 iterations are required for the Broyden algorithm, try adjusting the damping factor. Provide better initial guesses.
Mass balance not converged in maximum number of iterations, the maximum error is varying erratically between iterations, and the history file shows reasonable rates	If using the Broyden algorithm, try decreasing the damping factor by logarithmic steps (0.5, 0.3, 0.1…0.0001) until the problem converges. If the problem persists, try using the Newton algorithm. Provide better initial guesses.
Mass balance not converged in maximum number of iterations, the maximum error is varying erratically between iterations, and the history file shows some reaction rates or attribute rates are much larger than others (or are erratic between iterations)	Check the specified rate constants in the kinetic models, especially activation energies. Verify the units of the activation energies. Verify flow rates of catalysts and initiators in the feed streams to the reactor. If using user kinetics, check your subroutine for errors. Verify the reactor volume (residence time).
Mass balance not converged in maximum number of iterations. Reaction rates are very high, as expected.	Try using the Newton algorithm with good initial guesses. If this fails, delete the initial guesses and try using the integration initialization.
Mass balance not converged in maximum number of iterations. Some reacting components (catalysts, initiators) are present in very small quantities.	Try adjusting the "trace" parameter in order-of-magnitude steps from the default (1×10^{-3}) down to the concentration of the trace components. If this fails, reset trace to the default value and try integration initialization.

continued

Table 5.4 Troubleshooting RCSTR Convergence Problems (cont.)

Problem	Solution
Energy balance loop does not converge, or mass-balance loop does not converge after the second energy balance loop iteration, or temperature step-size cutting (T-CUT) iterations appear in the diagnostic messages	Verify that the reactor converges with the temperature specified. If not, see items listed above, otherwise, provide a better temperature estimate (T-Est). If the problem persists, try adjusting the maximum temperature step-size (Max-Tstep) from 50°C to 10°C or even 5°C.
Residence time loop does not converge, or mass-balance loop does not converge after the second residence-time loop.	Verify that the reactor converges with the residence time specified. If not, see items listed above, otherwise, provide better volume estimates. If the problem persists, try adjusting the maximum volume step-size (Max-Vstep)
Flash failures appear during the mass-balance loop; the step-size cutting (X-CUT) diagnostic message appears.	This may be a physical property problem; it may reflect overly-tight flash tolerances; or the flash may be unstable when starting from retention values. Try decreasing the flash retention threshold from 1×10^5 to 1×10^2. If the problem persists, loosen the local and global flash tolerance levels or increase the maximum number of flash iterations.
Reactor converges but an error message says that the mass-balance does not close	Check the molecular weights of each reactant and product. Verify that reaction stoichiometry is correct.

RPlug



The RPlug model represents an ideal plug-flow reactor with one or more phases. The model assumes perfect radial mixing within and between the phases, phase equilibrium, and no-slip conditions between the phases (e.g., the phases all have the same residence time). Dead zones, back-mixing, and other types of non-ideal plug-flow behavior can be represented using RPlug in combination with other blocks.

Temperature

RPlug allows many options for specifying the reactor duty or temperature, as summarized in Table 5.5.

Table 5.5 Temperature Specification Options in RPlug

Type	Specifications	Calculations
ADIABATIC	None	Temperature is calculated at each axial position based on the enthalpy balance.
T-SPEC	Process stream temperature as a function of axial position (linear interpolation between the points)	Duty is integrated along the length of the reactor. The model reports the net duty across the reactor
T-COOL-SPEC • heat transfer routine optional	Heat media stream temperature (assumed constant along length of reactor). Overall heat-transfer coefficient. Area is determined from length, diameter , and number of tubes: A=N✍DL	Duty is integrated along the length of the reactor. The temperature of the process stream is determined from the energy balance. The model reports the net duty across the reactor
CO-COOL • coolant stream required • heat transfer routine optional	Heat media (coolant) stream temperature, composition, and flow rate. Overall heat-transfer coefficient. Area is determined from length, diameter, and number of tubes: A=N✍DL.	Duty is integrated along the length of the reactor and is reported as a net value. The temperature of the process and coolant streams are determined from the energy balance.
COUNTER-COOL • coolant stream required • heat transfer routine optional	Heat media (coolant) composition, flow rate and effluent temperature. Overall heat-transfer coefficient. Area is determined from length, diameter, and number of tubes: A=N✍DL.	Duty is integrated along the length of the reactor and is reported as a net value. The temperature of the process and coolant streams are determined from the energy balance. A design specification may be used to fit coolant feed temperature by adjusting coolant outlet temperature.

RPlug allows one process stream and one heat media stream. Reactions can occur only in the process stream. Heat transfer calculations are carried out between the process stream and the heat media stream. The heat media stream represents a coolant stream or a heating stream and the heat media stream flows co- or counter-current to the process stream.

If a heat media stream is not specified, the model assumes a constant heat media temperature and solve for the process fluid temperature.

The heat transfer rate or heat transfer coefficient value is calculated as a function of axial position, stream conditions, etc., by a user-written Fortran subroutine. This feature is used to develop rigorous models non-reactive heat exchangers

Pressure

The pressure at the reactor entry can be specified as an absolute value or as a pressure drop relative to the feed stream. In Aspen Plus, pressure drops are expressed as non-positive pressure specifications given in absolute pressure units.

The pressure drop across the reactor can be specified as a constant or calculated in a user-written Fortran subroutine. If the pressure drop is specified, the model assumes it is linear along the length of the reactor.

Residence Time

The RPlug model assumes a cylindrical geometry. The length, diameter, and number of tubes are specified. The process fluid is assumed to move through the tubes, and the coolant is assumed to flow on the outside of the tubes.

The total reactor volume cannot be specified, but the aspect ratio (length/diameter) has no influence on the model predictions. Thus, the diameter can be set to 1.12838 units, which corresponds to an area of 1.0000 $units^2$. With this area, the length in units and volume in $units^3$ have the same numerical value, thus the specified length is equal to the volume.

The phase volumes cannot be specified independently. Instead, the RPlug model assumes that the phases move through the reactor without slipping past each other. This assumption is valid for situations where one phase is dispersed as droplets or bubbles in a second, continuous phase, such as dew in a vapor phase or small gas bubbles in a liquid phase. This assumption is not valid for multiphase plug flow reactors with controlled levels.

With this assumption in place, the reactor residence time is equal to the residence time of each phase present in the reactor. The residence time is calculated by numerical integration.

One work-around for the no-slip assumption is to write a user kinetic subroutine (or a step-growth mass-transfer routine) which calls the flash model directly. Then, specify the reactor as liquid-only and set the diameter to the hydraulic diameter of the reactor.

Calculating Residence Time

Equation to Calculate Residence Time in RPlug:

$$\theta = \frac{\pi D^2 N}{4}\int_{z=0}^{z=L}\frac{dz}{F_z\sum_j f_{j,z}v_{j,z}}$$

Where:

🗎 = reactor residence time

D = tube diameter

N = number of tubes

z = axial position in reactor of length L

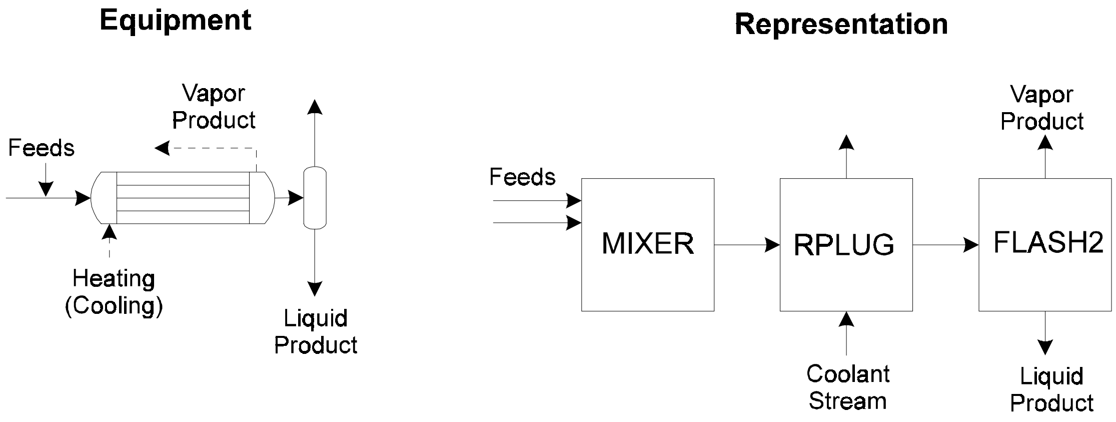
F_z = total molar flow rate of process stream at axial location z

$f_{j,z}$ = molar fraction of phase j at axial location z

$v_{j,z}$ = molar volume of phase j at axial location z

Multiphase Reactors

The RPlug models have one process fluid outlet stream which contains all of the phases exiting the reactor. As shown in Figure 5.12, a flash block is used in conjunction with these blocks to split the liquid and vapor phases from the mixed outlet stream of the reactor. In this application, it is good practice to specify PRES=0 (no pressure drop) and DUTY=0 in the flash block to ensure that the phase split occurs at conditions which are consistent with the reactor outlet. Another option is to specify temperature and to use a transfer block to copy the RPlug outlet stream temperature to the flash specifications.



Specify DUTY=0, PRESSURE=0 in the FLASH2 block to keep the flash results consistent with the RCSTR model or use a TRANSFER block to copy the temperature from the RPLUG process outlet stream to the FLASH2 block temperature.

Figure 5.12 *Representation of Two-Phase Plug-Flow Reactors*

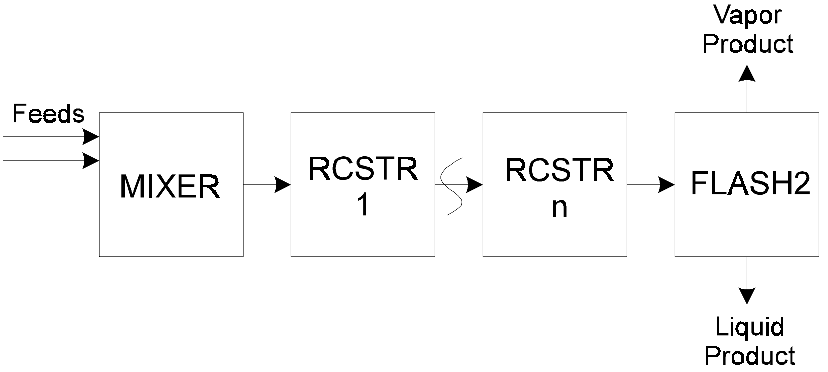
Reactors with Non-Ideal Mixing

Back-mixed plug flow reactors can be modeled using a recycle stream or by breaking the reactor down into a series of RCSTR blocks. The recycle-stream approach has the advantage of RPlug's profile-based input and output plotting, but it requires a flowsheet convergence loop which may be difficult to converge, especially if the circulation ratio is large. The series-of-CSTRs approach does not require recycle loop convergence, but the results are not as easily interpreted as the RPlug model.

Reactors with dead zones can be represented using parallel plug-flow reactors, as shown in Figure 5.13. The dead zone is represented by a plug-flow reactor with a large residence time. The active zone is represented as a plug-flow reactor with a shorter residence time. The volumes of the two reactors sum to the total volume of the real reactor. This approach assumes the dead zone reaches steady state.

As always, the simplest model which agrees with process data is the best choice.

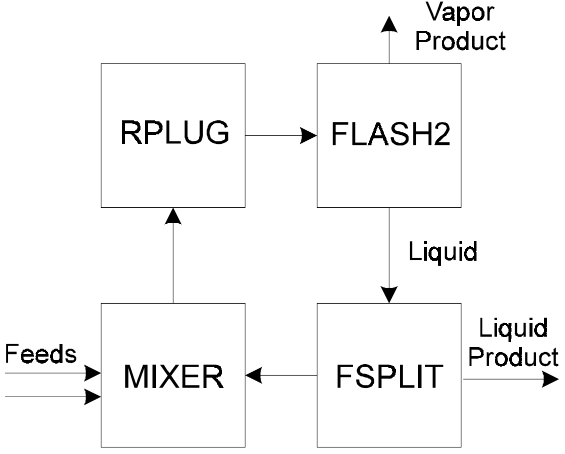
Series of CSTRs Approach



The degree of back-mixing is characterized by "n", the number of CSTRs. The volume of each CSTR block is set to the total reactor volume divided by the number of stages. This approach is quick and easy, but the features of RPLUG, such as rigorous heat transfer, cannot be applied.

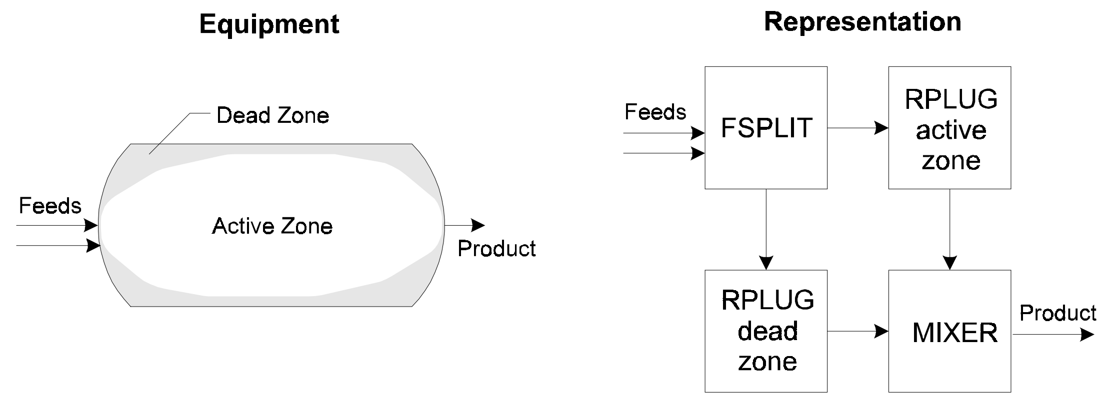
As the number of CSTRs increases, the reactor behaves more like a plug flow reactor, as the number decreases, the reactor behaves more like a CSTR.

Recycle Stream Approach



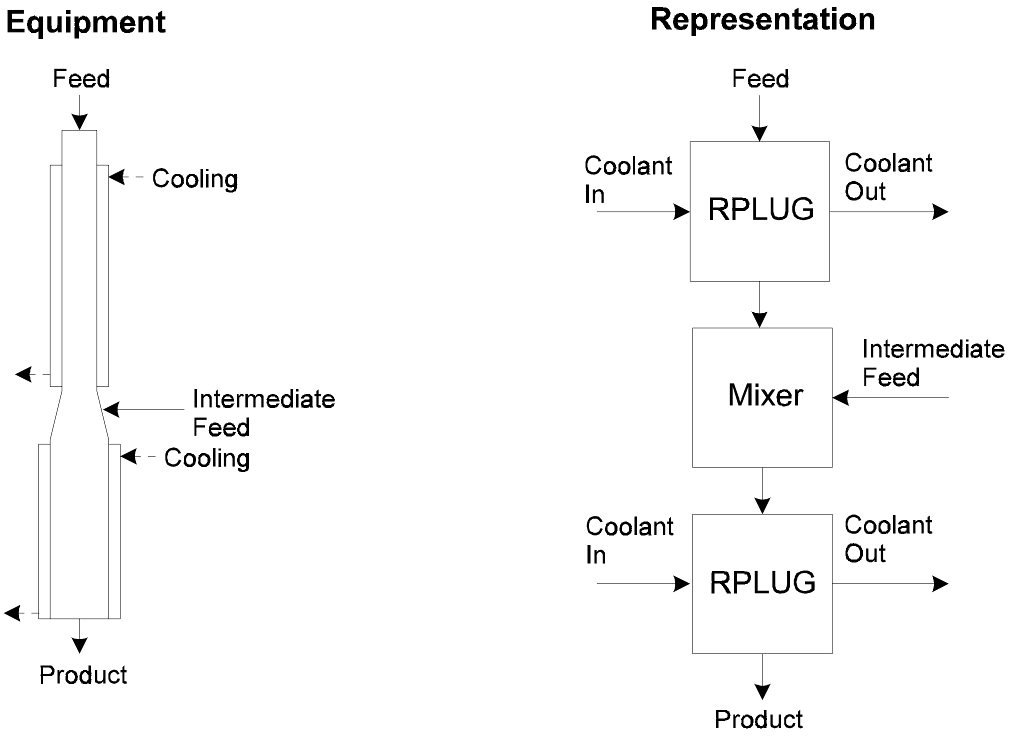
The degree of back-mixing is characterized by the ratio of the circulation rate to the product rate. As this circulation ratio approaches zero, the reactor approaches a plug-flow reactor. As the circulation ratio approaches unity, the reactor approaches a CSTR (but, it will be very difficult to converge the flowsheet). Give a good estimate for the recycle flow rate and composition to enhance the flowsheet convergence.

Figure 5.13 *Two Methods for Representing Back-Mixing in Two-Phase Plug Flow Reactors*



The dead zone is characterized by its residence time, which is determined from the specified volume in the plug flow reactor representing the dead zone and by the flow rate feeding the dead zone. The temperature of the dead zone material can be set higher than the temperature of the active zone material to account for higher temperatures near the walls of the vessel.

Figure 5.14 *Methods for Representing Plug Flow Reactors with Dead Zones*



Some reactors have injection ports part way along the axis. RPLUG does not allow intermediate feeds. Instead, the real reactor must be broken down into sections, with each section treated as a plug flow reactor. The same technique can be used to model reactors with multiple heating or cooling zones.

Figure 5.15 *Representation of Reactors with Injection Ports*

Troubleshooting Convergence Problems

To diagnose numerical problems in RPlug, set the terminal reporting level to “7” in the Block-Options form. With this setting, the RPlug block will report the normalized axial location, residence time (in seconds), pressure (in Pascal), temperature (in K), and vapor molar fraction at each converged step.

The history level controls the diagnostic messages written to the history file (.HIS file). The maximum mass-balance error is reported at level 5. At level 6, the model reports reacting component flow rates and component attribute values. At level 7, the model also reports the rates of change of these variables.

First, simplify the problem by specifying temperature instead of duty or heat-transfer parameters (coolant temperature, U, or coolant stream). Specify the reactor as “liquid-only”. This will eliminate many possible sources of error and help focus the problem on kinetics and integration parameters.

Scaling Factors

RPlug uses Gear’s variable-step-size algorithm to numerically integrate the mass, energy, and attribute conservation equations along the axial dimension of the reactor. At each axial step, the conservation equations are solved by a trial-and-error technique.

Like RCSTR, RPlug solves the conservation equations using scaling factors to normalize the variables. The values of these scaling factors can have a strong influence on the speed and reliability of the integration.

The Gear integrator in Aspen Plus offers two error scaling options (ERR-METHOD in RPlug), summarized in Table 5.6. The static scaling method uses a constant set of scaling factors throughout the reactor. The dynamic scaling method updates the scaling factors based on the previously converged step. Both scaling methods conform to minimum scaling rules which are determined by a parameter called “cutoff”, which is analogous to “trace” in RCSTR.

Table 5.6 RPlug Scaling Factors

Variable Type	**Static Scaling**	**Dynamic Scaling**
Enthalpy	10^5 (SI units)	10^5 (SI units)
Component Mole Flows	The scaling factor at z = 0 to 1.0 is set to the larger of: • Component mole flow in inlet stream • Cutoff x total mole flow rate of inlet Scaling factors are held constant	The scaling factor at z = z + ✆z is set to the larger of: • Component mole flow at z • Cutoff x total mole flow rate at z Scaling factors are updated at each step
Class 2 Attributes	The scaling factor at z = 0 to 1.0 is set to the larger of: • Attribute value in inlet stream • (Attribute scaling factor from the TBS table) x (mole flow rate of the attributed component in the inlet) • (Cutoff) x (total mole flow rate at the inlet) x (Attribute scaling factor from the TBS table) Scaling factors are held constant	The scaling factor at z = z + ✆z is set to the larger of: • Attribute value at z • (Attribute scaling factor from the TBS table) x (mole flow rate of the attributed component in the inlet) • (Cutoff) x (total mole flow rate at the inlet) x (Attribute scaling factor from the TBS table) Scaling factors are updated at each step

The static scaling method may result in faster integration for many types of problems, but there are potential numerical problems when using this method. Consider an irreversible reaction "A→ B" in a plug-flow reactor in which component "B" is not present in the feed. The scaling factor for component "A" will be set very large and the scaling factor for "B" will be set to the minimum scaling factor. This will result in relatively loose tolerance for the mass balance in "A" and tight tolerance for the mass balance in "B". Further, as the reaction approaches completion the component "B" will have a large flow rate but a small scaling factor. This makes the conservation equation for "B" difficult to solve, which will result in small integration steps.

Consider the same situation with dynamic scaling. The initial scaling factors are the same as the static case. With each new step, however, the scaling factors are updated to the variable values from the previous step (Figure 5. 16). This keeps the scaled variables close to one throughout the integration.

One pitfall of dynamic scaling, however, occurs when a variable value decreases and approaches zero. As the value and the scaling factor get progressively smaller, small absolute errors become large scaled errors. This also makes the solution difficult, and leads to small steps in the integrator. This problem can be controlled by setting the minimum scaling factor (cutoff in the convergence form) to a reasonable value. The default value, 10^{-10} is much too small for most problems. A value of 10^{-5} is reasonable for most situations, and will result in better model performance.

In general, the dynamic scaling method results in tighter convergence, but it requires more simulation time than the static scaling method. This does not apply to every case, however, and it may also depend on the solver algorithm. It is a good idea to experiment with these parameters to find the most reliable convergence strategy for each reactor in each model. When component attributes are present, as in polymerization kinetics, dynamic scaling is used by default.

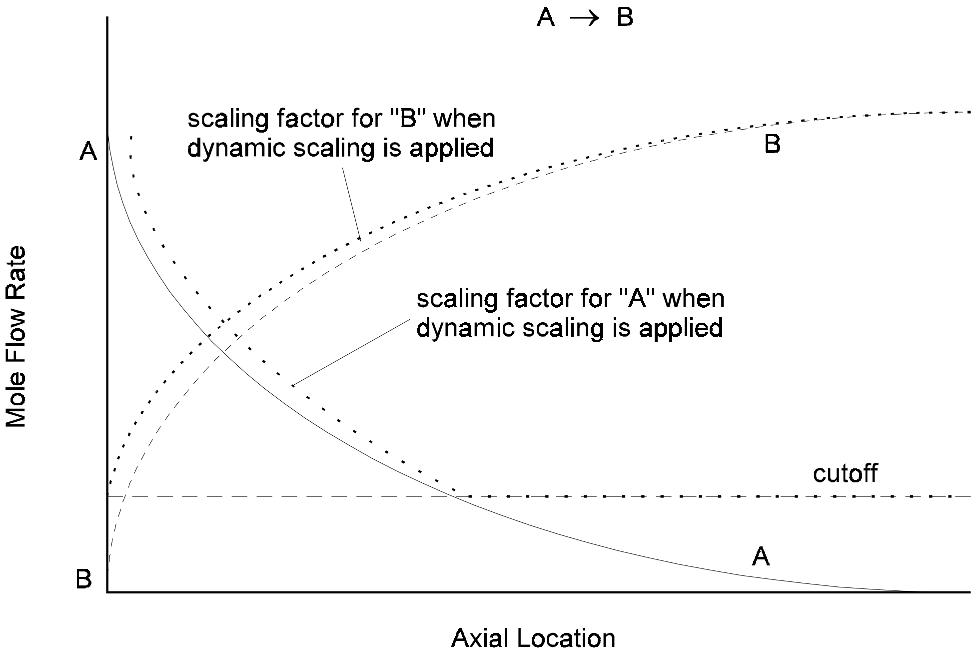


Figure 5.16 *Dynamic Scaling Factors in Integration*

Solver Method

At each step during the integration, the energy, mass, and attribute conservation equations are solved by trial-and-error. One of the two “corrector” algorithms, direct substitution or Newton’s method, can be selected. The Newton algorithm perturbs each variable to determine the slope, resulting in a smaller number or larger steps compared to the Direct algorithm. Since the perturbation passes require some time, it is difficult to predict if the Newton’s method or the Direct method is best for a given problem. In general, the Newton’s method appears to give the best performance with polymerization kinetics, but it is a good idea to try using each algorithm with both dynamic and static scaling to determine the best combination of convergence parameters for a particular problem.

The corrector tolerance is set as a ratio from the integration tolerance (Corr-Tol-Ratio). By default, the corrector tolerance is ten times tighter than the integration tolerance (the corrector tolerance ratio is 0.1). For some problems, especially those involving reactors with heat transfer calculations, the optimal corrector tolerance ratio may be higher than 0.1, but this ratio should not be set larger than 1.0. The flash tolerance should be tighter than the corrector tolerance. Otherwise, round-off errors in the flash calculations make the corrector tolerance difficult to achieve. The model always uses the smaller of the specified RPlug flash tolerance (in the convergence form) or the global flash tolerance.

Other Integration Parameters

By default, the initial step size in RPlug is set to one percent of the reactor length (Hinit=0.01). If the solver cannot converge the equations with this step size, it will cut the step size by a factor of ten. This process will repeat up to six times. If the solver still cannot converge, the reactor calculation fails with an error message "solver cannot converge with minimum step size".

Frequently, reaction rates or heat transfer rates are much faster near the entrance of the reactor than at the exit of a reactor due to step changes in temperature or pressure or due to the high concentrations of reactants at the inlet of the reactor. For these types of problems, the minimum step size may need to be reduced. For step-growth kinetics, try using an initial step size of 1×10^{-4}. Smaller initial step-sizes may be required for addition kinetics, especially if quasi-steady-state approximations are not applied.

The maximum number of integration steps defaults to 1000. For very "stiff" kinetics, e.g., kinetics with fast reaction rates involving trace components, the maximum number of steps may need to be increased, especially if the corrector is using direct substitution. If more than 5000 steps are required, try changing the corrector method, scaling method, or increase the cutoff level.

RPlug stores many types of results at regular intervals (printing points). The number of intervals defaults to ten, but the number of print points can be increased to get smoother plots. Since the integration steps do not necessarily correspond to the print points, the model uses polynomial interpolation to determine the results for a print point based on the steps before and after this point. If the integration step sizes are very large, the interpolation algorithm may give strange results, such as sine waves. This problem can be fixed by reducing the maximum step size (Max-StepSize) to a value smaller than the increments between print points (this forces the model to use linear interpolation). By default, the maximum step size is much larger than the reactor length.

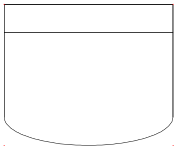
Summary

Table 5.7 summarizes common problems encountered when using the RPlug unit operation block.

Table 5.7 Troubleshooting RPlug Convergence Problems

Problem	Solution
Solver cannot converge for initial step	Try reducing the initial step size by orders of magnitude from the default (10^{-2}) to 10^{-8}. If the problem persists, try increasing the cutoff parameter from 10^{-10} to 10^{-5}. If the problem still persists, verify the values and units of the rate constants in the kinetic model. Verify the heat-transfer coefficient if applicable. Verify the temperature, composition, and flow rates of the feed streams. Check the history file diagnostics for unusually high reaction rates.
Integration error: non-negativity violation.	This problem is usually related to infeasible reaction kinetics. If using a user kinetic routine, verify the code, otherwise, a zeroth-order reactant may be completely consumed. Check the history file diagnostics; look for the component flow rate or attribute element which has a value of zero and a negative rate of change.
Integration error: maximum number of steps is reached	Try increasing the cutoff parameter from 10^{-10} to 10^{-5}. If the problem persists, try different combinations of scaling method and corrector method. As a last resort, try increasing the number of steps to 5000. If the problem still continues, search for errors in the kinetics; check the diagnostics for unreasonable reaction rates.
Integration error: corrector tolerance cannot be achieved	Tighten the flash tolerance to a value below the corrector tolerance. Loosen the integration tolerance to 1×10^{-3}. Increase the corrector tolerance ratio to 0.2, 0.3, 0.5. If the problem continues, verify the kinetics and heat-transfer parameters. Check history diagnostics.
Flash failures appear during the integration	This may be a physical property problem; it may reflect overly-tight flash tolerances, loosen the local and/or global flash tolerance levels or increase the maximum number of flash iterations.
Reactor converges but an error message says that the mass-balance does not close	Check the molecular weights of each reactant and product. Verify that reaction stoichiometry is correct.

RBatch



RBatch is a rigorous model for batch and semi-batch reactors. Any number of continuous feed streams can be specified in addition to a batch charge stream. Optionally, a vapor vent may be considered. The RBatch model does not have a vent condenser option; Aspen Custom Modeler is required to rigorously model batch polymerization reactors with vent condensers or overhead columns.

The RBatch model assumes feed and product accumulator holding tanks with continuos outlets. The accumulator concept provides a bridge between the continuous steady-state modeling environment in Aspen Plus and the inherently dynamic nature of batch reactors. The conversion between continuous streams and discreet charges and dynamic product accumulations is controlled by specified cycle times and continuous feed stream profiles specified in the reactor.

Temperature

RBatch allows many options for specifying the reactor duty or temperature, as summarized in Table 5.8. The temperature or duty can be specified as a time-varying function. Heat transfer can be accounted for by assuming a constant coolant temperature, heat transfer area, and heat transfer coefficient, or by writing a Fortran routine which returns the instantaneous duty at each time step.

If the temperature or temperature profile is specified, RBatch assumes a temperature controller. If the reactor is single-phase, or if the reactor volume is specified, the model assumes perfect temperature control, otherwise, the model uses a proportional-integral-derivative (PID) controller equation to represent a temperature controller:

$$Q_t = M_t^{reactor}\left[K\left(T_t - T_t^s\right) + \frac{K}{I}\int_0^t \left(T_t - T_t^s\right)dt + KD\frac{d\left(T_t - T_t^s\right)}{dt}\right]$$

Where:

Q_t = instantaneous heat duty, J/sec

$M_t^{reactor}$ = mass in reactor at time *t*, kg

T_t = temperature in reactor at time *t*, K

T_t^s = temperature setpoint at time *t*, K

t = time, sec

K = proportional gain (J/kg-K)

I = integral time, sec

D = derivative time, sec

By default, the proportional gain is 2500 J/kg-K, which results in very tight control at the expense of excessive simulation time. The speed of the model can be increased by reducing the gain (try a value of 25 J/kg-K).

Table 5.8 Temperature Specification Options in RBatch

Type	Specifications	Calculations
T-SPEC	Reactor temperature	The model reports the temperature profile, and the instantaneous and cumulative duty profiles.
T-PROFILE	Reactor temperature as a function of time. Linear interpolation is used to determine temperatures between specified points.	The model reports the temperature profile, and the instantaneous and cumulative duty profiles.
T-COOL-SPEC	Heat media stream temperature. Overall heat-transfer coefficient. Heat exchange surface area.	The temperature of the reactor is determined from the energy balance at each time step. The model reports the temperature profile, and the instantaneous and cumulative duty profiles.
DUTY-SPEC	Instantaneous heat duty (assumed constant for entire cycle). Set the duty to zero to model an adiabatic reactor.	The temperature of the reactor is determined from the energy balance at each time step. The model reports the temperature profile.
DUTY-PROFILE	Instantaneous heat duty as function of time. Linear interpolation is used to determine duty between specified points.	The temperature of the reactor is determined from the energy balance at each time step. The model reports the temperature profile, and the instantaneous and cumulative duty profiles.
USER-DUTY	Heat transfer subroutine name	The user routine returns the instantaneous heat duty at each time step. The temperature of the reactor is determined from the energy. The model reports the temperature profile, and the instantaneous and cumulative duty profiles..

Pressure

If the reactor volume is not specified, the RBatch model assumes the reactor operates as a closed system with a variable volume. The pressure at the reactor be specified as constant value or as a time-varying profile.

If the reactor volume is specified, and there is a vent stream attached to the reactor, the flow rate of the vent stream is determined from the specified pressure or pressure profile. The vent flow is positive when the calculated reactor pressure exceeds the specified reactor pressure.

If the reactor volume is specified, there is no vent stream attached to the reactor, and the pressure profile is not specified, then the pressure is determined by the temperature and molar volume of the material inside the reactor.

If the reactor volume is controlled, a pressure controller model can be linked to a continuous feed stream. The flow rate of the feed stream is adjusted to maintain a constant pressure inside the vessel. The continuous feed stream flow rate can decrease to zero, but it cannot reverse direction if the pressure exceeds the specified setpoint. The model uses a proportional-integral-derivative (PID) controller equation to represent the pressure controller:

$$F_t = \left[K\left(P_t - P_t^s\right) + \frac{K}{I}\int_0^t \left(P_t - P_t^s\right)dt + KD\frac{d\left(P_t - P_t^s\right)}{dt} \right]$$

Where:

F_t = instantaneous flow rate, kmol/sec

P_t = pressure in reactor at time t, Pa

P_t^s = pressure setpoint at time t, Pa

t = time, sec

K = proportional gain (kmol/sec)/Pa

I = integral time, sec

D = derivative time, sec

Reactor Volume

If the reactor pressure is not specified, then RBatch will predict the reactor pressure based on a specified reactor volume. The pressure will be manipulated by a trial-and-error algorithm to satisfy the specified volume.

If pressure and volume are both specified, you must either attach a vent stream to the reactor or attach a continuous make-up stream and pressure controller to the reactor.

Residence Time

The residence time of the reactor is controlled by user-specified stop criteria. You can specify whether RBatch should halt the reaction when the stop criterion variable is approached from above or below. If several stop criteria are specified, RBatch stops at the first stop criteria it reaches.

In addition to stop criteria, you must specify a maximum time for the reactor. This prevents runaway calculations in the event that none of the stop criteria are feasible.

The stop criteria may include one or more of the following:

- A maximum reaction time
- A maximum or minimum component mole or mass fraction in the reactor
- The amount of material (mass, moles, or volume) in the reactor or vent accumulator
- A maximum vent flow rate
- A maximum or minimum reactor temperature, pressure, or vapor fraction
- The value of a Prop-Set property (includes user Prop-Set properties or system properties such as viscosity, etc.)

Batch Operations

RBatch can represent batch or semi-batch reactors, depending on what streams are connected to it in the flowsheet. If a vent stream or time-varying continuous feed stream is connected to the RBatch block, then the model operates in semi-batch mode.

The batch reactor model is interfaced into the Aspen Plus continuous flow, steady-state modeling environment through the concept of holding tanks, as shown in Figure 5.17. The holding tanks convert the:

- Continuous batch charge stream to a discreet batch charge
- Final vent accumulator inventory to a continuous, time-averaged vent stream
- Final reactor inventory to a continuous, time-averaged reactor product stream

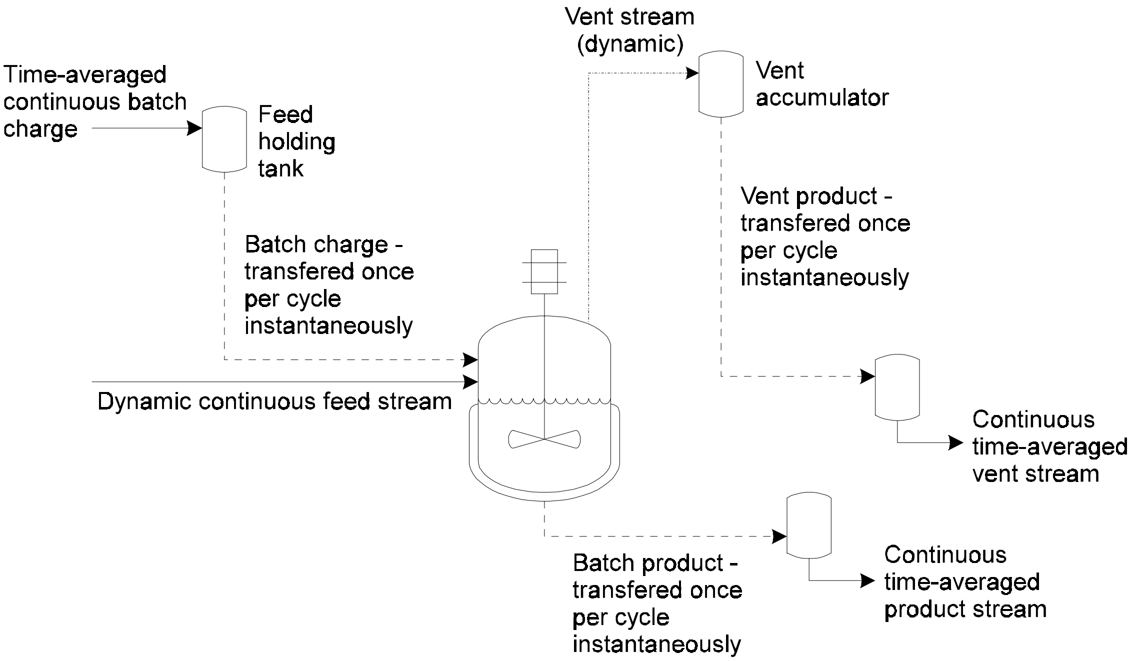


Figure 5.17 *RBatch Reactor Configuration*

Four types of streams are associated with RBatch:

- Continuous Batch Charge
- Time-Varying Continuous Feed
- Time-averaged Continuous Reactor Product
- Time-averaged Continuous Vent Product

Continuous Batch Charge: The material transferred to the reactor at the start of the cycle. The mass of the batch charge equals the flow rate of the batch charge stream, multiplied by the batch cycle time. The mass of the batch charge is equivalent to accumulating the batch charge stream in a holding tank during a reactor cycle. The contents of the batch charge holding tank are instantaneously transferred to the reactor at the start of each batch cycle.

Time-Varying Continuous Feed: Streams which are fed to the reactor over some discreet interval *during* the batch cycle. The composition, temperature, pressure, component attribute values, and time-averaged flow rate of the stream are specified in the flowsheet. The flow rate of the continuous feed streams can be specified as a constant value, a time-varying profile, or manipulated by the pressure controller model to meet a time-varying pressure setpoint.

Time-averaged Continuous Reactor Product: This stream is determined by dividing the final reactor inventory by the cycle time. This is analogous to instantaneously dumping the reactor contents to a large holding tank at the end of the cycle, and continuously draining the tank throughout each cycle.

Time-averaged Continuous Vent Product: This stream is determined by dividing the final vent accumulator inventory by the cycle time. During the batch cycle, the time-varying continuous vent stream is accumulated in the vent accumulator. The model assumes the vent accumulator contents are instantly drained to a large holding tank at the end of the cycle, and the holding tank contents are continuously removed throughout the cycle.

Cycle Time

RBatch is a dynamic batch reactor model which is interfaced into the Aspen Plus continuous steady-state modeling environment. The interface requires converting batch charges and accumulator inventories into continuous stream flow rates. The cycle time is used to convert the batch charge flow rate into the initial reactor inventory. The cycle time is also used to convert the vent accumulator inventory and the reactor inventory into vent and reactor product streams.

For example, assuming a reactor has a cycle time of two hours and that no continuous feed streams are specified, then:

- If the batch charge stream is set to 50 kg/hour, the initial reactor inventory is 100 kg.
- If at the end of the reaction cycle, the vent accumulator contains 30 kg of material, the time-averaged continuous vent stream flow rate is 15 kg/hr. The composition of the time-averaged vent stream will be the same as the final composition in the vent accumulator.
- The final reactor inventory will be 70 kg, and the time-averaged reactor product flow rate will be 35 kg/hr.

RBatch allows you to specify a feed time and down time instead of the cycle time. In this case, the time-averaged batch charge stream is divided by the feed time to calculate the initial batch inventory. The time-averaged product flow rates are based on the cycle time, which is calculated from the sum of the feed time, the down time, and the reaction time. This option is not recommended unless it is used to correct the mass balance for the influence of time-varying continuous feed streams.

Troubleshooting Convergence Problems

To diagnose numerical problems in RBatch, set the terminal reporting level to “7” in the Block-Options form. With this setting, RBatch reports the time (in seconds), pressure (in Pascal), temperature (in K), and vapor mole fraction at each converged integration step.

The history level controls the diagnostic messages written to the history file (.HIS file). The maximum mass-balance error is reported at level 5. At level 6, the model reports reacting component flow rates and component attribute values. At level 7, the model also reports the rates of change of these variables.

First, simplify the problem by specifying temperature instead of duty or heat-transfer parameters (coolant temperature, U, or heat transfer subroutine). Specify the reactor as “liquid-only”. Specify the reactor pressure, but not the reactor volume. This will eliminate many possible sources of error and help focus the problem on kinetics and integration parameters. Once the model works with these settings, then revert the settings to duty, volume, and so on, making sure the model converges with each new specification.

Scaling Factors

RBatch uses Gear’s variable-step-size algorithm to numerically integrate the mass, energy, and attribute conservation equations for the reactor and the mass-balance equations for the vent condenser (if applicable). At each time step, the conservation equations are solved by a trial-and-error technique.

RBatch solves the conservation equations using scaling factors to normalize the variables. The values of these scaling factors have a strong influence on the speed and reliability of the integration.

The Gear integrator in Aspen Plus offers two error scaling options (ERR-METHOD), summarized in Table 5.9. The static scaling method uses a constant set of scaling factors throughout the reactor. The dynamic scaling method updates the scaling factors based on the previously converged. The “cutoff” parameter is the minimum scaling factor used in dynamic scaling.

Table 5.9 RBatch Scaling Factors

Variable Type	Static Scaling	Dynamic Scaling
Enthalpy	10^5 (SI units)	10^5 (SI units)
Component Mass Inventory In Reactor and Vent Accumulator	The scaling factor for each component inventory equation is set to: 0.1 * (mass of batch charge stream) Scaling factors are held constant	The scaling factor at $t = t + \Delta t$ is set to the larger of: Component mass flow at t Cutoff x total mass flow rate at t Scaling factors are updated at each step
Class 2 Attribute Inventory in Reactor and Vent Accumulator	The scaling factor of each component attribute is set to: (Attribute scaling factor from the TBS table) x (cycle time) (this is the attribute inventory at time = 0) Scaling factors are held constant	The scaling factor at $t = t + \Delta t$ is set to the larger of: Attribute inventory at time = t Cutoff Scaling factors are updated at each step

The static scaling method may result in faster integration for many types of problems, but there are potential numerical problems when using this method. Consider an irreversible reaction “A→ B” in a plug-flow reactor in which component “B” is not present in the feed. The scaling factor for component “A” will be set very large and the scaling factor for “B” will be set to the minimum scaling factor. This will result in relatively loose tolerance for the mass balance in “A” and tight tolerance for the mass balance in “B”. Further, as the reaction approaches completion the component “B” has a large flow rate but a small scaling factor. This makes the conservation equation for “B” difficult to solve, which will result in small integration steps.

Consider the same situation with dynamic scaling. The initial scaling factors are the same as the static case. With each new step, however, the scaling factors are updated to the variable values from the previous step (Figure 5.18). This keeps the scaled variables close to unity throughout the integration.

One pitfall of dynamic scaling, however, occurs when a variable value decreases and approaches zero. As the value and the scaling factor get progressively smaller, small absolute errors become large scaled errors. This also makes the solution difficult, and leads to small steps in the integrator. This problem can be controlled by setting the minimum scaling factor (cutoff in the convergence form) to a reasonable value. The default value, 10^{-10} is much too small for most problems. A value of 10^{-5} is reasonable for most situations, and results in better model performance.

In general, the dynamic scaling method results in tighter convergence, but it requires more simulation time than the static scaling method. This does not apply to every case, however, and it may also depend on the solver algorithm. It is a good idea to experiment with these parameters to find the most reliable convergence strategy for each reactor in each model. When component attributes are present, as in polymerization kinetics, dynamic scaling is used by default.

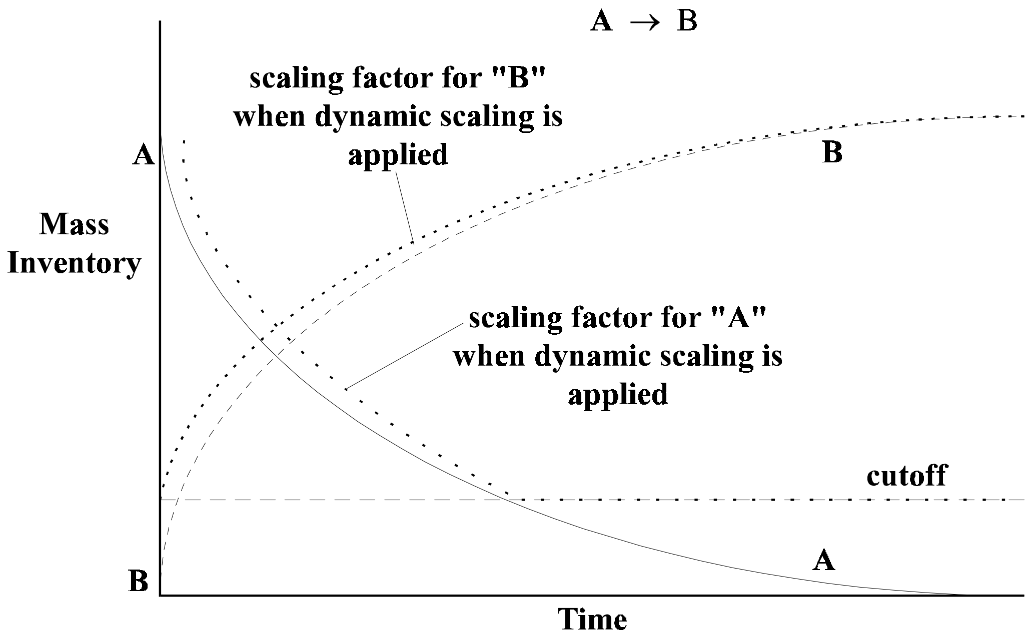


Figure 5.18 Dynamic Scaling Factors in Integration

Solver Method

At each step during the integration, the energy, mass, and attribute conservation equations are solved by trial-and-error. Two “corrector” algorithms, direct substitution and Newton’s method, can be selected. The Newton algorithm perturbs each variable to determine the slope, resulting in a smaller number or larger steps compared to the Direct algorithm. Since the perturbation passes require some time, it is difficult to predict if Newton’s method or the Direct method is best for a given problem. In general, Newton’s method appears to give the best performance with polymerization kinetics, but it is a good idea to try using each algorithm with both dynamic and static scaling to determine the best combination of convergence parameters for a particular problem.

The corrector tolerance is set as a ratio from the integration tolerance (Corr-Tol-Ratio). By default, the corrector tolerance is ten times tighter than the integration tolerance (the corrector tolerance ratio is 0.1). For some problems, especially those involving reactors with heat transfer calculations, the optimal corrector tolerance ratio may be higher than 0.1, but this ratio should not be set larger than 1.0. The flash tolerance should be tighter than the corrector tolerance. Otherwise, round-off errors in the flash calculations make the corrector tolerance difficult to achieve. The model always uses the smaller of the specified RPlug flash tolerance (in the convergence form) or the global flash tolerance.

Other Integration Parameters

By default, the initial step size in RBatch is set to one tenth of a second (Hinit=0.1). If the solver cannot converge the equations with this step size, it will cut the step size by a factor of ten. This process will repeat up to six times. If the solver still cannot converge, the reactor fails with an error message “solver cannot converge with minimum step size”.

Frequently, initial reaction rates or heat transfer rates are very fast, so the minimum step size may need to be reduced. For step-growth kinetics, the default value should be sufficient. Smaller initial step-sizes may be required for addition kinetics, especially if quasi-steady-state approximations are not applied.

The maximum number of integration steps defaults to 1000. For very “stiff” kinetics, e.g., kinetics with fast reaction rates involving trace components, the maximum number of steps may need to be increased, especially if the corrector is using direct substitution. If more than 5000 steps are required, try changing the corrector method, scaling method, or increase the cutoff level.

RBatch stores many types of results at regular intervals (printing points). The number of intervals depends on the reaction time. Since the integration steps do not necessarily correspond to the print points, the model uses polynomial interpolation to determine the results for a print point based on the steps before and after this point. If the integration step sizes are very large, the interpolation algorithm may give strange results, such as sine waves. This problem can be fixed by reducing the maximum step size (Max-StepSize) to a value smaller than the increments between print points (this forces the model to use linear interpolation). By default, no maximum step size is enforced.

RBatch has the option to stop exactly at print points and vent accumulator points instead of interpolating these points. When the “exact” option is set to “yes”, the model adjusts the integration steps to exactly match these points. This requires extra steps in the integrator which may slow down the model, but it results in more accurate simulations.

Summary

Table 5.10 below summarizes common problems encountered when using RBatch.

Table 5.10 Troubleshooting RBatch Convergence Problems

Problem	Solution
Solver cannot converge for initial step	Try reducing the initial step size by orders of magnitude from the default (10^{-1}) to 10^{-8}. If the problem persists, try increasing the cutoff parameter from 10^{-10} to 10^{-5}. If the problem still persists, verify the values and units of the rate constants in the kinetic model. Verify the heat-transfer coefficient if applicable. Verify the temperature, composition, and flow rates of the feed streams. Check the history file diagnostics for unusually high reaction rates.
Integration error: non-negativity violation.	This problem is usually related to infeasible reaction kinetics. If using a user kinetic routine, verify the code, otherwise, a zeroth-order reactant may be completely consumed. Check the history file diagnostics; look for the component flow rate or attribute element which has a value of zero and a negative rate of change.
Integration error: maximum number of steps is reached	Try increasing the cutoff parameter from 10^{-10} to 10^{-5}. If the problem persists, try different combinations of scaling method and corrector method. As a last resort, try increasing the number of steps to 5000. If the problem still continues, search for errors in the kinetics; check the diagnostics for unreasonable reaction rates.
Integration error: corrector tolerance cannot be achieved	Tighten the flash tolerance to a value below the corrector tolerance. Loosen the integration tolerance to 1×10^{-3}. Increase the corrector tolerance ratio to 0.2, 0.3, 0.5. If the problem continues, verify the kinetics and heat-transfer parameters. Check history diagnostics.
Flash failures appear during the integration	This may be a physical property problem; it may reflect overly-tight flash tolerances, loosen the local and/or global flash tolerance levels or increase the maximum number of flash iterations.
Reactor converges but an error message says that the mass-balance does not close	Set the cycle time instead of the feed time. Check the molecular weights of each reactant and product. Verify that reaction stoichiometry is correct.

TREATMENT OF COMPONENT ATTRIBUTES IN UNIT OPERATION MODELS

As described in previous chapters, Polymers Plus includes two classes of component attributes. Class-2 attributes are “primary conserved quantities” and always have flow-type units (attribute value / unit time). These attributes include the zeroth moment of the polymer (polymer molecule flow rate), the segment flow rates, etc. Class-0 attributes are secondary quantities which can be derived from the primary quantities.

The class-2 attributes follow flow-based mixing rules. In other words, if two streams are mixed, the product stream class-2 attributes are equal to the sum of the feed stream class-2 attributes. These mixing rules apply to each unit operation which allows multiple feeds of the same type (for example, multiple process fluid feeds). In the distillation models, these mixing rules apply on a tray-by-tray basis (e.g., if two or more feed streams enter the same tray).

The blocks with more than one outlet (Flash2, Flash3, Sep, etc.) assume that the class 2 polymer attributes split according to mass mixing rules. For example, if 90% of the mass of the polymer flows to the liquid phase, then 90% of the polymer molecules also flow with the liquid phase. This approach is identical to assuming that the properties of the polymer, such as the molecular weight distribution, are not fractionated in any way; instead, the molecular weight distribution of each polymer component in each of the product phases is identical to that of the polymer in the feed stream. Table 5.11 summarizes the attribute handling for the different models.

Table 5.11 Attribute Handling

Basic Unit Operation Models	
Block	**Component Attribute Handling**
Dupl	All attributes in feed stream are copied to each outlet stream.
FSplit SSplit Sep Sep2	Class 2 attributes divide in proportion to flow rate of attributed component. Class 0 attributes are recalculated for each outlet stream. Equation to calculate outlet stream attributes: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed, out = outlet) A = class-2 component attribute value (in = mixed feed, out = outlet)
Flash2 Flash3	Class 2 attributes divide in proportion to flow rate of attributed component. Class 0 attributes are recalculated for each outlet stream. Polymer components are not fractionated by the phase equilibrium models used by these blocks. Equation to calculate outlet stream attributes: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed, out = outlet) A = class-2 component attribute value (in = mixed feed, out = outlet)
Mult	Class 2 attributes multiply in proportion to flow rate of attributed component. Class 0 attributes are recalculated for each outlet stream. Equation to calculate outlet stream attributes: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed, out = outlet) A = class-2 component attribute value (in = mixed feed, out = outlet)
Mixer Heater*	Class 2 attributes are summed across all feed streams. Class 0 attributes are recalculated for the outlet stream. Equation to calculate outlet stream attributes: $A_{out} = \sum_{feeds} A_{in}$ A = class-2 component attribute value (in = mixed feed, out = outlet)

**** This also applies to any block which allows multiple feed streams and uses an "implied" mixer to calculate the net feed stream.***

continued

Table 5.11 Attribute Handling (cont.)

Distillation Models	
Block	**Component Attribute Handling**
RadFrac	Component attribute conservation equations are included in this model at the tray-by-tray level. The class-2 attributes are calculated at each tray by the following equation: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed to tray, out = outlet from tray) A = class-2 component attribute value (in = mixed feed to tray, out = outlet from tray) The RadFrac model does not allow polymer reaction kinetics.
MultiFrac / BatchFrac	These unit operation blocks do not consider component attributes. Polymers must be converted to oligomer components if polymer fractionation is to be considered in these models.
Reactor Models	
RStoic RYield	If user specified attributes in the COMP-ATTR form, they are used for the product stream. Otherwise, class 2 attributes divide in proportion to the flow rate of the attributed component. Class 0 attributes are recalculated for each outlet stream. Equation to calculate outlet stream attributes: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed, out = outlet) A = class-2 component attribute value (in = mixed feed, out = outlet)
RGibbs REquil	Polymer and heterogeneous catalyst components may not participate in the reactions in these blocks. The class 2 attributes divide in proportion to the flow rate of the attributed component. Class 0 attributes are recalculated for each outlet stream. Equation to calculate outlet stream attributes: $A_{out} = \frac{F_{out}}{F_{in}} A_{in}$ F = flow rate of attributed component (in = mixed feed, out = outlet) A = class-2 component attribute value (in = mixed feed, out = outlet)
RCSTR RPlug RBatch	When using Polymers Plus reaction kinetics, these models calculate the class-2 component attributes using standard conservation equations. These models can be used with a user-written Fortran subroutine through the "USER" reaction option. If the user kinetics include component attributes, then the "COMP-ATTR" field in the user kinetics form of the reactor model must be set to "yes". In RCSTR, initial guesses for the outlet attribute values can be specified in the COMP-ATTR form.

REFERENCES

Chan, W.-M., P. E. Gloor, A. E. Hamielec, "A Kinetic Model for Olefin Polymerization in High-Pressure Autoclave Reactors," *AIChE J.*, **39**, No. 1 (1993).

Chaudhari, R. V., Y. T. Shah, Recent Advances in Slurry Reactors, Concepts and Design of Chemical Reactors, S.A. Whitaker and A. Cassano eds., Gordon and Breach Science Publishers, Switzerland (1986).

Henderson, J. N., T. C. Bouton, eds., "Polymerization Reactors and Processes," *ACS Symp. Ser.*, American Chemical Society (1979).

Rodriguez, F., Principles of Polymer Systems, Taylor & Francis, New York (1996).

Trambouze, P., H. van Landeghem, J. P. Wauquier, Chemical Reactors: Design/Engineering/Operation, Editions Technips, Paris (1988).

Walas, S. M., Chemical Process Equipment Selection and Design, Butterworths, Boston (1988).

5.2 PLANT DATA FITTING

Polymers Plus simulation models can be fit to plant or laboratory data using Data-Fit. One or more sets of measured data is provided which may include model inputs or results. Data-Fit adjusts or estimates input parameters to find the best match between the model predictions and data. Data-Fit can also reconcile measured data against the model.

Data-Fit minimizes the weighted sum of square errors, where each error is the difference between a reconciled input or calculated output and the data. In statistical terms, Data-Fit performs either ordinary least squares or maximum likelihood (errors-in-variables) estimation.

Topics covered include:

- Data Fitting Applications
- Data Fitting For Polymer Models
- Steps in Using the Data Regression Tool (including troubleshooting tips)

This section emphasizes using the Data-Fit tool to fit process reaction kinetic parameters. A more general description of this tool is available in the *Aspen Plus Reference Manual*.

DATA FITTING APPLICATIONS

The data regression tool in Aspen Plus can be used to fit model parameters and reconcile process data. These applications may be carried out simultaneously.

Parameter regression usually involves adjusting model parameters to improve the agreement between model predictions and process data. For example, process rate constants may be manipulated to match the measured polymer molecular weight and monomer conversion. Manipulated parameters may include reaction rate or equilibrium constants, physical property constants, or equipment specifications. Fitted parameters may include model predictions such as reactant conversion, product yield, by-product content, polymer component attributes, stream compositions or flow rates, or equipment heat duty, temperature, pressure, or holdup.

Data reconciliation runs involve manipulating one or more sets of model inputs to match model predictions to process data. For example, the average feed rate of a makeup stream can be estimated based on the flow rate and composition of the feed and product streams. Manipulated data typically includes feed stream flow rates and compositions, equipment operating conditions, heat transfer coefficients, etc.

The Data-Fit model can be used to reconcile input data and fit model parameters simultaneously. Simultaneous regression and reconciliation is typically used to fine-tune models which already match process data and trends relatively well.

DATA FITTING FOR POLYMER MODELS

Polymer process models frequently include non-ideal phase equilibrium, reaction kinetics, and complicated unit operations. Fitting these complex models against process and laboratory data is not a trivial task. A great deal of consideration must be given to the way this problem is approached.

A detailed example describing how to fit a free-radical reaction kinetics problem is included in the *Polymers Plus Examples & Applications Case Book*.

A general procedure for fitting complex models is given below.

Step1. Process Data Review

Collect data for the process. Sources of data include process information management system (PIMS), process design documents (PDDs), process flow diagrams (PFDs). Verify reproducibility / standard deviations of data by collecting multiple data sets for each case. Verify steady state by collecting data at regular intervals over several plant residence times. Verify data feasibility against mass and energy balance calculations.

Step 2. Literature Search

Collect information about the process. Sources of data include in-house lab data, databanks, trade journals, conference notes, polymer handbooks, on-line electronic databases, experimental designs, etc.

Step 3. Preliminary Model Fitting

Carry out physical property data regression, property constant parameter estimation runs. Test the parameters against all pertinent data from steps 1 and 2. To the extent possible, verify pure component physical properties and phase equilibrium predictions using Property Analysis tools.

Step 4. Preliminary Model Development

Develop a basic model of the process, ignoring details such as non-ideal mixing, heat transfer, etc. Specify temperature instead of duty, volume instead of residence time. Use parameters from steps 1-3.

Step 5. Trend Analysis

Use the sensitivity feature to evaluate trends between model outputs (conversion, polymer attributes, etc.) and model inputs (rate constants, operating conditions, etc.) Compare the predicted trends against available process or lab data. If the trends are not well matched, adjust specific model parameters to improve the predicted trend. Model fitting may be carried out using Sensitivity, Design-Specification, Data-Fit, or by trial and error.

Step 6. Model Refinement

Use the Data-Fit tool to carry out simultaneous parameter estimation and data reconciliation. Relax model assumptions, such as perfect mixing, as needed. Bring model up to the appropriate level of detail, fitting key parameters at each development step.

Data Collection and Verification

The first step in fitting a model is to collect and review data. Sources of data may include process information management system (PIMS), process design documents (PDDs), and process flow diagrams (PFDs), shift log sheets, and laboratory analysis reports. It is important to verify the reproducibility of the data by collecting several duplicate sets of each datum. Duplicate data are especially important for analytical measurements such as melt flow index and intrinsic viscosity.

For continuous processes, it is a good idea to verify that the process operates under steady-state conditions by collecting data at regular intervals. The data should be collected at regular intervals over a period which exceeds the cumulative residence time of the key unit operations in the process.

Verify data feasibility against mass and energy balance calculations. It is impossible to force a rigorous model to match data which violates the fundamental conservation equations.

When possible, obtain calibration data for unit operating conditions, especially level calibration data for reactors and flow rate calibration data for flow meters. The method and assumptions used to calibrate these instruments must be taken into consideration for data reconciliation runs.

Literature Review

Before you regress process data, it is a good idea to collect information about the process. Sources of data include in-house lab data, databanks, trade journals, conference notes, polymer handbooks, on-line electronic databases, experimental designs, and so on.

The open and in-house process literature may contain a wealth of information about key model parameters. Further, these sources may provide additional sources of fundamental data which can be used to independently evaluate model parameters.

Simulation studies described in trade journals are an excellent source of insight and know-how related to model development. These studies frequently point out which assumptions are valid and which parameters are important. In addition, these papers may elucidate reaction mechanisms or physical phenomena which should be considered in a rigorous process model.

The physical property and rate constant data reported in the open literature are never perfect, but they do serve as a good starting point for fitting the model.

Preliminary Parameter Fitting

It is important to determine as many of the model parameters as possible early in the model development process. Try to decouple the parameters from each other whenever possible. For example, find ways to establish phase equilibrium parameters independently of reaction equilibrium constants. Make simplifying assumptions to reduce the number of unknown parameters.

Physical property parameters should be firmly established before fitting rate constants. When data are available, use the physical property data regression system (DRS) to fit the density, enthalpy, heat capacity, and vapor pressure of pure components. If phase equilibrium data are available, use DRS to regress phase equilibrium parameters.

When property data are unavailable for a component, the property constant estimation system (PCES) can be used to estimate property parameters from molecular structure. These estimations, however, should be checked against process data. If data are available for components with similar structures, they can be used to estimate the properties of components which are not found in the databank.

Table 5.12 lists some of the key physical property parameters in Polymers Plus and describes how they influence polymerization kinetics.

Table 5.12 Influence of Physical Property Parameters on Kinetics

Property	Parameters	Influence on Polymerization Reaction Kinetics
Density	DNLRKT, DNLVK	Concentration is proportional to density. Reaction kinetics depend on component concentrations.
Vapor pressure	PLXANT, HENRY	The vapor pressure controls phase equilibrium of volatile components in vapor-liquid systems. The phase equilibrium strongly influences concentrations, which controls kinetics.
Enthalpy	DHFORM, DHFVK, DHFVKM, DHSUB, DHCON, DHFMDP	The component enthalpies influence the predicted heat duties and temperatures in the model.
Heat capacity	CPIG, CPL, CPLVK, CPCVK	The heat capacity controls the influence of temperature on enthalpy.
Transition temperatures	TMVK, TGVK	Phase transitions occur at the melting point and glass point. Predicted enthalpy, density, and heat capacity for polymer and oligomer components depend on the phase regime.
Phase equilibrium		In multiphase reactors the phase equilibrium determines the component concentrations in each phase, which influences the reaction rates.
Solubility (of a solid)	K-SALT	The solubility parameter influences the concentration of partially soluble solids in the liquid phase. When catalysts, inhibitors, or monomers are fed as solids, this parameter controls their concentration, which in turn controls their reaction rate.

If reaction kinetic parameters are unavailable from in-house or open literature, it may be necessary to carry out experiments to determine the magnitude of the rate constants. Carry out the reactions under controlled conditions to isolate the influence of reaction kinetics from phase equilibrium, mass transfer, heat transfer, etc. For example, carry out the experiments in sealed tubes so the liquid phase concentrations are unaffected by phase equilibrium.

Reaction experiments should be performed over a range of temperatures to allow determination of the activation energies.

Preliminary Model Development

Once the preliminary parameter fitting is complete, these parameters can be used to develop a preliminary model. At this stage of the model development process, it may be best to use simplified models for some of the ancillary operations which are not directly involved in the polymerization reactors. For example, it may be more convenient to represent distillation columns using the non-predictive Sep or Sep2 models instead of the RadFrac or MultiFrac rigorous distillation models.

The most important rule for model development is to “keep it simple”. Model development must be carried out in several stages. Add detail to the model one step at a time. Each generation of the model can yield valuable insights into the process and can provide substantial benefit to the model developer. At each stage in the process, fit the appropriate model parameters and validate the model against all sources of available data. Verify the predicted trends against process data, operator experience, and engineering know-how. Over time, the level of detail and power of the model can be increased.

During the preliminary development, use the most basic specifications possible. For example, in the RCSTR model specify temperature and reacting phase volume instead of duty and residence time. This approach will make the model run faster and will help to isolate the influence of property parameters from reaction kinetic parameters.

Once the preliminary model is complete, it can be tested against process data. Major discrepancies between the data and the model predictions should be addressed during this step.

Trend Analysis

Use the preliminary model to carry out trend evaluation studies. The sensitivity feature can be used to examine the influence of process variables on the model predictions. Compare these trends against process data. If the predicted trends are not consistent, adjust the appropriate model parameters to improve the match. For example, if the predicted slope of the monomer conversion versus temperature curve is less than the measured slope, the activation energy of the polymerization reaction may be too low.

Use the sensitivity tool to examine the influence of the model parameters on the model predictions and to determine which parameters are important in the model. Parametric studies can be carried out by manipulating two or more variables in a sensitivity study.

It is good practice to include as many model predictions as possible in each sensitivity study. The simulation runs take the same amount of time regardless of the number of defined variables. It is much easier to understand the predicted trends when the sensitivity results are detailed.

Once you know which parameters are critical to the model predictions, the data regression tool can be used to adjust these parameters to match specific trends. Keep the number of manipulated parameters to a minimum until all of the key parameters are established independently.

Model Refinement

The Data-Fit tool is the best choice for refining the fit between the model predictions and the process data, especially when several sets of data are available. Data-Fit can adjust several model parameters simultaneously, capturing subtle interactions among the parameters to get the best overall match between the process data and model predictions.

When the model predictions cannot match the process data, the assumptions in the model may be too broad. Perhaps the process is limited by heat- or mass-transfer, or a reactor is not ideally mixed. Maybe there are additional side reactions which should be considered in the model, or the rate expression needs to be modified to account for some unusual aspect of reaction kinetics. These issues can be addressed during the model refinement process by adding new layers of detail to the model. Avoid adding more detail than necessary, however, because model fitting is a process of diminishing returns.

Model refinement is an open-ended process. The model parameters can be tuned more accurately as more data become available from the process. Bad data points are easier to spot when there are more sets of data to compare.

It is impossible for a simulation model to match process data perfectly. There are several sources of error that lead to differences between the model results and process data, including:

- Variations in process operating conditions due to disturbances, excursions from steady state, control system actions, etc.
- Imperfect calibration of flow meters, level controllers, etc.
- Analytical error in lab measurements
- Simplifications and assumptions in the model, such as ideal mixing, isothermal and isobaric vessels, phase equilibrium, etc.
- Errors in the model parameters.

STEPS IN USING THE DATA REGRESSION TOOL

There are three steps involved in using the data regression tool:

- Creating a base-case model
- Entering lab or process data and operating conditions into data sets
- Defining regression cases

Step 1. Creating a base-case model

If the regression tool is being used to fit reaction kinetic parameters from lab batch reactor data, use the RBatch model with an appropriate reaction kinetic model.

If the model parameters are being regressed from process data, develop a model of the process. Before setting up the data fit run, make sure the model predictions are reasonable and that the model is robust (converges without errors) over the ranges of each manipulated parameter. You can use sensitivity blocks to screen the model for accuracy and to test how robust the model is.

The rate constants and property parameters entered into the base case model become the initial estimates for the regression.

Step 2. Entering lab or process data and operating conditions into data sets

There are two types of data sets used with the regression tool, “Point-Data” and “Profile-Data”:

Use	To specify
Point-Data	• Operating conditions for steady-state unit operation models. • Feed streams for continuous processes or batch charge streams. • Analytical data, measured flow rates, or composition data for product streams. • Polymer or catalyst component attribute data for product streams.
Profile-Data	• Operating profiles for batch reactors or plug-flow reactors, including temperature, pressure, and duty profiles, continuous feed stream profiles, etc. • Time-series measured data for a batch reactor or data along the axial profile of a plug-flow reactor. • Note: Component attribute profiles and user variable profiles are not available as profile data in this release of Polymers Plus. To fit profile data for these types of variables, treat each data point in the profile as a point datum, and specify the coinciding stop-time (RBatch) or length (RPlug) of the reactor as another point datum in the same data set.

Step 3. Defining regression cases

For each case, specify the parameters to be adjusted and the data sets to be fitted. Several regression cases can be included in the same simulation run. The cases are run sequentially; a prompt will appear on the screen which allows you to specify which cases to include in the run, and the sequence order of the cases. Each successive case uses the fitted parameters and reconciled data from the previous case. If the data regression is run again, the previously fit values are used as initial estimates unless the simulation is reinitialized.

Identifying Flowsheet Variables

You must identify each measured and manipulated variable considered in the regression. Most types of variables, such as stream flow rates, equipment operating conditions, and component attribute values can be accessed directly using the variable accessing system..

In the data regression and data set forms, you cannot access vector data, such as the stream vector and component attribute vector. You must access each stream variable or component attribute element as a separate scalar variable.

When specifying feed stream data, avoid using mole, mass, or volume fractions as variables in the data set. If the composition of the feed stream changes from one validation case to another, specify the flow rates of the components in the stream. If the composition is constant but the flow rate changes, specify the composition and base-case flow rate in the model, and specify the total stream flow rate as a point-data variable. This avoids problems with normalizing fractions and reduces the number of variables handled by the data-fit algorithm.

Some unit operation models have both input and results variables for the same operating condition. For example, in the RCSTR model you can access the specified heat duty (DUTY), or the calculated reactor duty (QCALC). If a variable is an INPUT variable in the regression it must be specified in the unit operation model.

For example, if the reactor duty is a manipulated INPUT variable in the regression, it must be specified as an input variable (DUTY), and the reactor duty must be specified in the reactor model. If the reactor duty is a measured RESULTS variable, it must be specified as a results variable (QCALC), and is usually not specified in the model (the temperature is specified instead). Table 5.13 provides a cross-reference of commonly-used INPUT and RESULTS variables for key specifications related to several unit operation models.

Some measured data, such as polymer melt index and intrinsic viscosity, are not predicted by the standard property sets in Polymers Plus. The best way to include these properties in a data regression is to write a user Prop-Set property subroutine. Each user property can be linked to a property set. Property sets can be accessed as stream-property variables.

Table 5.13 Cross Reference of Selected Input and Results Variables

Model	Operating Condition	Input Variable	Results Variable
Rbatch	Cumulative reactor duty	DUTY	QCALC
RCSTR with one phase	Duty	DUTY	QCALC
	Pressure	PRES*	use outlet stream pressure
	Temperature	TEMP	TCALC
	Reactor volume	VOL	VOL-CALC
	Reactor residence time	RES-TIME	RT-CALC
RCSTR with multiple phases	Reacting phase volume	REACT-VOL	VOLL-CALC for liquid volume
			VOLV-CALC for vapor volume
			VOLLS-CACL for total liquid+solid volume
	Reacting phase residence time	PH-RES-TIME	VOLL-CALC for liquid residence time
			RTV-CALC for vapor residence time
			RTLS-CALC for liquid or solid residence time
RPlug	Duty	DUTY	QCALC
	Pressure (process fluid)	PRES* (feed)	REAC-PRES**
	Temperature (process fluid)	SPEC-TEMP**	REAC-TEMP**
	Residence time (process fluid)	RES-TIME	RT-CALC (entire reactor)
			REAC-RESTIM** (residence time at a profile point)
Flash2 and Flash3	Duty	DUTY	QCALC
	Pressure	PRES*	use outlet stream pressure
	Temperature	TEMP	use outlet stream temperature
RadFrac and MultiFrac	Condenser duty	Q1	COND-DUTY
	Reboiler duty	QN	REB-DUTY
	Reflux ratio	basis-RR***	RR
	Boilup ratio	basis-BR***	BR
	Stage temperature	STAGE-TEMP	TEMP
	Stage pressure	STAGE-PRES	PRES
	Design specification setpoint	VALUE	various - it depends on the specification

**** The pressure variable is treated as a pressure drop if the specified value is non-positive.***

***** Specify location (RPlug) or stage number (RadFrac / MulitFrac)***

****** Basis can be MOLE, MASS, or STDVOL - the variable specified in the data set must match the variable specified in the column***

Manipulating Variables Indirectly

In-line Fortran blocks can be used to enforce assumptions in the model or to manipulate variables indirectly. Using these techniques to reduce the number of manipulated variables can greatly enhance the speed and reliability of the regression.

Example 1: Using Fortran Blocks to Enforce Modeling Assumptions

Suppose:

- Your process involves a catalyst and an initiator.
- The key variables involved in the regression cases are the process operating conditions and the monomer feed rate. The catalyst and initiator flow rates are always proportional to the monomer feed rate.

Create a Fortran block and define the monomer, catalyst, and initiator flow rates as flowsheet variables. Specify the monomer flow rate as a “read variables” and the catalyst and initiator flow rates as “write variables” as shown below:

```
FORTRAN SETCAT
  DEFINE FLOMON MASS-FLOW STREAM=FEED COMPONENT=MONOMER
  DEFINE FLOINI MASS-FLOW STREAM=ADDITIVE COMPONENT=PEROXIDE
  DEFINE FLOCAT MASS-FLOW STREAM=CATALYST COMPONENT=METAL
  READ-VARS FLOMON
  WRITE-VARS FLOINI FLOCAT
C       Specify the base-case flow rates in kg/hr below
F       BCMON = 1200.0
F       BCCAT = 20.0
F       BCINI = 5.0
C       Calculate the flow rates of initiator and catalyst
F       FLOINI = FLOMON * BCINI / BCMON
F       FLOCAT = FLOMON * BCCAT / BCMON
```

Define the monomer flow rate as a variable in a point-data set. During the data regression run, the regression model will write the monomer flow rate for each case. The Fortran block will be executed each time the regression block manipulates the monomer flow rate. The Fortran block will read the new monomer flow rate, calculate the initiator and catalyst flow rates, and write their values.

Using this technique to indirectly manipulate the additive flow rates reduces the number of variables in the regression, making the regression faster and more reliable. The cost of this approach is that the indirectly manipulated variables (catalyst and initiator flow rates) cannot be reconciled (the model has no information regarding the standard deviations of these variables).

Example 2: Using Parameters and Fortran Blocks to Indirectly Manipulate Process Variables

Suppose:

- Your polymerization process uses two monomers.
- The key variables involved in the regression cases are the monomer ratio and the polymer production rate. You want to vary these parameters in the data regression.

In the base-case model, define the monomer ratio and production rate as “parameter” variables in a Fortran block. Specify the base-case monomer ratio and production rate in the same Fortran block. Specify this block to sequence “first”, as shown below:

```
FORTRAN INITIAL
  DEFINE RATIO  PARAMETER 1
  DEFINE PRODRT PARAMETER 2
  SEQUENCE FIRST
C       specify monomer mole ratio
F       RATIO = 1.05
C       specify polymer production rate, kg/hr
F       PRODRT = 2000.0
```

Create a second Fortran block. Define the monomer flow rates as flowsheet variables. Access the monomer mole ratio and production rate parameters. Specify the parameter variables as “read variables” and the monomer flow rate variables as “write variables”. After solving the algebra, the Fortran block can be defined as shown below:

```
FORTRAN ADJUST
  DEFINE RATIO  PARAMETER 1
  DEFINE PRODRT PARAMETER 2
  DEFINE FLOM1 MOLE-FLOW STREAM=FEED COMPONENT=MONO-1
  DEFINE FLOM2 MOLE-FLOW STREAM=FEED COMPONENT=MONO-2
  READ-VARS RATIO PRODRT
  WRITE-VARS RATEM1 RATEM2
C       w = mole weight of each monomer
F       WM1 = 150.23
F       WM2 = 230.30
C       calculate average molecular weight of monomers
F       RATINV = 1.0 / RATIO
F       WMAVG = ( 1.0 + RATINV ) * ( WM1 + WM2*RATINV )
C       calculate monomer flow rates in kmol/hr
F       FLONET = PRODRT / WMAVG
F       FLOM1 = FLONET / ( 1.0 + RATINV )
F       FLOM2 = FLONET - RATEM1
```

The production rate and mole ratio parameters can be accessed as parameter variables in the data-set. The standard deviation for the production rate and mole ratio variables may be specified; the units of the standard deviations are the same as the units of the parameters.

Entering Point Data

There are two types of point data: input variables and result variables. Input variables include feed stream flow rates, equipment operating conditions, and other parameters which are inputs to the simulation model. Result variables include product stream flow rates or composition, polymer or catalyst component attributes, stream properties, or any other simulation calculation which can be compared to measured process data.

If some results data are missing from one or more sets of data, they can be left blank on the input forms. The model will estimate the values of these results and tabulate them after the regression run.

Unknown input data may also be estimated. Leave the input field blank and specify large standard deviations (for example, 50%) for each missing datum. Supply a realistic initial guess and make sure the standard deviation results in reasonable bounds for each missing variable.

The upper and lower bounds for reconciled unknown input variables are determined from the specified standard deviation and the “bound factor”, which defaults to ten:

Lower bound = Measured value - (Bound Factor)*(Standard Deviation)

Upper bound = Measured value + (Bound Factor)*(Standard Deviation)

Make sure these limits are reasonable. In particular, the limits for a stream flow rate must not allow the stream flow rate to become zero or negative.

Entering Profile Data

The plug-flow reactor model (RPlug) predicts results at various points along its length axis. The batch reactor model (RBatch) predicts results at various points in time during the batch cycle. You can define profile data sets to specify the operating profiles as input data, or to fit the model to measured results data.

To do this, specify the time and value for each datum in the profile. You can specify standard deviations for results variables. Data reconciliation is not allowed for input profile data.

Table 5.14 lists the profile data sets which are currently available for these reactor models. Other types of profiles, including component attribute, user variable, and user Prop-Set property profiles are planned for a future releases of Polymers Plus.

If you are fitting component attribute or user Prop-Set property profiles, you must treat the measured variables as point data for the reactor outlet stream. Use the reactor length or stop-time as an additional point data. Each profile point must be treated as a separate data case in the data set.

If some results data are missing from one or more sets of profile data, they can be left blank on the input forms. The model will estimate the values of these results and tabulate them after the regression run.

Table 5.14 Summary of Available Types of Profile Data

Model	Variable Type	Description	Profile Name
RBatch, RPlug	Input	Temperature of process fluid	TEMPERATURE
		Pressure of process fluid	PRESSURE
		Instantaneous reactor duty	DUTY
	Results	Partial pressure of a component	PARTIAL-PRES
		Molar concentration of a component in the liquid phase	MOLECONC-L
		Molar concentration of a component in the vapor phase	MOLECONC-V
		Mole fraction of a component in the liquid phase	MOLEFRAC-L
		Molar fraction of a component in the vapor phase	MOLEFRAC-V
		Mass concentration of a component in the liquid phase	MASSCONC-L
		Mass concentration of a component in a slurry phase	MASSCONC-LS
		Mass fraction of a component in the liquid phase	MASSFRAC-L
		Cumulative reactor heat duty	CUM-DUTY
		Instantaneous vent mole flow rate	VENT-MOLFLOW
		Instantaneous vent volume flow rate	VENT-VOLFLOW
RBatch	Input	Feed stream component flow rates	not applicable

Entering Standard Deviations

Standard deviations may be specified for input and result variables. The standard deviation is the level of uncertainty in the measurement. You can enter the value as an absolute or percent error (append a percentage sign, %, to the value). Statistically determined standard deviations may be available from an on-line process information management system (PIMS), from lab databases, or from other information resources. When the standard deviations are not available, you can enter your best estimate of the expected error based on your experience or the specifications of the instrument.

The objective function of the data regression is to minimize the sum of weighted square errors. For results variables, each error is defined as the difference between the reconciled or specified datum and the value calculated by the model. Each error is scaled against the square of the standard deviation:

$$\text{Objective function} = \sum_i \frac{\text{Measurement}_i - \text{Prediction}_i}{(\text{Standard deviation})_i^2}$$

If the specified standard deviation of a variable is too small, the model over-emphasizes the importance of the variable during the fitting process. This may cause the model to make unreasonable adjustments in some parameters to force good fits to variables with small standard deviations.

You must be careful to consider both the precision and accuracy of each variable. For example, a variable may have a low standard deviation because it is very precise (it reproduces well in successive trials), but the measurement may be inaccurate (it may not reflect the true value of the measured parameter). Consider the case where a level controller may show little deviation in the liquid volume in a reactor, but the calibration of the level transducer may not be accurate to within ten percent of the real liquid volume. In this case, the standard deviation of the specified liquid volume should be large enough to reflect the accuracy of the volume, not the deviation of the liquid level.

If standard deviations are specified for input variables, the model reconciles these variables. If you do not specify the standard deviation of an input variable, the model assumes the specified values are exact. Reconciling input variables accounts for measurement errors in the operating conditions and can lead to better models, but it can substantially increase how long the run takes to complete.

Standard deviations must be specified for each of the result variables. Specify reasonable standard deviations to keep the model from forcing a match by making wild adjustments to the parameters. The specified standard deviations are probably too small (or the data quality is poor) if several of the parameters reach their upper or lower bounds.

Defining Data Regression Cases

You can fit any number of data sets in the same regression case. Point-Data and Profile-Data may both be included. Each regression case must involve at least one estimated parameter and at least one reconciled input variable. There are no upper limits to the number of estimated parameters and reconciled inputs, however the required simulation time is very sensitive to the number of variables included in each regression case.

Each input variable with a non-zero standard deviation is reconciled (adjusted). The reconciled inputs are tabulated in the regression results.

Each estimated parameter must be defined in the base case, or have a default value (such as a physical property parameter). The specified values for the base case run are used as the initial guesses for the regression. If the base-case value lies outside the specified bounds, the boundary condition closest to the base case value is used.

Sequencing Data Regression Cases

For data fit problems, Aspen Plus will:

- Run the base-case simulation
- Execute the data regression
- Replace the base-case parameter values with the estimated parameter values and rerun the base-case simulation

If Case-Study or Sensitivity blocks are present, Aspen Plus runs them after the regression is complete. The estimated parameter values are used to calculate the results for these blocks.

Flowsheet convergence loops and Design-Specification loops are used in the preliminary and final base-case simulations and they are sequenced inside the data regression loop. The sequencing of Fortran blocks and Transfer blocks depends on which variables are accessed.

If more than one regression is included in a simulation, the regressions can be affected sequentially. Each successive regression uses the estimated parameters from the previous regression.

Regression blocks can be manually sequenced if the automatic sequence does not meet the needs of a particular run, however automatic sequencing is usually the best choice.

Interpreting Data Regression Results

The key results of the data regression tool are:

- The Chi-square statistic and critical Chi-square value for the fit.
- Estimates and standard deviations for each estimated parameter.
- A table of the measured values, estimated values, and normalized residuals for each data set.

The Chi-square value is an indicator of the quality of the fit. A model is considered well fit if the Chi-square value falls below the critical Chi-square value. The reliability of different fits or different modeling approaches can be tested by comparing the Chi-square values of the fits. For example, suppose a reactor is thought to have non-ideal mixing. This assumption can be evaluated by developing two models, one which assumes ideal mixing (one CSTR stage) and one which assumes non-ideal mixing (a series of CSTR stages). The two models can be fit against the same data using the same parameters. The model with the lower Chi-square statistic represents the data more accurately, and can be considered the most realistic

Ideally, the standard deviations of the estimated parameters are small, and the confidence interval of each parameter is narrow. In practice, however, the standard deviation of the parameters may be relatively large. This does not necessarily indicate a poor fit. For example, if the activation energy and pre-exponential factor for a reaction are both included as estimated parameters in the data regression, then the standard deviation of the estimated pre-exponential factor will be large. In this example, small differences in one parameter (the activation energy) requires large differences in another parameter (the pre-exponential factor) to keep the model predictions relatively constant.

The residual values are indicative of the difference between the measured data and model predictions. For fitted data, the residuals are defined as:

$$\text{Residual}_i = (\text{Measured value}_i - \text{Predicted value}_i)^2 / (\text{Standard deviation}_i)$$

For reconciled data, the residuals are defined as:

$$\text{Residual}_i = (\text{Measured value}_i - \text{Estimated value}_i)^2 / (\text{Standard deviation}_i)$$

Review the residual values to verify they are sensible. Large residual values may indicate a major problem with the model or data, or may reflect an unreasonably tight standard deviation. Never specify extremely tight standard deviations. This causes the data regression algorithm to waste time attempting to obtain tight fits on some variables. If some data are considered extremely accurate, they should be assigned standard deviations of zero.

The regression results may be plotted against the initial estimates and measured data. Plots of this type include a 45° dotted line which indicates a “perfect fit”, e.g., each prediction is exactly equal to the measured data. Points which fall far from this line are the least well fit. Verify these outliers to make sure the data is correctly entered into the model and that the units of measurement are consistent.

Troubleshooting Convergence Problems

If the data regression tool fails to converge, check the objective function. A large objective function value indicates a poor fit between the model predictions and measured data. If the objective function is large, review the residual values for each type of measured data. Large residual values may indicate a very basic error in the data entry. For example, the data may be entered in the wrong units or there may be typing errors in the specified values. Always review the model thoroughly to eliminate these types of problems before adjusting convergence parameters or making other major changes to the regression.

Convergence errors can occur for a number of reasons. When a problem occurs, ask:

- Does the base case model converge well and give reasonable results?
- Is the base case model formulated to handle data which may be out of mass or energy balance?
- Are the initial estimates of the parameters good enough?
- Are the specified standard deviations reasonable?
- Do the model inputs completely determine the measured results?
- Do the specified bounds allow the regression to take the model into infeasible regions, causing the unit operation blocks or flowsheet convergence to fail?
- Are the assumptions and simplifications in the model reasonable?

Regression runs with many variables and runs for highly non-linear models may still be difficult to converge. In some cases, the convergence criteria may be unnecessarily tight. Table 5.15 summarizes several convergence parameters which can be used to tune a regression run. It is not necessary to adjust the convergence parameters for most regressions.

Table 5.15 Partial Summary of Data Regression Convergence Parameters

Parameter	Description
ALG-ITERATION	Maximum number of algorithm iterations. The default value is sufficient for nearly all problems
MAX-PASSES	Maximum number of flowsheet passes. This parameter may need to be increased for regressions involving a large number of variables.
SSQTOL	Convergence tolerance for sum of weighted square errors (Absolute objective function tolerance) This is the absolute tolerance for the objective function. The default tolerance is very tight, so regressions which converge to this tolerance should be reviewed thoroughly. Verify that the specified standard deviations are sensible. Change the default value of this parameter if you which to fit the model to achieve a particular objective function value.
RFCTOL	Relative objective function tolerance. The problem is considered converged if the model predicts that the maximum possible objective function is less than the product of the relative function tolerance and the current value of the objective function. For example, if RFCTOL is 0.1, then the model is converged when the predicted change in the objective function is less than ten percent of the objective function value for the current iteration.
XCTOL	Minimum variable step-size tolerance. The problem is converged if the relative step size in the variables falls below XCTOL and the objective function is decreasing slowly (less than 50% per iteration).
XFTOL	Minimum objective step-size tolerance
INIT-STEP	Factor used to determine initial step sizes. This factor can profoundly affect the performance of the algorithm. If the initial steps are too large or too small, the model must adjust the step size until appropriate step sizes are determined.
PERT-FACTOR	During the regression, the model determines the response of each variable to each other variable by making small adjustments, or pertubations, to the variables. The size of these adjustments is determined by the algorithm, this parameter is used to determine the maximum pertubation step sizes for each variable. You may need to increase this value when the fitted data are not very sensitive to the manipulated parameters, or decrease this value when the sensitivity is very strong.
BOUND-FACTOR	Factor used to determine lower and upper bounds for reconciled inputs. If the value is too large, the model may enter an infeasible region, for example a stream flow rate may go to zero. If the value is too small, the parameter ranges may be too narrow to allow fit the data.
INIT-METHOD	Method used to initialize the regression. Specify BASE-CASE to use the base case values to initialize the reconciled input parameters. Specify MEASUREMENTS to use the measured data to initialize the reconciled inputs.

Ensuring Well-Formulated Regressions

Poorly formulated regressions may result in large residual values and a large objective function. Before starting a regression run, use sensitivity studies to test the model. Verify that the manipulated parameters have a strong influence on the measured data. Don’t try to fit parameters which have only a weak impact on the model predictions.

Make sure the parameter ranges are sensible. It is a waste of time to fit a parameter within a narrow range (less than 5%). On the other hand, if the range is too large, the regression algorithm may push the model into an infeasible region. For example, if the distillate to feed ratio in a column is allowed to decrease to zero, the column model will fail.

The way the data regression is formulated has a major influence on how quickly and easily the problem converges. De-couple the manipulated variables as much as possible. For example, don’t fit the rate constants and phase equilibrium parameters at the same time if the two sets of parameters can be fit independently in two smaller data regression runs.

Use the weighing factors if some sets of data are more reliable than others. A larger weight may be assigned to a set of data that are based on long-term averages from the process information management system, lower weights might be assigned to data based on poorly kept records from the distant past.

Make sure the manipulated parameters can be determined from the available data. For example, the activation energy of a reaction cannot be determined from isothermal data.

The base-case file needs to be formulated in a robust manner. If the base case model does not converge reliably away from the base case condition, then it is likely that the regression run will fail. Use the sensitivity tool to verify that the model is stable over the entire range of each manipulated parameter and to verify that the model is sensitive to each parameter.

Where possible, use relative or normalized inputs instead of absolute inputs. For example, in column models use the distillate to feed ratio (D:F) instead of distillate flow rate. Use pressure drop specifications instead of pressure. These specifications make the model more reliable and help to avoid problems which occur if the measured data is out of mass balance with each other.

Fitting Activation Energy

It is tempting to try to fit activation energies and pre-exponential factors in the same regression run. This can lead to significant headaches if the problem is not approached right. Consider, for example, the standard Arrehnius rate expression:

$$k_{net} = k_o \exp^{\frac{-E_{act}}{RT}}$$

Using this expression, the net rate constant, k_{net}, is sensitive to the activation energy, E_{act}. If the activation energy is adjusted a little bit, a large adjustment must be made to the pre-exponential factor to offset this difference. In other words, the activation energy controls the magnitude of the reaction rate as well as the temperature sensitivity of the reaction rate.

A better approach is to use the modified Arrehnius expression:

$$k_{net} = k_o \exp^{\frac{-E_{act}}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)}$$

The parameter T_{ref} is a reference temperature which typically represents the middle of the temperature range used to estimate the activation energy. Using this formula, the net rate constant, k_{net}, remains constant at the reference temperature regardless of the value of the activation energy. With this approach, the pre-exponential factor, k_o, controls the magnitude of the reaction rate at the reference temperature. The activation energy, E_{act}, controls the temperature sensitivity of the rate constant. This makes it much easier to fit the model.

Scaling the Fitted Parameters

When several types of parameters are adjusted in the same run, the magnitude of the manipulated parameters may influence how well the data regression converges. Ideally, the manipulated parameters should be within several orders of magnitude of each other.

Suppose, for example, the manipulated parameters include rate constants for several different types of reactions. These expected values of the rate constants may differ by several orders of magnitude. In this situation, the regression procedure may over-emphasize the manipulated variables with the smallest magnitude.

You can get around this problem using two Fortran blocks as shown in Example 3. Use one Fortran block to define a PARAMETER variable for each manipulated variable in the regression. Initialize each parameter to one. Use a second Fortran block to READ these parameter values, to multiply them by base case values, and then WRITE the results to the manipulated variables. In the data regression block, manipulate the PARAMETER variables.

This technique allows the data regression to operate on normalized variables instead of absolute variables which makes it much easier for the regression algorithm to choose appropriate step sizes and ensures that the variables are given equal weighting by the algorithm.

Example 3: Using Fortran Blocks to Scale Manipulated Parameters

Problem Description: Suppose two pre-exponential factors are adjusted to match conversion and intrinsic viscosity, which are defined as user Prop-Set properties. The pre-exponential factors have very different magnitudes, so scaling is required to get a good fit.

Instead of manipulating the rate constants directly, use PARAMETER variables to define and manipulate correction factors for the rate constants. Use a Fortran block to initialize these correction factors to unity. Manipulate these PARAMETER variables in the regression. Use a Fortran block to adjust the pre-exponential factors using the correction factors manipulated by the data regression model.

```
USER-PROPERTY INT-VISC SUBROUTINE=USRPSP FLASH=YES
USER-PROPERTY CONVERSN SUBROUTINE=USRPSP FLASH=YES
PROP-SET INT-VISC INT-VISC
PROP-SET CONVERSN CONVERSN

DATA-SET DS-1
  DEFINE CAT MASS-FLOW STREAM=CATALYST SUBSTREAM=MIXED COMPONENT=CAT
  DEFINE TEMP BLOCK-VAR BLOCK=CSTR1 SENTENCE=PARAM VARIABLE=TEMP
  DEFINE VISC STREAM-PROP STREAM=PRODUCT PROPERTY=INT-VISC
  DEFINE CONV STREAM-PROP STREAM=PRODUCT PROPERTY=CONVERSN
  USE STD-DEV 0.001    0.1 0.002 0.0050 /
      DATA    0.025  290.0 0.844 0.8550 /
      DATA    0.023  295.0 0.842 0.8700 /
      DATA    0.055  280.0 0.850 0.9050 /
      DATA    0.033  292.0 0.835 0.9000

STEP-GROWTH MYMODEL
  RATE-CON 1 PRE-EXP=9.67D14 ACT-ENERGY=41.0
  RATE-CON 2 PRE-EXP=3.25D0  ACT-ENERGY=0.0
etc…

FORTRAN INITIAL
  DEFINE P1 PARAMETER 1
  DEFINE P2 PARAMETER 2
  P1 = 1.0D0
  P2 = 1.0D0
  EXECUTE FIRST

FORTRAN ADJUST
  DEFINE P1   PARAMETER 1
  DEFINE P2   PARAMETER 2
  DEFINE EXP1 REACT-VAR REACTION=MYMODEL VAR=PRE-EXP SENT=RATE-CON ID1=1
  DEFINE EXP2 REACT-VAR REACTION=MYMODEL VAR=PRE-EXP SENT=RATE-CON ID2=2
C     specify base case pre-exponential factors for side rxn 1 and 2
F     BASE1 = 9.67D14
F     BASE2 = 3.25D0
C     calculate pre-exponential factors using correction factors
C     manipulated by the data regression block
F     EXP1 = BASE1 * P1
F     EXP2 = BASE2 * P2
  READ-VARS P1 P2
  WRITE-VARS EXP1 EXP2

REGRESSION FIT-1
  DATA DS-1
  VARY PARAMETER 1 LABEL=”CORRECT” “FACTOR” “RXN #1”
  LIMITS 0.1 10.0
  VARY PARAMETER 2 LABEL=”CORRECT” “FACTOR” “RXN #2”
  LIMITS 0.1 10.0
```

5.3 USER MODELS

This section discusses the features available in Polymers Plus for incorporating user modules into a simulation model.

Topics covered include:

- User Unit Operation Models
- User Kinetic Models
- User Physical Property Models

More details concerning User Models are available in *Aspen Plus User Models*.

USER UNIT OPERATION MODELS

There are cases where users may need to create special models to represent a process. Usually these models can be configured by combining several of the standard unit operation building blocks. For more complex reactor geometries or in order to represent highly non-ideal systems users may need to provide their own model as a Fortran subroutine.

There are two user unit operation blocks available: USER and USER2. The first allows a limited number of inlet and outlet streams. The second allows multiple inlet and outlet streams. Both unit operations take full advantage of the Aspen Plus flowsheeting capabilities. The required Fortran subroutine must process the feed streams and return the condition and composition of the outlet streams.

User Unit Operation Models Structure

There are three stages to the execution of Aspen Plus unit operation models: input processing, simulation calculations, and report writing. Normally, the implementation of a new model requires that all three stages be accounted for. However, in the case of USER2 models, a generic framework handles the input setup and processing stage. A Fortran subroutine must be written to perform the simulation calculations and for writing the report. If no report writer is provided Aspen Plus automatically echoes the input data in the report.

Figure 5.19 summarizes the simulation sequence of a unit operation model.

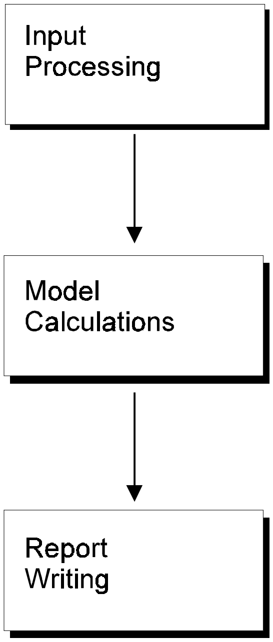


Figure 5.19 *Stages in Unit Operation Model Execution*

User Unit Operation Model Calculations

A user unit operation model can be programmed to represent any unit operation. Most applications would include combinations of the following: separations, reactions, heat transfer, mass transfer, mixing and splitting. There are some common steps that are found in the simulation calculations within unit operation models, including user models. These steps include:

- Feed processing
- Physical properties and phase equilibrium calculations
- Unit operation calculations (kinetics, heat transfer, mass transfer, etc…)
- Results storage and outlet stream initialization

Utilities are available to facilitate each of these steps. Table 5.16 summarizes the utilities available.

Table 5.16 Fortran Utilities and Monitors

Stream Handling *	
NPHASE	Determines number of substreams
LPHASE	Finds the location of a substream within a stream
SSCOPY	Copies a substream from one stream to another
NSVAR	Determines the size of the stream vector
Component Attribute Handling *	
GETDPN	Find the number average degree of polymerization
GETMWN	Find the number average molecular weight
LCATT	Finds the location of a component attribute in the stream vector
Component Handling *	
CPACK	Packs out trace components
ISPOLY	Determines if a component is a polymer
ISSEG	Determines if a component is a segment
ISOLIG	Determines if a component is an oligomer
KCCID	Finds the component index (position in stream vector)
Property Monitors **	
KVL	Calculates vapor-liquid equilibrium ratio (K-value)
KLL	Calculates liquid-liquid equilibrium ratio
ENTHL	Calculates liquid mixture enthalpy
VOLV	Calculates liquid mixture molar volume
FUGLY	Calculates liquid mixture fugacity coefficient
IDLGAS	Performs ideal gas calculations
VISCL	Calculates liquid mixture viscosity
Flash Routine ***	
FLASH	Flash monitor
Error Handling ****	
IRRCHK	Function to check diagnostic level
ERRPRT	Error printing routine
WRTTRM	Writer to terminal file or control panel
Report Writer ****	
RPTHDR	Report pagination /header writer

*** See Appendix H of this User Guide.***

***** See Aspen Plus User Models, Chapter 3.***

****** See Aspen Plus User Models, Chapter 2.***

******* See Aspen Plus User Models, Chapter 4.***

Stream Processing

In order to perform its calculations the user model must be able to read and process the Aspen Plus stream structure. The stream structure is documented in Appendix C of *Aspen Plus User Models*. Example 1 shows a USER2 model routine.

Note that data in the streams coming in and out of the model are stored in SI units.

There are several utilities available for stream processing. These perform functions such as finding the number of stream variables, i.e. the size of the stream vector, copying one stream to another, finding the total number of substreams, and finding specific substreams within a stream. Several stream handling utilities are documented in *Appendix H: Fortran Utilities* of this User Guide.

In addition to the standard composition and state information found in the stream structure, there are also component attributes. If the user model processes polymers, then component attributes must be processed and their outlet stream values must be calculated and stored. The attributes available include polymer properties such as degree of polymerization, molecular weight, polydispersity, and copolymer composition. These are documented in Section 2.2 of this User Guide. In order to process attributes, there are Fortran utilities available that perform functions such as copying attributes from one stream to another, retrieving number average molecular weight and degree of polymerization, retrieving copolymer composition, locating specific component attributes within the stream vector, and determining the size of a vector component attribute. The component attribute handling utilities are documented in Appendix H.

There are also utilities for processing components: for excluding trace components, for determining component type (polymer, oligomer, segment, catalyst), etc. These can be found with the component attribute processing utilities.

Example 1: USER2 Model Routine

```
C----------------------------------------------------------------------
      SUBROUTINE USRMOD (NMATI,  SIN,    NINFI,   SINFI,  NMATO,
     2                   SOUT,   NINFO,  SINFO,   IDSMI,  IDSII,
     3                   IDSMO,  IDSIO,  NTOT,    NSUBS,  IDXSUB,
     4                   ITYPE,  NINT,   INT,     NREAL,  REAL,
     5                   IDS,    NPO,    NBOPST,  NIWORK, IWORK,
     6                   NWORK,  WORK,   NSIZE,   SIZE,   INTSIZ,   LD)
C----------------------------------------------------------------------
C
      IMPLICIT NONE
C
C     DECLARE VARIABLES USED IN DIMENSIONING
C
      INTEGER NMATI, NINFI, NMATO, NINFO, NTOT,
     +        NSUBS, NINT,  NPO,   NIWORK,NWORK,
     +        NSIZE
C
#include "ppexec_user.cmn"
      EQUIVALENCE (RMISS, USER_RUMISS)
      EQUIVALENCE (IMISS, USER_IUMISS)
C
#include "dms_plex.cmn"
      EQUIVALENCE (IB(1), B(1))
      REAL*8 B(1)
C
#include "dms_rglob.cmn"
C
#include "dms_global.cmn"
C
#include "dms_ipoff1.cmn"
C
#include "dms_ncomp.cmn"
C
C     DECLARE FUNCTIONS
C
      INTEGER SHS_LCATT, DMS_KCCIDC
      INTEGER XMW, LMW
C
C     DECLARE ARGUMENTS
C
      INTEGER IDSMI(2,NMATI),      IDSII(2,NINFI),
     +        IDSMO(2,NMATO),      IDSIO(2,NINFO),
     +        IDXSUB(NSUBS),ITYPE(NSUBS), INT(NINT),
     +        IDS(2,3),     NBOPST(6,NPO),
     +        IWORK(NIWORK),INTSIZ(NSIZE),NREAL, LD,    I
      INTEGER KH2O
      REAL*8 SIN(NTOT,NMATI),     SINFI(NINFI),
     +       SOUT(NTOT,NMATO),    SINFO(NINFO),
     +       WORK(NWORK),  SIZE(NSIZE)
C
C     DECLARE LOCAL VARIABLES
C
      INTEGER IMISS
      REAL*8 REAL(NREAL),  RMISS, WATER
C
      INTEGER IDXP, LZMOM, LMWN, IMWN(2), IZMOM(2)
      REAL*8 AMWP, ZMOM
C     INITIALIZE ARRAY OF ATTRIBUTE NAMES
      DATA IZMOM / "ZMOM","    " /
      DATA IMWN  / "MWN ","    " /
```

continued

Example 1: USER2 Model Routine (cont.)

```
C
C----------------------------------------------------------------------
C
C     BEGIN EXECUTABLE CODE
C
C----------------------------------------------------------------------
C     OFFSETS TO COMPONENT MOLECULAR WEIGHTS
      XMW(I) = DMS_IFCMNC('MW') + I
C
C     FIRST COPY FIRST INLET TO FIRST OUTLET
C
      DO 100 I = 1, NTOT
        SOUT(I,1) = SIN(I,1)
 100  CONTINUE
C
C     INITIALIZE THE SECOND OUTLET
C
      DO 200 I = 1, NCOMP_NCC+1
        SOUT(I,2) = 0D0
 200  CONTINUE
C
      DO 300 I = NCOMP_NCC+2, NCOMP_NCC+9
        SOUT(I,2) = RMISS
 300  CONTINUE
C
C     FIND LOCATION OF COMPONENT ATTRIBUTES
C     IDXP is position of polymer component in component list.
C     Can be obtained with ispoly function
C     find location of attributes in stream
      LZMOM  = SHS_LCATT( 1, IDXP, IZMOM )
      LMWN   = SHS_LCATT( 1, IDXP, IMWN )
      IF (LZMOM .NE. 0) ZMOM = SOUT(LZMOM+1,1)
C
C     EXAMPLE OF FINDING A COMPONENT POSITION BY NAME
C
      KH2O = DMS_KCCIDC ( 'H2O' )
C
C     CAN ALSO PASS POSITION AS PARAMETER IN INT VECTOR
C     E.G. KH2O = INT(2)
      IF ( KH2O .EQ. 0 ) GO TO 999
C
C     PUT COMPONENT (WATER) IN THE SECOND OUTLET
C
      WATER = SIN(KH2O,1)
      SOUT(KH2O,1) = 0D0
      SOUT(NCOMP_NCC+1,1) = SIN(NCOMP_NCC+1,1) - WATER
      SOUT(KH2O,2) = WATER
      SOUT(NCOMP_NCC+1,2) = WATER
C
 999  RETURN
      END
```

Physical Property Calculations

Physical properties and phase equilibrium calculations can be performed within a user model. Property methods, models, and parameters specified in the input either through a built-in or a user-defined property method, can be used for the user model calculations. This can be done through property monitors. The user model requests the property of interest by calling a specific monitor, sets the state information and calculation codes in the call to the monitors, and in turn obtains thermodynamic properties such as fugacity coefficients, enthalpies, entropies, molar volumes, etc. A flash calculation routine is also available. See Table 5.16 for a listing of frequently used property monitors. The FLASH routine and the property monitors are documented in Chapters 2 and 3 of *Aspen Plus User Models*. See also User Physical Property Models later in this section.

Unit Operation Calculations

The purpose of a user unit operation block is to allow the flexibility to program user correlations or algorithms to represent a process. Independently from the physical property calculations for which monitors are provided, users can take advantage of the Fortran subroutine structure to incorporate the calculations needed to represent their process. *Aspen Plus System Management* documents programming guidelines to be followed when defining the model calculations. The calculations performed within a user unit operator model for a polymer system are similar to those that could be performed within a kinetic model. See User Kinetic Models later in this section.

Diagnostics

Throughout the simulation calculations, a user model may call the Aspen Plus error handler to issue diagnostic messages ranging from fatal errors to warnings and information. The error handler is documented in Chapter 4 of *Aspen Plus User Models*. These diagnostics can be written to the terminal or the control panel. The USER labeled commons contains output file numbers through which the terminal, control panel and simulation files can be accessed. See Appendix A of *Aspen Plus User Models* for a description of the USER labeled common.

User Unit Operation Report Writing

A report section can be included for a user model in the Aspen Plus simulation report. This requires a Fortran report writer subroutine. To write the report a report pagination utility is available. This utility is documented in Chapter 4 of *Aspen Plus User Models*. Note that in the user interface the integer and real arrays for the user model are displayed on the results screen of the user model.

USER KINETIC MODELS

User kinetic models are primarily intended for situations where the polymerization phenomena taking place are highly complex and cannot be represented by the built=in models. Users can write their own equations for the rate of change of components and the attributes of the polymer that they are intending to track. This is done through a USER reaction block. The USER block can be used in conjunction with built-in models. The user model gives the basic framework for specifying the reaction stoichiometry and the rate constant parameters. The user kinetic model requires a Fortran subroutine which performs all the computations that are required for computing the rates of change for components in the reactive phase and rates of change for polymer attributes. The structure of this subroutine is documented in Chapter 11 of *Aspen Plus User Models*. For polymerization kinetics user model, there are specific calculations that are typically performed. These include:

- Locating the polymer component attributes within the stream vector. This is done through the utility routine SHS-LACTT. Users need to determine and provide IDXP which is the component index for the polymer.

```
LDPN   = SHS_LCATT( 1, IDXP, ICATYP( 1, IDPN  ) )
LZMOM  = SHS_LCATT( 1, IDXP,ICATYP( 1, IZMOM ) )
```

- Retrieving the polymer attribute values from the stream vector SOUT. The following code shows how to retrieve DPN from SOUT. Other attributes can be similarly obtained.

```
IF( LDPN .GT. 0 .AND. SOUT(LDPN+1) .GT. 0D0) DPN = SOUT(LDPN+1)
```

- Calculating the specific volume of the reacting phase from the stream vector SOUT. From the stream vector, calculate the total number of moles and volume of the reacting phase. This example assumes that the reacting phase is a single liquid phase.

```
CALL SHS_CPACK (SOUT, NCK, IDXX, XX, TOTFLO)
        CALL PPMON_VOLL (
     +  TEMP, PRES, XX, NCK, IDXX, NBOPST, 4, 1, SVOL, DV, KER)
        VFLOW1 = SLIQRX
      VFLOW = SVOL * SOUT(NCK+1)
```

- Calculating molar concentration of each component and class 2 attributes in the reacting phase. This is obtained by dividing the mole fraction of the component in the reacting phase by the molar volume of the reacting phase. It is also shown how to compute concentration of ZMOM, a class 2 attribute for the polymer.

```
DO 50 I = 1, NC
         CONC(I) = XX(I)/SVOL
50     CONTINUE

IF(LZMOM .GT. 0  .AND. VFLOW .GT. RGLOM_RMIN)
                          ZMOM=SOUT(LZMOM+1)/VFLOW
```

- Loading the rate constants for each reaction in the reacting phase. The vector REALR will hold the values of the kinetic constants.

```
DO 200 I = 1, NR
         AK(I) = REALR(I)
200   CONTINUE
```

- Calculating the rate of reaction for each component and returning that information to the reactor. The rate equations are user derived. For example assume that the following user reactions are to be included in the user kinetics:

$$A1 + A2 \xrightarrow{k1} A3 + Waste1 \qquad k1$$

$$A3 \xrightarrow{k2} Waste2$$

The rate constants for user reactions are obtained as:

AK(1) = k1

AK(2) = k2

The reaction rate for the components (1=A1, 2=A2, 3=A3) are calculated as:

```
RATES(1) = -AK(1)*CONC(1)*CONC(2)*VFLOW
RATES(2) = -AK(1)*CONC(1)*CONC(2)*VFLOW
RATES(3) = (AK(1)*CONC(1)*CONC(2) - AK(2)*CONC(3))*VFLOW
```

- Calculating rate of change for Class 2 attributes for the polymer. The user is responsible for deriving the expression for the rate of change of attribute values.

```
DO 400 I = 1, NTCAT
       RATCAT(I) = 0D0
400 CONTINUE
C
```

The following example code explains the above steps in greater detail.

Note that the data coming in and out of the model are stored in SI units.

Example 2: User Kinetic Subroutine

```
C-----------------------------------------------------------------------------
      SUBROUTINE USRKIP (SOUT,   NSUBS,  IDXSUB,   ITYPE,  NINT,
     2                   INT,    NREAL,  REAL,     IDS,    NPO,
     3                   NBOPST, NIWORK, IWORK,    NWORK,  WORK,
     4                   NC,     NR,     STOIC,    RATES,  FLUXM,
     5                   FLUXS,  XCURR,  NTCAT,    RATCAT, NTSSAT,
     6                   RATSSA, KCALL,  KFAIL,    KFLASH, NCOMP,
     7                   IDX,    Y,      X,        X1,     X2,
     8                   NRALL,  RATALL, NUSERV,   USERV,  NINTR,
     9                   INTR,   NREALR, REALR,    NIWR,   IWR,
     *                   NWR,    WR,     NRL,      RATEL,  NRV,
     1                   RATEV)
C-----------------------------------------------------------------------------
      IMPLICIT NONE
C
C     DECLARE VARIABLES USED IN DIMENSIONING
C
      INTEGER NSUBS, NINT,  NPO,   NIWORK,NWORK,
     +        NC,    NR,    NTCAT, NTSSAT,NCOMP,
     +        NRALL, NUSERV,NINTR, NREALR,NIWR,
     +        NWR
C
#include "ppexec_user.cmn"
      EQUIVALENCE (RMISS, USER_RUMISS)
      EQUIVALENCE (IMISS, USER_IUMISS)
C
C
C
C.....RCSTR...
#include "rcst_rcstri.cmn"
#include "rxn_rcstrr.cmn"
C
C.....RPLUG...
#include "rplg_rplugi.cmn"
#include "rplg_rplugr.cmn"
      EQUIVALENCE (XLEN, RPLUGR_UXLONG)
      EQUIVALENCE (DIAM, RPLUGR_UDIAM)
C
C.....RBATCH...
#include "rbtc_rbati.cmn"
#include "rbtc_rbatr.cmn"
C
C.....PRES-RELIEF...
#include "prsr_presri.cmn"
#include "rbtc_presrr.cmn"
C
C.....REACTOR (OR PRES-RELIEF VESSEL OR STAGE) PROPERTIES...
#include "rxn_rprops.cmn"
      EQUIVALENCE (TEMP, RPROPS_UTEMP)
      EQUIVALENCE (PRES, RPROPS_UPRES)
      EQUIVALENCE (VFRAC, RPROPS_UVFRAC)
      EQUIVALENCE (BETA, RPROPS_UBETA)
      EQUIVALENCE (VVAP, RPROPS_UVVAP)
      EQUIVALENCE (VLIQ, RPROPS_UVLIQ)
      EQUIVALENCE (VLIQS, RPROPS_UVLIQS)
```

continued

Example 2: User Kinetic Subroutine (cont.)

```
C
C     INITIALIZE RATES
C
C
C     DECLARE ARGUMENTS
C
      INTEGER IDXSUB(NSUBS),ITYPE(NSUBS), INT(NINT),
     +        IDS(2),NBOPST(6,NPO),IWORK(NIWORK),
     +        IDX(NCOMP),   INTR(NINTR),  IWR(NIWR),
     +        NREAL, KCALL, KFAIL, KFLASH,NRL,
     +        NRV,   I
      REAL*8 SOUT(1),      WORK(NWORK),
     +       STOIC(NC,NSUBS,NR),  RATES(1),
     +       FLUXM(1),     FLUXS(1),     RATCAT(NTCAT),
     +       RATSSA(NTSSAT),      Y(NCOMP),
     +       X(NCOMP),     X1(NCOMP),    X2(NCOMP)
      REAL*8 RATALL(NRALL),USERV(NUSERV),
     +       REALR(NREALR),WR(NWR),      RATEL(1),
     +       RATEV(1),     XCURR
C
C     DECLARE LOCAL VARIABLES
C
      INTEGER IMISS, IDPN(2), IZMOM(2), XMW
      REAL*8 REAL(NREAL),  RMISS, XLEN,  DIAM,  TEMP,
     +       PRES,  VFRAC, BETA,  VVAP,  VLIQ,
     +       VLIQS
      DATA IDPN / "DPN ", "    " /
      DATA IZMOM / "ZMOM", "    " /
C     BEGIN EXECUTABLE CODE
C     ASSUME WE ARE USING A BATCH REACTOR. FOR OTHER REACTORS THE
C     PROCEDURE IS SIMILAR
C     OFFSETS TO COMPONENT MOLECULAR WEIGHTS
      XMW(I)=DMS_IFCMNC('MW')+I
C
C     FIND INDEX OF SPECIES BY NAME
      IDXP=DMS_KCCIDC('POLY')
C
C
C     DETERMINE POINTERS TO POLYMER ATTRIBUTES

      LDPN   = SHS_LCATT( 1, IDXP, IDPN )
      LZMOM  = SHS_LCATT( 1,  IDXP, IZMOM )
C
C     GET POLYMER ATTRIBUTES VALUES FROM SOUT
C

      IF( LDPN .GT. 0 .AND. SOUT(LDPN+1) .GT. 0D0) DPN = SOUT(LDPN+1)

C-------------------------------------------------------------------
C     GET REACTING PHASE SPECIFIC MOLAR VOLUME, SVOL ASSUMING IT IS
C     LIQUID
C
        CALL SHS_CPACK (SOUT, NCK, IDX, X, TOTFLO)
        CALL PPMON_VOLL (
     +  TEMP, PRES, X, NCK, IDX, NBOPST, 4, 1, SVOL, DV, KER)
        VFLOW1 = SLIQRX
```

continued

Example 2: User Kinetic Subroutine (cont.)

```
C
C
C     GET VOLUME OF REACTING PHASE, VFLOW
C
      VFLOW = SVOL * SOUT(NCK+1)
C

C-----------------------------------------------------------------
C
C.....CALCULATE MOLAR CONCENTRATIONS OF COMPONENTS AND CLASS 2
C     ATTRIBUTES

    DO 50 I = 1, NC
       CONC(I) = XX(I)/SVOL
50   CONTINUE
IF(LZMOM .GT. 0  .AND. VFLOW .GT. RGLOM_RMIN)
                          ZMOM=SOUT(LZMOM+1)/VFLOW

C-------------------------------------------------------------------
C     INITIALIZE THE RATES FOR COMPONENTS TO ZERO
C
      DO 100 I = 1, NC
        RATES(I) = 0D0
 100  CONTINUE
C
C-------------------------------------------------------------------
C     LOAD REACTION RATE CONSTANTS FROM THE REALR
       DO 200 I = 1, NR
         AK(I) = REALR(I)
200   CONTINUE
C
C-------------------------------------------------------------------
C     CALCULATE REACTION RATES FOR COMPONENTS
C
      DO 300 I = 1,  NC
      DO 310 J = 1,  NC
         M = COMPUTE CORRECT INDEX
        RATES(I) = RATES(I)  - AK(M) * CONC(I)*CONC(J)*VFLOW
300   CONTINUE

C
C
C    CALCULATE RATES FOR CLASS-2 ATTRIBUTE EXAMPLE
C-------------------------------------------------------------------

DO 400 I = 1, NTCAT
       RATCAT(I) = 0D0
400 CONTINUE

C
C    INITIALIZE ATTRIBUTES OF INTEREST IN THIS WAY
C    FOR ARRAY ATTRIBUTES THIS GIVES FIRST LOCATION IN ARRAY
C    RACAT(LZMOM - (NC+9) + 1) = 0

     RETURN
     END
```

USER PHYSICAL PROPERTY MODELS

There is often a need among industrial users to calculate one or more physical properties based on in-house or literature correlations and expressions that are not available in Polymers Plus. In such cases, users can take advantage of physical property user models.

A user subroutine needs to be supplied for each user model which will calculate the desired property. For each physical property, a fixed subroutine name and argument list exists; these can be found in *Aspen Plus User Models, Chapter 6*. An example of a simple user subroutine which calculates and returns the liquid molar enthalpy of a mixture (HLMX) is provided below. For instructions on how to use user physical property models from the graphical user interface, see Section 3.1.

User model development in polymer simulation is very similar to that in the simulation of standard components. In case some polymer attributes are needed for the calculation of a user property, these can be retrieved by calling the appropriate utility routine (see Table 5.16 for a summary of the utilities available). The following can be helpful while developing a physical property user model in Polymers Plus:

- The index vector, IDX, contains the indexes of the components present in the current calculation run. For example, if the first component present currently is listed third in the component list, then: IDX(1) = 3.
- Parameter values are retrieved using the utility DMS_IFCMNC. For example, suppose you want to pick up the molecular weight of a component. You need to define an integer array with elements the locations of the molecular weights of all the components in the component list on the plex vector, B:

  XMW(I) = DMS_IFCMNC('MW') + I

  Then, the molecular weight of the component listed third in the component list is B(XMW(3)).

- In polymer user models, it is often necessary to identify whether a particular component is polymer, oligomer, or segment. This is done by the utility logical functions ISPOLY, ISOLIG, and ISSEG. For instance, suppose you want to perform a certain manipulation on the polymer components present in your run:

  ```
  IF (ISPOLY(I)) GO TO 10
  ```

  Which will send the calculation to line number 10 if the component with index I is a polymer component.

- The mole fraction vector X (or Z) is based on the apparent molecular weight of the polymer components. If you need to perform calculations for a polymer run where the mole fractions are needed, then you must use the true mole fractions (which are based on the true molecular weight of the polymer) rather than the apparent mole fractions X. This is done by a conversion utility routine called XATOXT:

  ```
  CALL XATOXT( N, IDX, XMW, X, XTRUE)
  ```

Where: XMW is the vector of the apparent molecular weights, IDX is the index vector, X is the stream apparent mole fraction vector, and XTRUE is the vector which contains the mole fractions based on the true molecular weight of the polymer.

- Polymer attributes needed for calculations in user physical property models are retrieved using utility subroutines. For a list of available utilities see Table 5.16. As an example, to get the number average degree of polymerization, DPn, for a particular component you must give:
  `CALL GETDPN( 1, 1, I, DPN )`

  Where I is the component index. For a detailed description of all the polymer utilities available see Appendix H.
- Users can call several Aspen Plus subroutines to perform specific tasks. For example, routine IDLGAS will return the ideal-gas properties of the components and their mixture, while PL001 will return the vapor pressure of the desired components (see in *Aspen Plus User Models, Chapter 3*).
- After calculating a molar property, the appropriate conversion must be made so that the returned property is based on the apparent mole basis. For instance, after the calculation of the liquid enthalpy of a polymer component based on the true molecular weight, the following conversion should be made:

  HL_app = HL_true * MW_app / MW_true

A sample user subroutine that calculates and returns the mixture liquid enthalpy is given in the Example 3.

Note that the data coming in and out of the model are stored in SI units.

Example 3: User subroutine for mixture liquid enthalpy calculation

```
C----------------------------------------------------------------------
      SUBROUTINE HL2U (T     ,P     ,Z     ,N     ,IDX   ,
     1                 IRW   ,IIW   ,KCALC ,KOP   ,NDS   ,
     2                 KDIAG ,QMX   ,DQMX  ,KER   )
C
C----------------------------------------------------------------------
C    HV2U IS A USER MIXTURE ENTHALPY SUBROUTINE
C
C   THIS USER SUBROUTINE CALCULATES THE LIQUID ENTHALPY OF A BINARY
C   MIXTURE CONTAINING ONE POLYMER AND ONE SOLVENT.
C
C
C     NAME OF MODULE: HL2U
C
C
       IMPLICIT NONE
C
      DIMENSION Z(N), IDX(N), KOP(10)
      DIMENSION D(15)
```

continued

Example 3: User subroutine for mixture liquid enthalpy calculation (cont.)

```
C... USER DIMENSION
      DIMENSION XTRUE(10)
C
C
#include "dms_ncomp.cmn"
#include "ppexec_user.cmn"
#include "dms_plex.cmn"
C
      EQUIVALENCE (IB(1), B(1))
      INTEGER XMW, DHFORM, CPIG, II, DMS_IFCMNC
      INTEGER IMON, IPOL, IIMON, IIPOL, I, N, J, ISEG
      REAL*8 DELT1, DELT2, DELT3, DELT4, H_MON, H,POL,
     *       HM_MIX, AVG_MW, T, TREF, QMX
C
C----------------------------------------------------------------------
C
C  STATEMENT FUNCTIONS FOLLOW
C
      XMW(I) = DMS_IFCMNC('MW') + I
      DHFORM(I) = DMS_IFCMNC('DHFORM') + I
      CPIG(I,J) = DMS_IFCMNC('CPIG') + 11*(J - 1) + I
C
C     *** NOTE *******************************************
C
C     PARAMETERS ARE LOCATED USING THE UTILITY DMS_IFCMNC
C     AND THE NAME OF THE PARAMETER. FOR EXAMPLE,
C     DMS_IFCMNC('MW') RETRIEVES THE LOCATIONS WHERE THE
C     COMPONENT MOLECULAR WEIGHTS ARE STORED.
C
C     ****************************************************
C
      DO 100 I=1,10
      XSEG(I) = 0.D0
 100  CONTINUE
C
      TREF = 298.15
C
C----------------------------------------------------------------------
C
C    *** NOTE *******************************************
C    COMPONENT ID FOR MONOMER *HARD-WIRED* AT POSITION 2
C    COMPONENT ID FOR POLYMER *HARD-WIRED* AT POSITION 3
C    ****************************************************
C
       IMON = 2
       IPOL = 3
       ISEG = 4
C
C
C## BOTH Z AND XSEG ARE PACKED: XSEG(IPOL) CONTAINS MOLE FRAC OF SEGMENT
C
      CALL XATOXT(N, IDX, B(XMW(1)), Z, XTRUE)
C
C     POLYMERIC SPECIES PROP-SET PROPERTIES
C
      DELT1 = T - TREF
      DELT2 = (T**2 - TREF**2)/2.D0
      DELT3 = (T**3 - TREF**3)/3.D0
```

continued

Example 3: User subroutine for mixture liquid enthalpy calculation (cont.)

```
      DELT4 = (T**4 - TREF**4)/4.D0
      H_MON = B(DHFORM(IMON)) + B(CPIG(1,IMON))*DELT1 +
     + B(CPIG(2,IMON))*DELT2 + B(CPIG(3,IMON))*DELT3 + B(CPIG(4,IMON))
     +*DELT4
      H_POL = B(DHFORM(IPOL)) + B(CPIG(1,IPOL))*DELT1 +
     + B(CPIG(2,IPOL))*DELT2 + B(CPIG(3,IPOL))*DELT3 + B(CPIG(4,IPOL))
     +*DELT4
C
C     *** NOTE ********************************************
C     IN CASE A COMPONENT ATTRIBUTE WAS NEEDED FOR THE
C     CALCULATION OF THE POLYMER ENTHALPY, THE APPROPRIATE
C     UTILITY ROUTINE SHOULD BE CALLED.
C
C     FOR EXAMPLE, SUPPOSE THE NUMBER-AVERAGE DEGREE OF
C     POLYMERIZATION (DPn) OF THE POLYMER WAS NECESSARY.
C     THE UTILITY ROUTINE GETDPN CAN BE USED TO RETURN
C     THE DESIRED ATTRIBUTE:
C
C   CALL GETDPN (1, 1, IPOL, DPN)
C
C   THE ARGUMENTS HAVE THE FOLLOWING MEANING:
C
C   1    = CONVENTIONAL SUBSTREAM
C   1    = DPN FOR 1 COMPONENT IS REQUESTED (NCP=1)
C   IPOL = POLYMER COMPONENT INDEX
C   DPN  = RETURNED VALUE OF THE NUMBER AVERAGE
C          DEGREE OF POLYMERIZATION
C
C     ****************************************************
C
      IIMON = 0
      IIPOL = 0
      DO 10 I=1,N
      II = IDX(I)
      IF (II.EQ.IMON) IIMON = I
      IF (II.EQ.IPOL) IIPOL = I
 10   CONTINUE
C
      HM_MIX  = H_MON*XTRUE(IIMON) + H_POL*XTRUE(IIPOL)
      AVG_MW  = B(XMW(IMON))*Z(IIMON) + B(XMW(IPOL))*Z(IIPOL)
C
C
C     CONVERT FROM TRUE TO APPARENT MOLE BASIS
      QMX = HM_MIX * AVG_MW / B(XMW(ISEG))
C
C
 999  CONTINUE
      RETURN
      END
```

REFERENCES

Aspen Plus Reference Manual, "User Models," Aspen Technology, Inc. (1998).

Aspen Plus Reference Manual, "System Management," Aspen Technology, Inc. (1998).

5.4 APPLICATION TOOLS

This section discusses the tools available for applying Polymers Plus features to solve real-life problems.

The topics covered include:

- Example Applications for a Simulation Model
- Application Tools Available in Polymers Plus
- Model Variable Accessing

EXAMPLE APPLICATIONS FOR A SIMULATION MODEL

The main purpose of a simulation model is to provide the engineer with a deeper understanding of the molecular and macroscopic processes which are vital to a polymer manufacturing process. This understanding will eventually lead to improvements in various aspects of the process related to safety, productivity, and polymer product quality. These are some typical scenarios in which a simulation model is used to meet this objective.

A model may be used to:

- Identify superior grade transition policies and better plant startup and shutdown procedures which minimize offspec polymer product
- Reduce the number of lengthy and costly experiments on bench, pilot, and plant scale for polymer product and polymerization process development
- Train process engineers, chemists, plant operators
- Identify sources of variance in polymer product quality
- Provide data for the design of rupture discs and vent lines
- Find optimal temperature profiles for a continuous reactor train which minimize reaction medium viscosity while meeting product specifications
- Investigate monomer feed policies for a semi-batch copolymerization process for keeping the chemical composition distribution narrow
- Design a free-radical initiator mix to maximize productivity under the constraints of safe reactor operations

APPLICATION TOOLS AVAILABLE IN POLYMERS PLUS

Several analysis and flowsheeting tools are available in Polymers Plus to configure a model for performing analyses and studies of a process. These include:

- Fortran - used to incorporate Fortran calculations in the simulation
- DESIGN-SPEC - used to apply specifications on process variables
- SENSITIVITY - used to examine the effect of varying one or more process variables
- CASE-STUDY - used to compare between different sets of operating conditions
- OPTIMIZATION - used to perform optimization calculations

For each of these tools, with the exception of Fortran, Aspen Plus sets a loop around a model, flowsheet section, or entire flowsheet. Within this loop, selected operating variables are manipulated and key process variables are sampled. This is illustrated in Figure 5.20. The categories of accessible flowsheet variables are described in Model Variable Accessing.

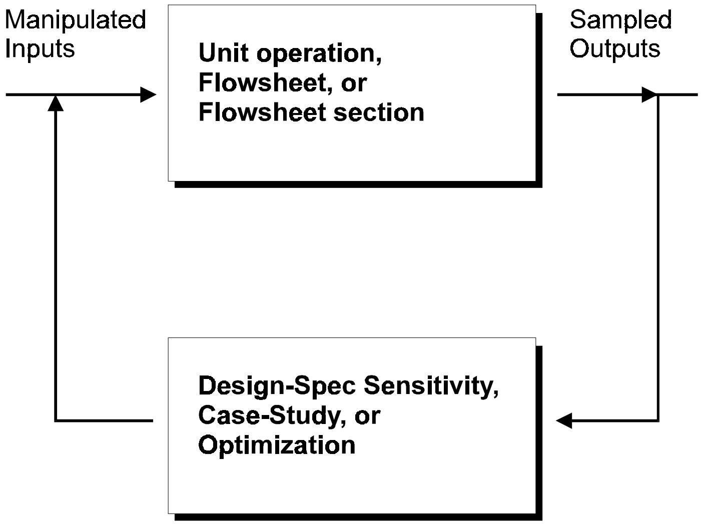


Figure 5.20 Calculation Procedure for Analysis and Flowsheeting Tools

Note that in most cases Aspen Plus automatically generates the calculation sequence. You can also specify a sequence manually. For details on how use these tools in your simulations, see the *Aspen Plus User Guide*. Example uses of these features are given in the *Polymers Plus Examples and Applications Case Book.*

Fortran

Fortran blocks provide a mechanism for you to incorporate Fortran statements into the flowsheet calculations. This can be used calculate and set input variables based on special user inputs. For this reason, Fortran blocks can be used as feed-forward controllers. You can also use Fortran blocks to calculate and write results to the Aspen Plus report, control panel, or external file.

To use this block you must specify which model variables to sample or manipulate, enter the Fortran statements, and set the sequence in which the block must be executed during the flowsheet calculations.

An example use of a Fortran block as a feed-forward controller would be to hold the flowrate of a catalyst proportional to a monomer flow for a situation where that monomer flow varies.

DESIGN-SPEC

Design-Spec blocks allow you to set a process variable which is normally calculated during the simulation. For each specification, you must identify which process variable can be adjusted to meet that specification. For this reason, Design-Spec blocks can be used as feedback controllers.

To use this block you must specify which model variables must be fixed, what values they must be fixed at, and which model input variables can be manipulated. You can include Fortran statements in Design-Spec blocks.

An example use of a Design-Spec block would be to set a maximum amount for impurities in a product stream.

SENSITIVITY

Sensitivity blocks provide a mechanism for you to analyze the effect of operating variables which you select on the process. This block generates a matrix of manipulated variables versus sampled variables. If there is more than one manipulated variable, the sensitivity analysis is performed for each combination of manipulated variables. It is recommended that you use multiple Sensitivity blocks if you do not want to combine the manipulated variables.

To use this block you must specify which are the manipulated variables, which are the sampled variables, and how they must be tabulated. You can include Fortran statements in Sensitivity blocks.

An example use of a Sensitivity block would be to determine the effect of reactor temperature or pressure on the polymer product properties.

CASE-STUDY

Case-Study blocks provide another mechanism for you to analyze the effect of operating conditions on process variables. They allow you to make several runs in series for the entire flowsheet. Since a report is generated for the whole flowsheet for each case, you do not need to specify output variables to be sampled.

To use this block you must identify the case study variables, assign values for these variables, and specify reporting options.

An example use of Case-Study would be to investigate the effect of changing feed conditions and composition on key process variables.

OPTIMIZATION

Optimization blocks provide a mechanism for you to minimize or maximize an objective function calculated using key process variables. To define the objective function you would use Fortran statements.

To use this block you must define the objective function, specify manipulated variables, and define constraints, if any.

An example use of Optimization would be to find the optimal reactor temperature to meet polymer product property specifications while minimizing reaction medium viscosity.

MODEL VARIABLE ACCESSING

When using the various model analysis tools to perform sensitivity studies, optimization studies, case studies, or data fitting, or when applying design specifications, or adding in-line Fortran to a simulation model, users must access many different flowsheet variables. These flowsheet variables are grouped by type:

- Unit operation block variable
- Stream variable (including polymer component attributes)
- Reaction variable
- Physical property variable

A list of such variables is given in Table 5.13.

Table 5.13 Accessible Variables (Partial Listing)

Variable Type	Identifier	Description
Block	BLOCK-VAR	Unit operation block variable
		Unit operation block vector
Stream	STREAM-VAR	Non component dependent stream variable
	MOLE-FLOW	Component mole flow
	MOLE-FRAC	Component mole fraction
	MASS-FLOW	Component mass flow
	MASS-FRAC	Component mass fraction
	STDVOL-FLOW	Component standard volume flow
	STDVOL-FRAC	Component standard volume fraction
	STREAM-PROP	Stream Prop-Set property
	STREAM-VEC	Entire stream vector
	SUBSTRM-VEC	Entire substream vector
	COMPATTR-VAR	Component attribute element (Notes 1-4)
	COMPATTR-VEC	Component attribute (Notes 1-4)
	SUBSATTR-VAR	Substream attribute element
	SUBSATTR-VEC	Substream attribute
Reaction	REACT-VAR	Reaction variable (Note 5)
Physical	UNARY-PARAM	Unary physical property parameter
Properties	BI-PARAM	Binary physical property parameter

Notes

1. Component attributes may be accessed in several ways. They may be accessed through STREAM-VEC or through SUBSTRM-VEC. In this case, users are responsible for locating the desired attribute and attribute element within the stream or substream vector. See Table 5.14 for the MIXED substream vector structure.
2. Component attributes may also be accessed with COMPATTR-VAR. With COMPATTR-VAR, users must provide the element number for attributes having more than one element. See Section 2.2 to find out the dimensions of polymer component attributes. If the attribute is dimensioned by number of polymer segments, NSEGS, (e.g. SFLOW, or SFRAC polymer attributes), the ordering of elements follows the order in which the list of polymer segments was specified (See Section 2.1). For component attributes dimensioned by number of catalytic sites, each element represents a site number, i.e. site no. 1, no. 2, etc. For two-dimensional component attributes dimensioned by number of segments and number of catalytic sites (NSEGS*NSITES), the first dimension is NSEG, therefore, the ordering of elements is as follows: the list of specified segments is repeated for each site beginning with site no. 1.
3. Component attributes may also be accessed with COMPATTR-VEC. In this case, users are not required to provide an element number since the whole component attribute is returned as a vector having one or more elements. The ordering of elements within the attribute vector follows the description given in Note 2.
4. COMPATTR-VAR and COMPATTR-VEC are equivalent for component attributes having only one element.
5. REACT-VAR may be used to access kinetic constant parameters for reaction kinetic models, including free-radical, step-growth and Ziegler-Natta. The type of information required to access these parameters is model dependent. For free-radical, the reaction type (INIT-DEC, for example), and the reacting species are required, in addition to the name of the parameter to be accessed. The same is true for Ziegler-Natta which also requires a catalyst site type number. For step-growth, a reaction number is required. For the standard Aspen Plus reaction models, a reaction number, and/or substream identifier may be needed to locate the parameters.

Table 5.14 MIXED Substream Structure

Array Index	Description
1, ..., NCC	Component mole flows (kgmole/sec)
NCC + 1	Total mole flow (kgmole/sec)
NCC + 2	Temperature (K)
NCC + 3	Pressure (N/m^2)
NCC + 4	Mass enthalpy (J/kg)
NCC + 5	Molar vapor fraction
NCC + 6	Molar liquid fraction
NCC + 7	Mass entropy (J/kg-K)
NCC + 8	Mass density (kg/m^3)
NCC + 9	Molecular weight (kg/kgmole)
NCC + 10	$\left.\begin{array}{l}\text{value}_1\\ \\ \text{value}_{ncat1}\end{array}\right\}$ Values for component attribute 1 of component 1 (polymer or other attributed component)
	$\left.\begin{array}{l}\text{value}_1\\ \\ \text{value}_{ncat1}\end{array}\right\}$ Values for component attribute 2 of component 1 (polymer or other attributed component)
	$\left.\begin{array}{l}\text{value}_1\\ \\ \text{value}_{ncat1}\end{array}\right\}$ Values for component attribute 1 of component 2 (polymer or other attributed component)

Note

NCC is the number of conventional components (including polymers, segments and oligomers) entered on the Components Specifications Selection sheet. This parameter is stored as NCOMP_NCC in labeled common DMS_NCOMP (See Aspen Plus User Models Reference Manual, Appendix A).

REFERENCES

Aspen Plus User Guide, Version 10.0, Aspen Technology, Inc. (1998).

Convergence and Optimization in Aspen Plus, Aspen Technology, Inc., Course notes.

6 RUN-TIME ENVIRONMENT

This chapter discusses various topics related to working in the Polymers Plus environment.

The topics covered include:

- Polymers Plus Architecture
- Installation Issues
- Configuration Tips
- User Fortran
- Troubleshooting Guide
- Documentation and Online Help

POLYMERS PLUS ARCHITECTURE

Polymers Plus is a layered product. In other words, this product works in conjunction with a main program. This main program is Aspen Plus for steady-state simulation, or Aspen Custom Modeler for dynamic simulation. Polymers Plus brings to these simulators the polymer process technology in the form of component characterization, physical property models and databanks, kinetic models, and the associated input forms. Figure 6.1 illustrates the overall architecture.

As a result of this layered architecture the installation and configuration of Polymers Plus is closely tied to that of Aspen Plus for steady-state simulation and that of Aspen Custom Modeler for dynamic simulation. In this chapter we will focus on topics related to the Aspen Plus environment.

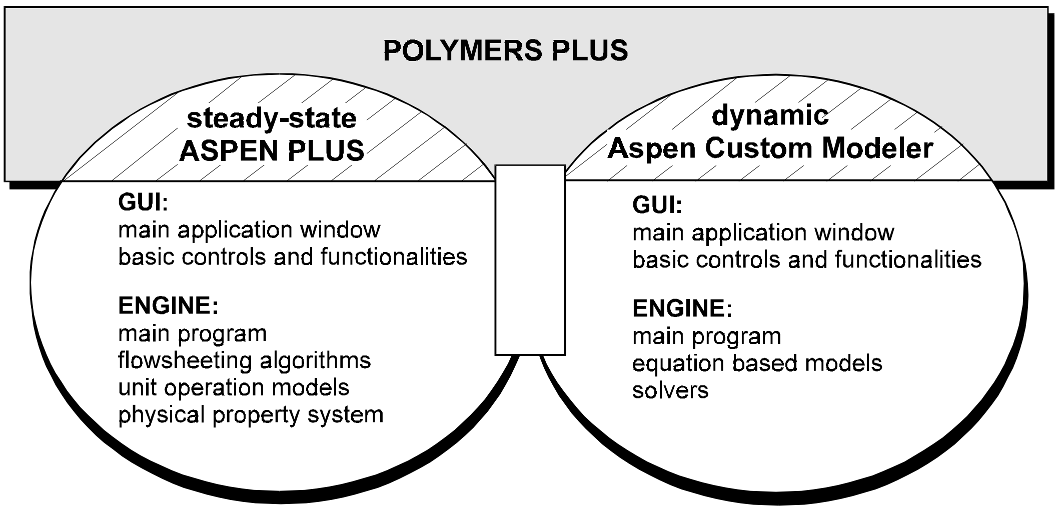


Figure 6.1 *Polymers Plus Architecture*

INSTALLATION ISSUES

Hardware Requirements

Polymers Plus is available on all the hardware platforms supported by Aspen Plus. These are Windows (WinNT and Win95) for the user interface, and Windows, Unix platforms, and VMS for the simulation engine. Consult the *Aspen Plus Installation Guide* for your hardware platform for the hardware requirements.

Installation Procedure

When installing the Aspen Plus graphical user interface select the Polymers Plus layered product installation option. If you hold a valid license this will give you access to the Polymers Plus user interface add-ons: input forms, databanks, online documentation, plotting capabilities, etc.

When installing the Aspen Plus licensing options you must ensure that you enter the Polymers Plus license authorization. Otherwise you will be unable to access Polymers Plus during the simulation calculations. For details about installing layered products, consult the *Aspen Plus Installation Guide* for your hardware platform.

CONFIGURATION TIPS

Startup Files

The information needed to launch the main Aspen Plus application window is recorded in startup files. These files define the type of simulation, default settings for the user interface, hosts for the simulation engine, run settings, etc. One type of startup file is used to define defaults for the type of simulation. This is the simulation template.

Simulation Templates

Simulation templates are available to help you get started setting up your model. These templates typically contain options such as unit sets, physical property method selection, and Table File Format (TFF) selection for stream result tables. Polymer simulation templates are available. You can create your own personal template to allow faster definition of a new simulation model. To use a simulation template, after starting Aspen Plus, on the startup box select the template startup option. Then choose one of the polymer simulation templates. This will automatically setup a global unit set, an appropriate polymer property method, and a polymer TFF for the stream tables.

To learn more about TFF files see the *Aspen Plus System Management Reference Manual*.

USER FORTRAN

User Fortran Templates

There are several ways for you to customize your models by adding calculations in Fortran. Section 2.4 described how to setup user a Prop-Set for calculating end-use properties. Section 5.3 described how to setup user unit operation models, user kinetic models, and user property models. Templates are available for your use in creating the Fortran files used in these features. You will find these templates in the following locations:

Version	Location
Windows	%asptop%\user
UNIX	$ASPDIR/user
VMS	ASP$ROOT:[USER]

User Fortran Linking

User Fortran calculations in the form of user routines are linked dynamically to Polymers Plus during a simulation. Within user Fortran, you will often access utilities located within Polymers Plus. In order to access these utilities, you will need to know the name of the object libraries where they are located. This applies to the utilities described in Appendix H. The name of the utility as shown in the example call sequence includes the name of the object library where it is located.

You can also create your own dynamic link libraries to hold your user Fortran files. Chapter 2 of *Aspen Plus System Management*, describes how to work with Fortran code modifications.

TROUBLESHOOTING GUIDE

Following are tips to help you diagnose and resolve problems you may run into while setting up or running Polymers Plus.

User Interface Problems

Here is a list of symptoms relating problems you may encounter when using the user interface. Possible causes and solutions are given for each symptom.

- The polymer input forms cannot be found on the GUI
- A file created without using polymer features appears incomplete in the components record
- Windows crashes during input specifications
- Windows crashes during simulation calculations
- Windows crashes after simulation is complete

Symptom: The polymer input forms cannot be found on the GUI.

Cause: The installation was not complete.

Solution: You must locate your installation CD and do an incremental installation of Polymers Plus. Select the custom installation and choose the Polymers Plus layered product.

– or –

Cause: Polymers Plus is installed but not enabled.

Solution: Enable Polymers Plus. From the Tools menu, select Options. On the Startup tab there is a box entitled **Enable forms for layered products**. Make sure you select Polymers Plus.

Symptom: A file created without using polymer features appears incomplete in the components record.

Cause: You visited the polymer record while creating the file, then later switched off Polymers Plus.

Solution: You must enable Polymers Plus (From the Tools menu, Select Options, click on the Startup tab). In the Data Browser, select Polymers (Polymers will appear as incomplete), right mouse click, select Delete.

Symptom: Windows crashes during input specifications.

Cause: An invalid operation was performed either by the Aspen Plus program or by another program running simultaneously.

Solution: Usually, when you crash, a backup file is created. Startup Aspen Plus again, then you should be able to recover your file. If the invalid operation was caused by Aspen Plus, repeat the input steps that lead to the crash, verify that it is reproducible, and submit the problem to Technical Support.

Symptom: Windows crashes during simulation calculations.

Cause: The simulation engine encountered an error that could not be transferred to the GUI.

Solution: Export an input summary. Run the input summary alone, then examine the run history for simulation errors. Change the input specifications associated with the error and rerun.

– **or** –

Cause: Aspen Plus ran out of resources to create run files. This can happen especially for large simulations. You may see error messages referring to the amount of virtual memory available.

Solution: Free-up some disk space and run again. Also, consult the *Aspen Plus System Management Reference Manual, Chapter 3*. An entire section is devoted to managing virtual memory on Win95 and WinNT.

– **or** –

Cause: Aspen Plus ran out of memory to load dynamic link libraries.

Solution: Free-up some disk space or increase the amount of memory available to the application then run again.

Symptom: Windows crashes after simulation is complete.

Cause: Aspen Plus could not load the simulation results.

Solution: If you are running on a remote hosts, there may have been a communication failure at the end of the simulation calculations. You can submit the run again or you can manually load the results file (.SUM) from the remote host.

– **or** –

If you are running on a local PC host, Aspen Plus may have run out of memory to load the results. Free-up some disk space or increase the amount of memory available to the application and run again.

– **or** –

If the load failure was not due to any of the above, there may be some information recorded in the results file (.SUM) which is causing the problem. Contact Technical Support and be prepared to supply the results file and/or your saved simulation file.

Simulation Engine Run-Time Problems

Here is a list of symptoms relating to problems you may encounter with the simulation engine at run-time. Possible causes and solutions are given for each symptom.

- During simulation calculations an error message occurs for a license failure
- A message box comes up stating that an error occurred in the Aspen Plus engine
- A run history message appears referring to a dynamic load module error
- A run history message appears which refers to "Virtual Memory Exhausted"
- After one run a subsequent run following an input change crashes

---

Symptom: During simulation calculations an error message occurs for a license failure.

Cause: The application could not find a valid free license to complete the simulation.

Solution: If the license error message refers to "Feature 10". This means that you do not have a license for Aspen Plus itself. If you are using a licensed installation, then this could be a temporary license failure. This can happen for multi-user sites, or if you are using a license manager located on a network. In that case, you simply need to try again later.

– **or** –

If you are using an installation with a single activator, then your license key file may be corrupted, the port where the activator is plugged in could be damaged, or the activator could be damaged. To correct your license key files, perform a license key installation again. If the problem is your activator, contact Technical Support to have it replaced.

– **or** –

If the license error message refers to another feature number, you may still have run into a temporary license failure (see above). In that case, try again. If this was not a temporary license failure, then you created a simulation file which uses features for which you are not licensed. If the message refers to "Feature 15", then you are trying to use Polymers Plus without a valid license. Other feature numbers refer to specific add-on products. You must contact AspenTech to obtain a valid Polymers Plus license.

Symptom: A message box comes up stating that an error occurred in the Aspen Plus engine.

Cause: See "Windows crashes during simulation calculations" under User Interface Problems. See also "After one run a subsequent run following an input change crashes" in this section.

Solution: See "Windows crashes during simulation calculations" under User Interface Problems. See also "After one run a subsequent run following an input change crashes" in this section.

Symptom: A run history message appears referring to a dynamic load module error.

Cause: Aspen Plus ran out of resources to load dynamic link libraries.

Solution: See "Windows crashes during simulation calculations" under User Interface Problems.

– or –

Cause: You are referencing user Fortran and do not have the compiled object file in your working directory. The working directory is the location from which you opened an existing file. If you created a file from a template or opened an existing file from a floppy or a write protected area (e.g. \xmp or \app) the working directory is as specified in Tools Options Startup.

Solution: Compile the user Fortran and place it in your run directory.

Symptom: A run history message appears which refers to "Virtual Memory Exhausted".

Cause: You ran out of virtual memory space to load the run files.

Solution: See the *Aspen Plus System Management Reference Manual, Chapter 3* which discusses virtual memory management.

Symptom: After one run a subsequent run following an input change crashes.

Cause: The problem size has changed as a result of the input or for other reasons Aspen Plus unsuccessfully tried to reuse the previous run data space. Usually an error message appears which states that a "Fatal error has been encountered".

Solution: Usually after the crash you should be able to recover your file and run with the input change. To prevent this from happening for the same run, reinitialize the simulation before making repeated runs. This is still a problem which should be reported to Technical Support.

DOCUMENTATION AND ONLINE HELP

Polymers Plus is available to you in electronic format. Your installation package includes a documentation CD-ROM. On there you will find this Polymers Plus User Guide, Volumes I and II, and also the Polymers Plus Examples and Applications Case Book. These are delivered in the Portable Document Format (PDF). They can be read using the Adobe Acrobat Reader®.

In addition, you will find Polymers Plus online help from within the application. This online help can be accessed from the main application window Help menu Help Topics Contents. In the Help contents go to "Using Polymers Plus" for step-by-step procedural instructions. Online Help can also be accessed through the fully integrated search index and Find menu. You can also request help from within the polymer input forms using the help key, F1, or using the "What's This" button

↖?.

REFERENCES

Aspen Plus Installation Guides for Windows, Unix and VMS, Version 10.0, Aspen Technology, Inc. (1998).

Aspen Plus Reference Manual, "System Management", Aspen Technology, Inc. (1998).

Aspen Plus User Guide, Version 10.0, Aspen Technology, Inc. (1998).

A COMPONENT DATABANKS

This appendix documents the Polymers Plus component databanks. There are currently two databanks available:

- POLYMER Databank - containing polymer pure component parameters (A•2)
- SEGMENT Databank - containing segment pure component parameters (A•8)

In addition users may retrieve parameters from the Aspen Plus databanks.

PURE COMPONENT DATABANK

The pure component databanks contain pure component data for over 1500 species. Typically components such as monomers, solvents, catalysts, initiators, etc. would be retrieved from the pure component databanks. The parameters in these databanks include those listed in Table A.1.

POLYMER DATABANK

POLYMER contains property parameters for polymers. The parameters stored in the databank are in Table A.1. The list of polymers contained in the databank is in Table A.2.

Note that a generic polymer component is available in the databank for custom designed polymers.

Table A.1 POLYMER Property Parameters

Parameter	No. Element	Description
CPIG	11	Ideal gas heat capacity
DGFVK	1	Free energy of formation, ideal gas reference state
DHFVK	1	Heat of formation, ideal gas reference state
DHVLWT	5	Heat of vaporization
MW*	1	Polymer reference molecular weight
OMEGA	1	Acentric factor
PC	1	Critical pressure
PLXANT	9	Antoine coefficient
TC	1	Critical temperature
VC	1	Critical volume
VLTAIT	9	Tait molar volume model coefficients
ZC	1	Critical compressibility factor

**** MW is a reference molecular weight calculated as the average segment molecular weight using:***

$$MW = \frac{\sum MWSEG}{NSEG}$$

For the generic polymer component MW is set to 1.

Table A.2 POLYMER Databank Components

Alias	Polymer Name
ABS	Acrylonitrile-butadiene-styrene
BR-1	Poly(butadiene)
CA-1	Cellulose-acetate
Cellulose	Cellulose
Chitosan	Chitosan
CPE	Chlorinated-Poly(ethylene)
CTA	Cellulose-triacetate
Dextran	Dextran
EVA	Ethylene-vinyl-acetate
EEA	Ethylene-ethyl-acrylate
EPR	Ethylene-propylene
HDPE	High-density-Poly(ethylene)
Heparin	Heparin
Hyaluronic	Hyaluronic-Acid
I-PB	Isotactic-Poly(1-butene)
I-PMMA	Isotactic-Poly(methyl-methacryl)
I-PP	Isotactic-Poly(propylene)
Keratan	Keratan-Sulfate
LDPE	Low-density-poly(ethylene)
LLDPE	Linear-low-density-poly(ethylene)
NBR	Nitrile-butadiene-rubber
NYLON6	Nylon-6
NYLON66	Nylon-66
PAA	Poly(acrylic-acid)
P(ACA&S)	Poly(acrylamide-styrene)
PALA	Poly(alanine)
PAMIDE	Poly(amide)
PAMS	Poly(alpha-methylstyrene)
P(AMS&AN)	Poly(a-methylstyrene-AN)
PAN	Poly(acrylonitrile)

continued

Table A.2 POLYMER Databank Components (cont.)

Alias	**Polymer Name**
PARA	Poly(acrylamide)
PARG	Poly(arginine)
PASN	Poly(asparagine)
PASP	Poly(aspartic-acid)
PB-1	Poly(1-butene)
PBA	Poly(n-butyl-acrylate)
PBMA	Poly(n-butyl-methacrylate)
P(BMA&S)	Poly(n-butyl-methac-styrene)
PBS-1	Poly(butadiene-styrene)
PBT	Poly(butylene-terephthalate)
PC-1	Poly(carbonate)
P(C&DMS)	Poly(carbonate-dimet-siloxane)
PCHMA	Poly(cyclohexyl-methacrylate)
PCYS	Poly(cysteine)
PD-1	Poly(decene-1)
PDMA	Poly(decyl-methacrylate)
PDMS	Poly(dimethylsiloxane)
P(DMS&S)	Poly(dimethylsiloxane-styrene)
PE	Poly(ethylene)
PEA	Poly(ethyl-acrylate)
PEEK	Poly(ether-ether-ketone)
PEG	Poly(ethylene-glycol)
P(EG&PG)	Poly(eth-glycol-prop-glycol)
PEMA	Poly(ethyl-methacrylate)
PEO	Poly(ethylene-oxide)
P(EO&POX)	Poly(eth-oxide-prop-oxide)
P(E&P)	Poly(ethylene-propylene)
PET	Poly(ethylene-terephthalate)
P(E&VAC)	Poly(ethylene-vinyl-acetate)
PGLN	Poly(glutamine)
PGLU	Poly(glutamic-acid)

continued

Table A.2 POLYMER Databank Components (cont.)

Alias	Polymer Name
PGLY	Poly(glycine)
PH	Poly(heptene-1)
PHA	Poly(n-hexyl-acrylate)
PHENOXY	Phenoxy
PHIS	Poly(histidine)
PHMA	Poly(n-hexyl-methacrylate)
PI	Poly(imide)
PIB	Poly(isobutylene)
PIBMA	Poly(isobutyl-methacrylate)
PILE	Poly(isoleucine)
PIP-1	Poly(isoprene)
PLEU	Poly(leucine)
PLYS	Poly(lysine)
PMA	Poly(methyl-acrylate)
P(MAA&MMA)	Poly(methac-acid-met-methac)
P(MAA&S)	Poly(methac-acid-styrene)
P(MAA&VAC)	Poly(methac-acid-vin-acetate)
PMET	Poly(methionine)
PMMA	Poly(methyl-methacrylate)
PMMS	Poly(m-methylstyrene)
PMP	Poly(4-methyl-1-pentene)
PMVPD	Poly(2-methyl-5-vinylpyridine)
PNA	Poly(sodium-acrylate)
POCS	Poly(o-chlorostyrene)
POE	Poly(oxyethylene)
POLYMER	Generic polymer component
POM	Poly(oxymethylene)
POMS	Poly(o-methylstyrene)
POP	Poly(oxypropylene)
PP	Poly(propylene)

continued

Table A.2 POLYMER Databank Components (cont.)

Alias	Polymer Name
PPA	Poly(n-propyl-acrylate)
PPBRS	Poly(p-bromostyrene)
PPEMA	Poly(n-pentyl-methacrylate)
PPG	Poly(propylene-glycol)
PPHE	Poly(phenylalanine)
PPO	Poly(phenylene-oxide)
PPMA	Poly(n-propyl-methacrylate)
PPMOS	Poly(p-methoxystyrene)
PPMS	Poly(p-methylstyrene)
PPOX	Poly(propylene-oxide)
PPRO	Poly(proline)
PPS	Poly(phenylene-sulfide)
PS-1	Poly(styrene)
PSBMA	Poly(sec-butyl-methacrylate)
PSER	Poly(serine)
PSF	Poly(sulfone)
P(S&VP)	Poly(sytrene-vinylpyrrolidone)
P(S&VPD)	Poly(styrene-4-vinylpyridine)
PT-1	Poly(tetrahydrofuran)
PTFE	Poly(tetrafluoroethylene)
PTHR	Poly(threonine)
PTRP	Poly(tryptophan)
PTYR	Poly(tyrosine)
PU-1	Poly(urethane-fiber)
PVA	Poly(vinyl-alcohol)
PVAC	Poly(vinyl-acetate)
P(VAC&VAL)	Poly(vin-acetate-vin-alcohol)
PVAL	Poly(valine)
PVAM	Poly(vinyl-amine)
PVC	Poly(vinyl-chloride)
PVCAC	Poly(vin-chloride-vin-acetate)

continued

Table A.2 POLYMER Databank Components (cont.)

Alias	Polymer Name
PVDC	Poly(vinylidene-chloride)
PVDF	Poly(vinylidene-fluoride)
PVF	Poly(vinyl-fluoride)
PVI	Poly(vinyl-isobutyl-ether)
PVME	Poly(vinyl-methyl-ether)
PVO	Poly(vinylpropionate)
PVP	Poly(vinylpyrrolidone)
PVPD	Poly(4-vinyl-pyridine)
SAN	Styrene-acrylonitrile
SBR	Styrene-butadiene-rubber
UF	Urea-formaldehyde

SEGMENT DATABANK

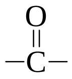
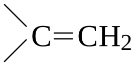
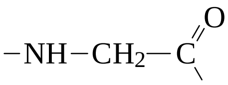
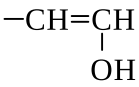
SEGMENT contains property parameters for polymer segments. The parameters stored in the databank are listed in Table A.3. A listing of the databank components is in Table A.4. Note that a special nomenclature was devised to identify polymer segments. The segment name consists of the name of the monomer from which it originates, followed by a label to identify it as a repeat unit (-R) or an end group (-E). In cases where several molecular structures are possible, a numeric subscript is used to differentiate the isomers. A similar convention is used for assigning component aliases.

Table A.3 SEGMENT Property Parameters

Parameter	No. Element	Description
ATOMNO	10	Vector of atomic number of chemical elements in segment (used with NOATOM)
CPCVK	6	Crystalline heat capacity
CPIG	11	Ideal gas heat capacity*
CPLVK	6	Liquid heat capacity
DGFVK	1	Free energy of formation, ideal gas reference state
DHCON	1	Enthalpy of condensation
DHFVK	1	Enthalpy of formation, ideal gas reference state
DHSUB	1	Enthalpy of sublimation
DNCVK	4	Crystalline density
DNGVK	5	Glass density
DNLVK	4	Liquid density
MW	1	Molecular weight
NOATOM	10	Vector of number of each type of chemical element in segment (used with ATOMNO)
TGVK	1	Glass transition temperature
TMVK	1	Melt transition temperature
VKGRP	24	Van Krevelen functional groups
VOLVW	1	Van der Waals volume
UFGRP	24	UNIFAC functional groups

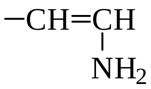
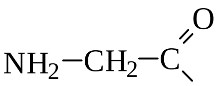
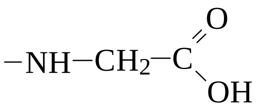
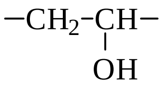
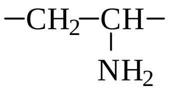
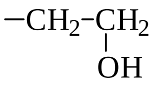
**** Estimated from Joback functional group.***

Table A.4 SEGMENT Databank Components

Alias	Segment Name	Molecular Structure
CF2-R	Methylene-fluoride-R	$-CF_2-$
CO-R	Carbonyl-R	 $-\overset{\overset{\displaystyle O}{\Vert}}{C}-$
CHF2-E	Methylene-fluoride-E	$-CHF_2$
CH2O-R	Oxymethylene-R	$-OCH_2-$
C2H2-R-1	cis-Vinylene-R	$-CH{=}CH-$
C2H2-R-2	trans-Vinylene-R	$-CH{=}CH-$
C2H2-E	Vinylidene-E	
C2H2CL-E	Vinyl-chloride-E	$-CH{=}CHCl$
C2H2F-E	Vinyl-fluoride-E	$-CH{=}CHF$
C2H2CL2-R	Vinylidene-chloride-R	$-CH_2-CCl_2-$
C2H2F2-R	Vinylidene-fluoride-R	$-CH_2-CF_2-$
C2H3-E	Vinyl-E	$-CH{=}CH_2$
C2H3CL-R	Vinyl-chloride-R	$-CH_2-CHCl-$
C2H3F-R	Vinyl-fluoride-R	$-CH_2-CHF-$
C2H3NO-R	Glycine-R	
C2H3O-E-1	Oxyvinyl-E	$-O-CH{=}CH_2$
C2H3O-E-2	Vinyl-alcohol-E	 $-CH{=}\underset{\underset{\displaystyle OH}{\vert}}{CH}$
C2H4-R	Ethylene-R	$-CH_2-CH_2-$

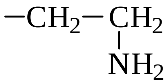
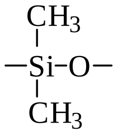
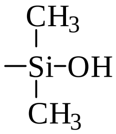
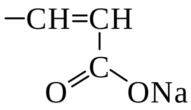
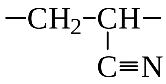
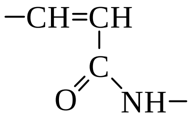
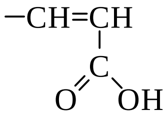
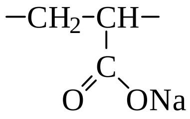
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C2H4N-E	Vinylamine-E-1	 $-CH{=}CH(NH_2)$
C2H4NO-E	Glycine-E-1	 $NH_2-CH_2-C(=O)-$
C2H4NO2-E	Glycine-E-2	 $-NH-CH_2-C(=O)OH$
C2H4O-R-1	Ethylene-oxide-R	$-CH_2-CH_2-O-$
C2H4O-R-2	Oxyethylene-R	$-O-CH_2-CH_2-$
C2H4O-R-3	Vinyl-alcohol-R	 $-CH_2-CH(OH)-$
C2H4O2-R	Ethylene-glycol-R	$-O-CH_2-CH_2-O-$
C2H5-E	Ethylene-E	$-CH_2-CH_3$
C2H5N-R	Vinylamine-R	 $-CH_2-CH(NH_2)-$
C2H5O-E-1	Ethylene-oxide-E-1	 $-CH_2-CH_2(OH)$
C2H5O-E-2	Ethylene-oxide-E-2	CH_3-CH_2-O-
C2H5O2-E	Ethylene-glycol-E	$-O-CH_2-CH_2-OH$

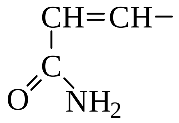
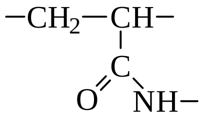
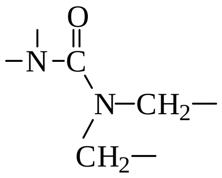
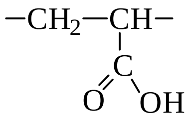
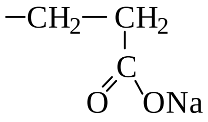
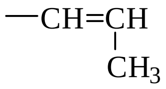
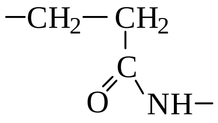
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C2H6N-E	Ethyleneamine-E	 $-CH_2-\underset{\substack{\mid\\ NH_2}}{CH_2}$
C2H6OSi-R	Dimethyl siloxane-R	 $-\overset{\substack{CH_3\\ \mid}}{\underset{\substack{\mid\\ CH_3}}{Si}}-O-$
C2H7OSi-E	Dimethyl siloxane-E	 $-\overset{\substack{CH_3\\ \mid}}{\underset{\substack{\mid\\ CH_3}}{Si}}-OH$
C3H2O2Na-E	Sodium acrylate-E-1	
C3H3N-R	Acrylonitrile-R	 $-CH_2-\underset{\substack{\mid\\ C\equiv N}}{CH}-$
C3H3NO-R	Acrylamide-R-1	
C3H3O2-E	Acrylic acid-E-1	
C3H3O2Na-R	Sodium-acrylate-R	

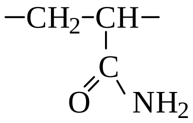
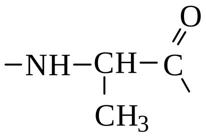
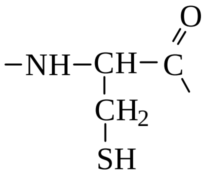
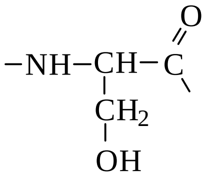
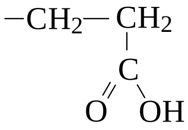
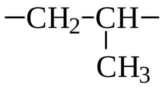
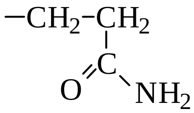
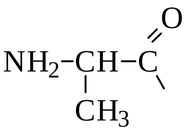
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C3H4NO-E	Acrylamide-E-1	
C3H4NO-B	Acrylamide-B	
C3H4N2O-B	Urea-formaldehyde-R	
C3H4O2-R	Acrylic-acid-R	
C3H4O2Na-E	Sodium-acrylate-E-2	
C3H5-E	Propylene-E-1	 $-CH{=}\underset{\displaystyle CH_3}{\underset{\vert}{CH}}$
C3H5Cl-R	2-chloropropylene-R	$-CH_2-CHCl-CH_2-$
C3H5NO-R-1	Acrylamide-R-2	

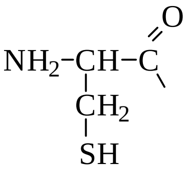
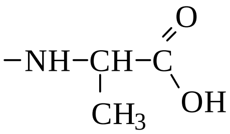
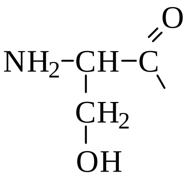
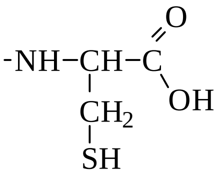
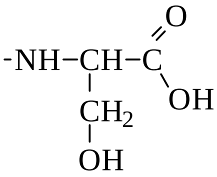
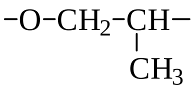
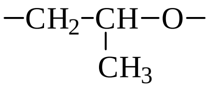
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C3H5NO-R-2	Acrylamide-R-3	
C3H5NO-R-3	Alanine-R	
C3H5NOS-R	Cysteine-R	
C3H5NO2-R	Serine-R	
C3H5O2-E	Acrylic-acid-E-2	
C3H6-R	Propylene-R	 $-CH_2-\underset{CH_3}{\underset{\mid}{CH}}-$
C3H6NO-E-1	Acrylamide-E-2	
C3H6NO-E-2	Alanine-E-1	

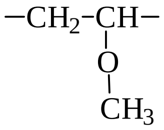
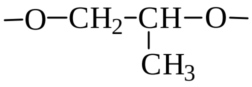
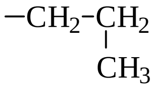
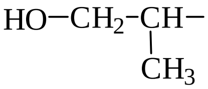
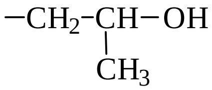
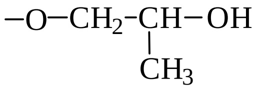
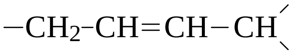
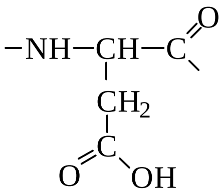
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C3H6NOS-E	Cysteine-E-1	 $\mathrm{NH_2{-}\underset{\substack{\vert\\CH_2\\\vert\\SH}}{CH}{-}C({=}O)\backslash}$
C3H6NO2-E-1	Alanine-E-2	 $\mathrm{{-}NH{-}\underset{\substack{\vert\\CH_3}}{CH}{-}C({=}O){-}OH}$
C3H6NO2-E-2	Serine-E-1	 $\mathrm{NH_2{-}\underset{\substack{\vert\\CH_2\\\vert\\OH}}{CH}{-}C({=}O)\backslash}$
C3H6NO2S-E	Cysteine-E-2	 $\mathrm{{-}NH{-}\underset{\substack{\vert\\CH_2\\\vert\\SH}}{CH}{-}C({=}O){-}OH}$
C3H6NO3-E	Serine-E-2	 $\mathrm{{-}NH{-}\underset{\substack{\vert\\CH_2\\\vert\\OH}}{CH}{-}C({=}O){-}OH}$
C3H6O-R-1	Oxypropylene-R	 $\mathrm{{-}O{-}CH_2{-}\underset{\substack{\vert\\CH_3}}{CH}{-}}$
C3H6O-R-2	Propylene-oxide-R	 $\mathrm{{-}CH_2{-}\underset{\substack{\vert\\CH_3}}{CH}{-}O{-}}$

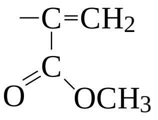
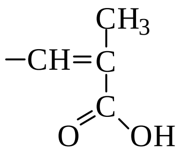
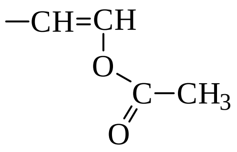
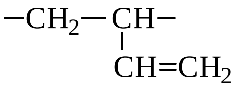
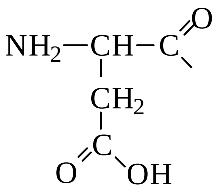
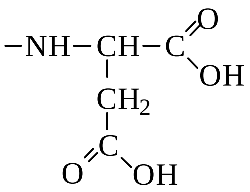
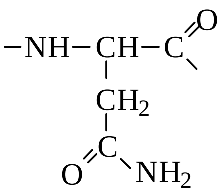
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C3H6O-R-3	Vinyl-methyl-ether-R	 $-CH_2-\underset{\substack{\vert \\ O \\ \vert \\ CH_3}}{CH}-$
C3H6O2-R	Propylene-glycol-R	 $-O-CH_2-\underset{\substack{\vert \\ CH_3}}{CH}-O-$
C3H7-E	Propylene-E-2	 $-CH_2-\underset{\substack{\vert \\ CH_3}}{CH_2}$
C3H7O-E-1	Oxypropylene-E	 $HO-CH_2-\underset{\substack{\vert \\ CH_3}}{CH}-$
C3H7O-E-2	Propylene-oxide-E	 $-CH_2-\underset{\substack{\vert \\ CH_3}}{CH}-OH$
C3H7O2-E	Propylene-glycol-E	 $-O-CH_2-\underset{\substack{\vert \\ CH_3}}{CH}-OH$
C4H5-B	Butadiene-B	 $-CH_2-CH=CH-CH<$
C4H5-E-1	Butadiene-E-1	$-CH=CH-CH=CH_2$
C4H5-E-2	Butadiene-E-2	$-CH_2-CH=C=CH_2$
C4H5NO3-R	Aspartic-acid-R	

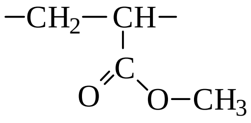
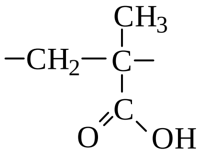
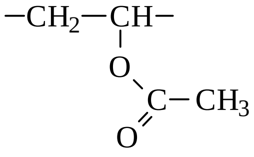
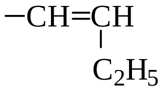
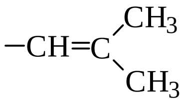
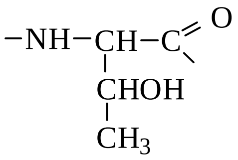
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C4H5O2-E-1	Methyl-acrylate-E-1	
C4H5O2-E-2	Methyl-acrylic-acid-E-1	
C4H5O2-E-3	Vinyl-acetate-E-1	
C4H6-R-1	Butadiene-R-1	$-CH_2-CH=CH-CH_2-$
C4H6-R-2	Butadiene-R-2	 $-CH_2-\underset{\displaystyle CH=CH_2}{\underset{\vert}{CH}}-$
C4H6NO3-E	Aspartic-acid-E-1	
C4H6NO4-E	Aspartic-acid-E-2	
C4H6N2O2-R	Asparagine-R	

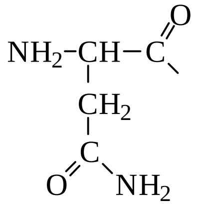
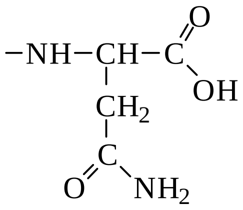
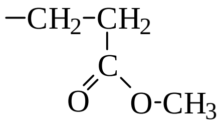
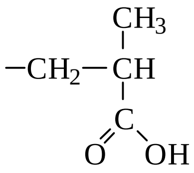
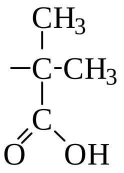
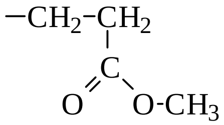
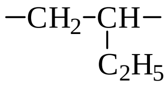
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C4H6O2-R-1	Methyl-acrylate-R	
C4H6O2-R-2	Methyl acrylic-acid-R	
C4H6O2-R-3	Vinyl-acetate-R	
C4H7-E-1	1-Butene-E	 $\begin{array}{l} -CH{=}CH \\ \quad\;\; \mid \\ \quad\;\; C_2H_5 \end{array}$
C4H7-E-2	Isobutylene-E	
C4H7-E-3	Butadiene-E-3	$-CH_2-CH_2-CH=CH_2$
C4H7-E-4	Butadiene-E-4	$-CH_2-CH=CH-CH_3$
C4H7NO2-R	Threonine-R	

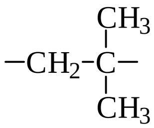
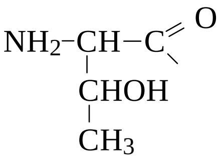
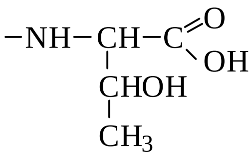
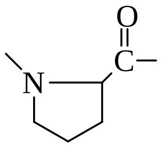
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C4H7N2O2-E	Asparagine-E-1	
C4H7N2O3-E	Asparagine-E-2	
C4H7O2-E-1	Methyl-acrylate-E-2	
C4H7O2-E-2	Methyl-acrylic-acid-E-2	
C4H7O2-E-3	Methyl-acrylic-acid-E-3	
C4H7O2-E-4	Vinyl-acetate-E-2	
C4H8-R-1	1-Butene-R	 $-CH_2-\underset{C_2H_5}{\underset{\mid}{CH}}-$

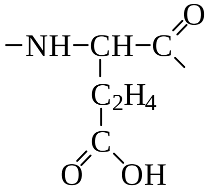
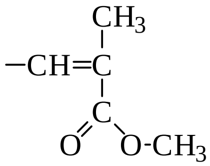
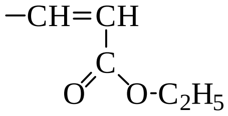
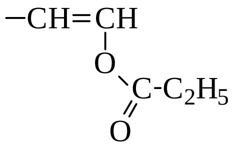
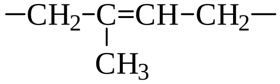
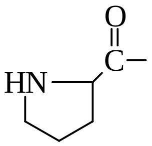
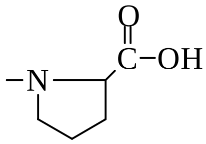
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C4H8-R-2	Isobutylene-R	 $-CH_2-\overset{CH_3}{\underset{CH_3}{C}}-$
C4H8NO2-E	Threonine-E-1	
C4H8NO3-E	Threonine-E-2	
C4H8O-R	Tetrahydrofuran-R	$-CH_2-CH_2-CH_2-CH_2-O-$
C4H8O2-R	Butylene-glycol-R	$-O-C_4H_8-O-$
C4H8O3-R	Diethylene-glycol-R	$-O-C_2H_4-O-C_2H_4-O-$
C4H9O-E-1	Tetrahydrofuran-E-1	$-C_4H_8-OH$
C4H9O-E-2	Tetrahydrofuran-E-2	C_4H_9-O-
C4H9O2-E	Butylene-glycol-E	$-O-C_4H_8-OH$
C4H9O3-E	Diethylene-glycol-E	$-O-C_2H_4-O-C_2H_4-OH$
C5H7NO-R	Proline-R	

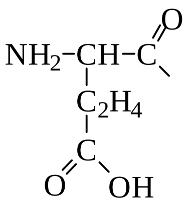
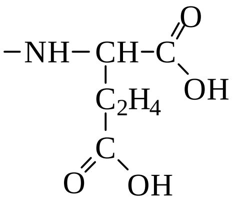
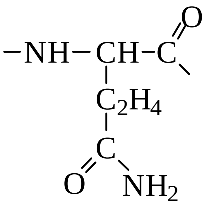
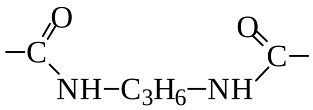
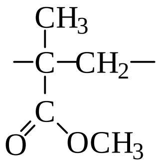
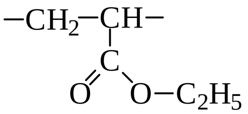
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C5H7NO3-R	Glutamic-acid-R	
C5H7O2-E-1	Methyl-methacrylate-E-1	
C5H7O2-E-2	Ethyl-acrylate-E-1	
C5H7O2-E-3	Vinyl-propionate-E-1	
C5H8-R	Isoprene-R	 $-CH_2-\underset{\displaystyle CH_3}{\underset{\vert}{C}}=CH-CH_2-$
C5H8NO-E	Proline-E-1	
C5H8NO2-E	Proline-E-2	

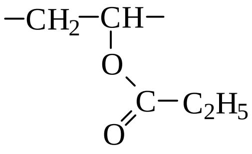
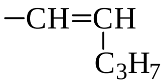
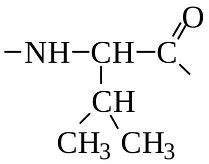
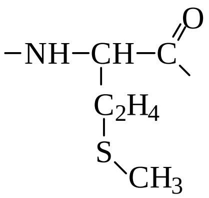
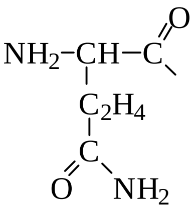
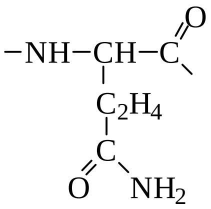
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C5H8NO3-E	Glutamic-acid-E-1	
C5H8NO4-E	Glutamic-acid-E-2	
C5H8N2O2-R-1	Glutamine-R	
C5H8N2O2-R-2	Trimethylene-diisocyanate-R	
C5H8O2-R-1	Methyl-methacrylate-R	
C5H8O2-R-2	Ethyl-acrylate-R	

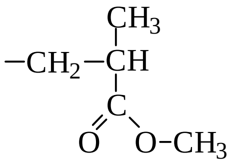
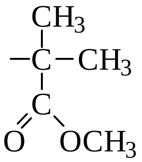
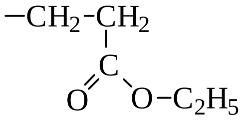
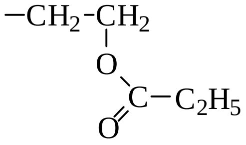
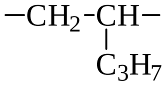
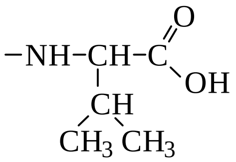
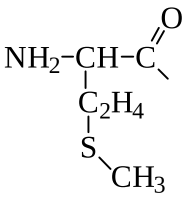
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C5H8O2-R-3	Vinyl-propionate-R	
C5H9-E	1-Pentene-E-1	 $-CH=\underset{\overset{\vert}{C_3H_7}}{CH}$
C5H9NO-R	Valine-R	
C5H9NOS-R	Methionine-R	
C5H9N2O2-E	Glutamine-E-1	
C5H9N2O3-E	Glutamine-E-2	

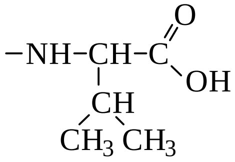
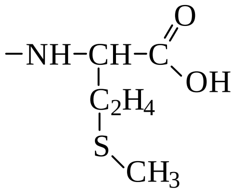
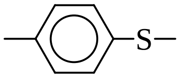
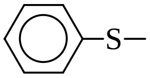
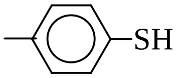
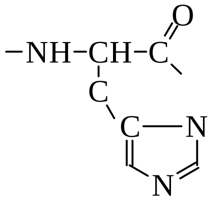
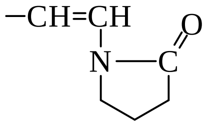
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C5H9O2-E-1	Methyl-methacrylate-E-2	
C5H9O2-E-2	Methyl-methacrylate-E-3	
C5H9O2-E-3	Ethyl-acrylate-E-2	
C5H9O2-E-4	Vinyl-propionate-E-2	
C5H10-R	1-Pentene-R	
C5H10NO-E	Valine-E-1	
C5H10NOS-E	Methionine-E-1	

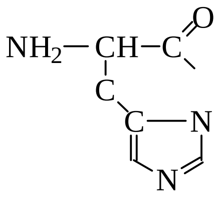
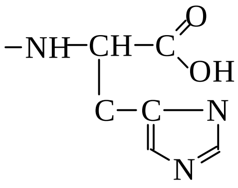
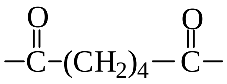
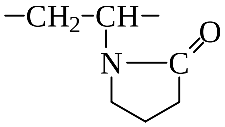
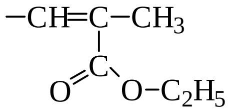
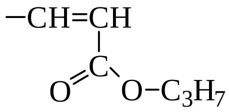
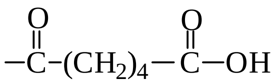
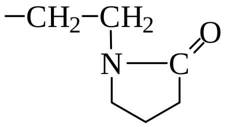
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C5H10NO2-E	Valine-E-2	
C5H10NO2S-E	Methionine-E-2	
C6H4S-R	Phenylene-sulfide-R	
C6H5S-E-1	Phenylene-sulfide-E-1	
C6H5S-E-2	Phenylene-sulfide-E-2	
C6H7N3O-R	Histidine-R	
C6H8NO-E	Vinylpyrrolidnone-E-1	

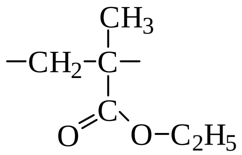
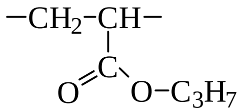
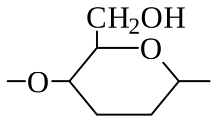
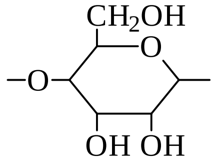
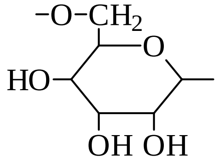
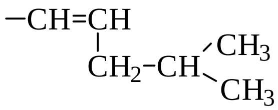
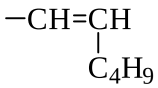
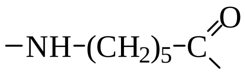
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H8N3O-E	Histidine-E-1	
C6H8N3O2-E	Histidine-E-2	
C6H8O2-R	Adipic-acid-R	 $-\overset{\overset{O}{\|}}{C}-(CH_2)_4-\overset{\overset{O}{\|}}{C}-$
C6H9NO-R	Vinylpyrrolidnone-R	
C6H9O2-E-1	Ethyl-methacrylate-E-3	
C6H9O2-E-2	n-Propyl-acrylate-E-1	
C6H9O3-E	Adipic-acid-E	 $-\overset{\overset{O}{\|}}{C}-(CH_2)_4-\overset{\overset{O}{\|}}{C}-OH$
C6H10NO-E	Vinylpyrrolidnone-E-3	

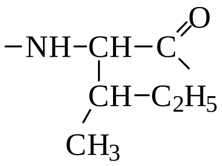
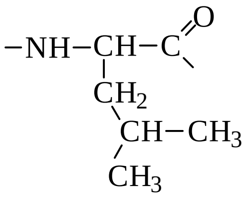
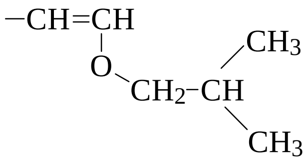
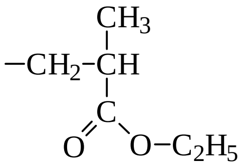
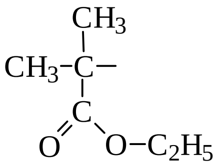
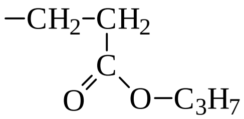
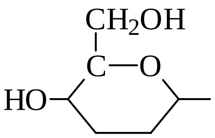
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H10O2-R-1	Ethyl-methacrylate-R-1	
C6H10O2-R-2	n-Propyl-acrylate-R	
C6H10O3-R	Amylose-R	
C6H10O5-R-1	Cellulose-R	
C6H10O5-R-2	Dextran-R	
C6H11-E-1	4-Methyl-1-pentene-E-1	
C6H11-E-2	1-Hexane-E-1	
C6H11NO-R-1	Caprolactam-R	 $-NH-(CH_2)_5-C(=O)-$

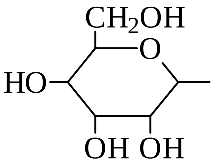
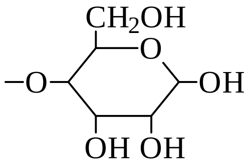
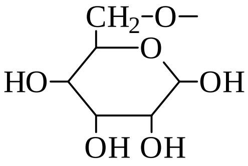
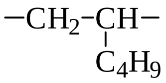
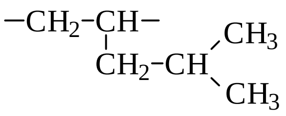
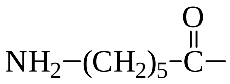
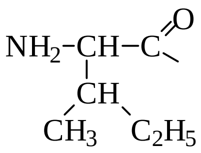
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H11NO-R-2	Isoleucine-R	
C6H11NO-R-3	Leucine-R	
C6H11O-E	Vinyl-isobutyl-ether-E-1	
C6H11O2-E-1	Ethyl-methacrylate-E-1	
C6H11O2-E-2	Ethyl-methacrylate-E-2	
C6H11O2-E-3	n-Propyl-acrylate-E-2	
C6H11O3-E	Amylose-E	

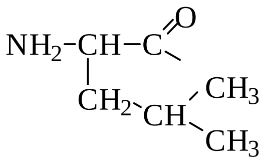
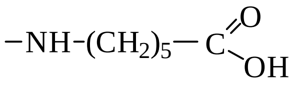
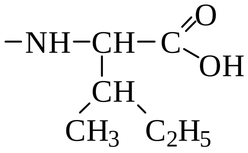
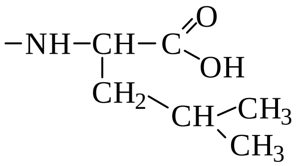
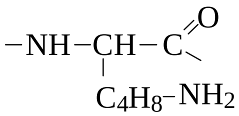
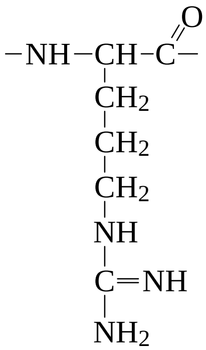
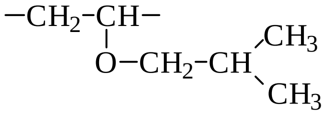
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H11O5-E	Cellulose-E-1	
C6H11O6-E-1	Cellulose-E-2	
C6H11O6-E-2	Dextran-E-2	
C6H12-R-1	1-Hexane-R	 $-CH_2-\underset{C_4H_9}{\underset{\vert}{CH}}-$
C6H12-R-2	4-Methyl-1-pentene-R	 $-CH_2-\underset{CH_2-CH(CH_3)_2}{\underset{\vert}{CH}}-$
C6H12NO-E-1	Caprolactam-E-1	 $NH_2-(CH_2)_5-\overset{O}{\overset{\Vert}{C}}-$
C6H12NO-E-2	Isoleucine-E-1	 $NH_2-\underset{CH(CH_3)(C_2H_5)}{\underset{\vert}{CH}}-C(=O)-$

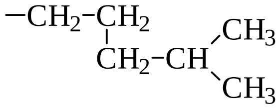
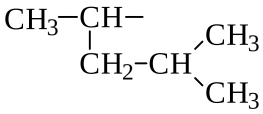
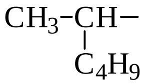
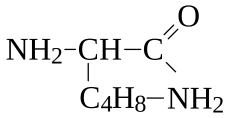
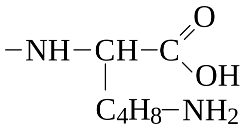
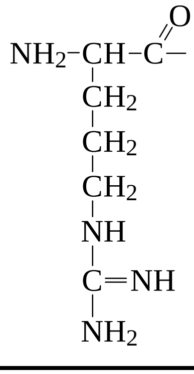
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H12NO-E-3	Leucine-E-1	
C6H12NO2-E-1	Caprolactam-E-2	 $-NH-(CH_2)_5-C(=O)OH$
C6H12NO2-E-2	Isoleucine-E-2	
C6H12NO2-E-3	Leucine-E-2	
C6H12N2O-R	Lysine-R	 $-NH-CH(C_4H_8-NH_2)-C(=O)-$
C6H12N4O-R	Arginine-R	
C6H12O-R	Vinyl-isobutyl-ether-R	

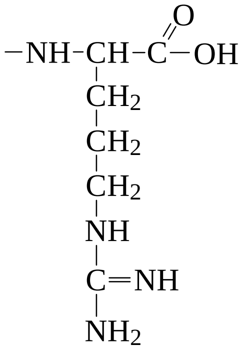
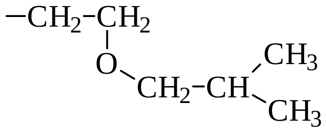
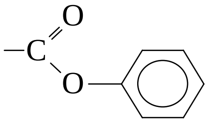
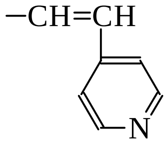
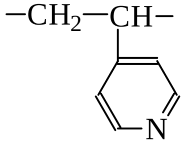
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H12O2-R	Hexamethylene-diol-R	$-O-(CH_2)_6-O-$
C6H13-E-1	4-Methyl-1-pentene-E-2	
C6H13-E-2	4-Methyl-1-pentene-E-3	
C6H13-E-3	1-Hexane-E-2	 $CH_3-\underset{\displaystyle C_4H_9}{\underset{\vert}{CH}}-$
C6H13N2O-E	Lysine-E-1	
C6H13N2O2-E	Lysine-E-2	
C6H13N4O-E	Arginine-E-1	

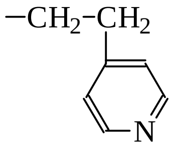
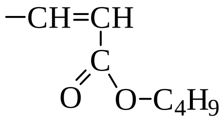
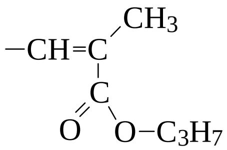
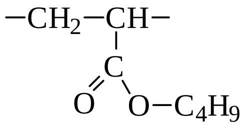
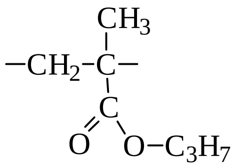
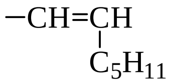
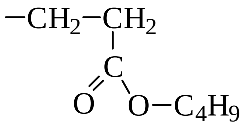
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C6H13N4O2-E	Arginine-E-2	
C6H13O-E	Vinyl-isobutyl-ether-E-2	
C6H13O2-E	Hexamethylene-diol-E	$-O-(CH_2)_6-OH$
C6H14N2-R	Hexamethylene-diamine-R	$-NH-(CH_2)_6-NH-$
C6H15N2-E	Hexamethylene-diamine-E	$-NH-(CH_2)_6-NH_2$
C7H5O2-E	Phenylcarbonate-E	
C7H6N-E	4-Vinyl-pyridine-E-1	
C7H7N-R	4-Vinyl-pyridine-R	

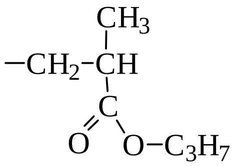
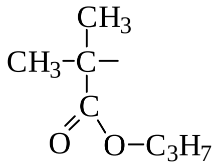
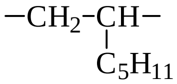
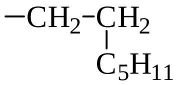
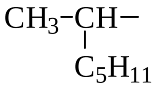
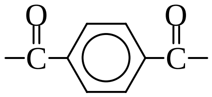
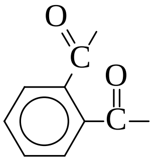
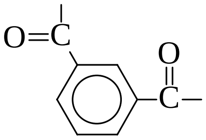
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C7H8N-E	4-Vinyl-pyridine-E-2	
C7H11O2-E-1	n-Butyl-acrylate-E-1	
C7H11O2-E-2	n-Propyl-methacrylate-E-1	
C7H12O2-R-1	n-Butyl-acrylate-R	
C7H12O2-R-2	n-Propyl-methacrylate-R	
C7H13-E	1-Heptene-E-1	 $-CH{=}\underset{\displaystyle C_5H_{11}}{\underset{\vert}{CH}}$
C7H13O2-E-1	n-Butyl-acrylate-E-2	

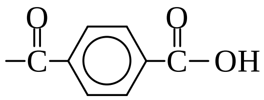
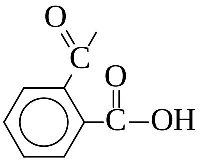
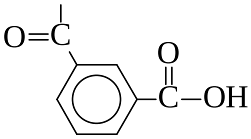
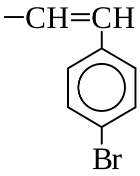
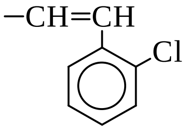
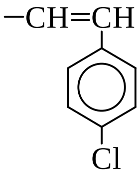
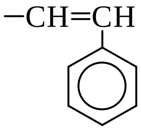
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C7H13O2-E-2	n-Propyl-methacrylate-E-2	
C7H13O2-E-3	n-Propyl-methacrylate-E-3	
C7H14-R	1-Heptene-R	 $-CH_2-\underset{C_5H_{11}}{\underset{\vert}{CH}}-$
C7H15-E-1	1-Heptene-E-2	 $-CH_2-\underset{C_5H_{11}}{\underset{\vert}{CH_2}}$
C7H15-E-2	1-Heptene-E-3	 $CH_3-\underset{C_5H_{11}}{\underset{\vert}{CH}}-$
C8H4O2-R	Terephthalate-R	
C8H4O2-R-1	Phthalate-R	
C8H4O2-R-2	Isophthalate-R	

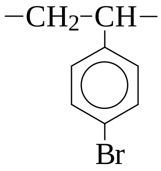
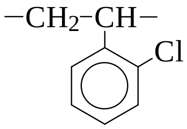
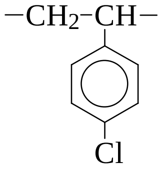
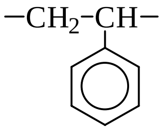
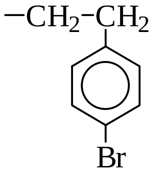
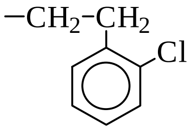
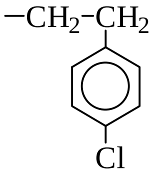
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H5O3-E	Terephthalic-acid-E	
C8H5O3-E-1	Phthalic-acid-E	
C8H5O3-E-2	Isophthalic acid-E	
C8H6Br-E	p-Bromostyrene-E-1	
C8H6Cl-E-1	o-Chlorostyrene-E-1	
C8H6Cl-E-2	p-Chlorostyrene-E-1	
C8H7-E	Styrene-E-1	

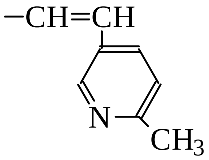
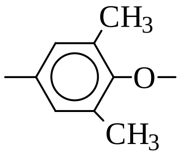
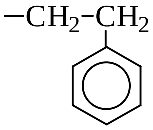
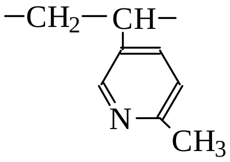
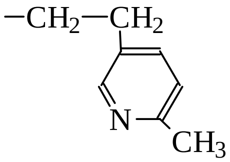
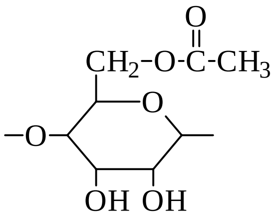
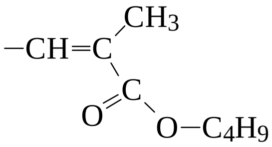
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H7Br-R	p-Bromostyrene-R	
C8H7Cl-R-1	o-Chlorostyrene-R	
C8H7Cl-R-2	p-Chlorostyrene-R	
C8H8-R	Styrene-R	
C8H8Br-E	p-Bromostyrene-E-2	
C8H8Cl-E-1	o-Chlorostyrene-E-2	
C8H8Cl-E-2	p-Chlorostyrene-E-2	

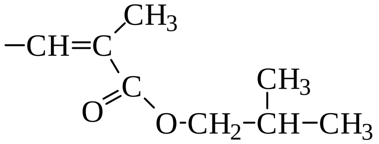
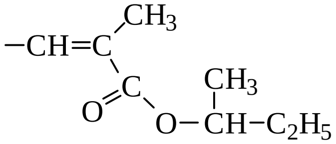
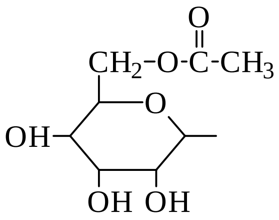
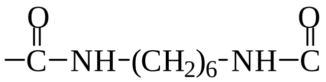
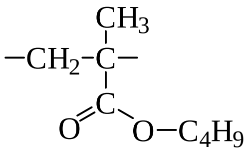
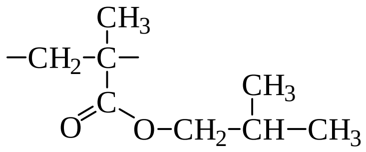
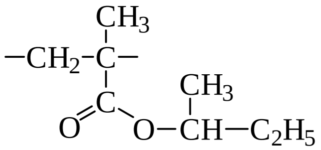
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H8N-E	2-Methyl-5-vinylpyridine-E-1	
C8H8O-R	Phenylene-oxide-R	
C8H9-E	Styrene-E-2	
C8H9N-R	2-Methyl-5-vinylpyridine-R	
C8H10N-E	2-Methyl-5-vinylpyridine-E-2	
C8H12O6-R	Cellulose-acetate-R	
C8H13O2-E-1	n-Butyl-methacrylate-E-1	

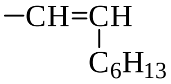
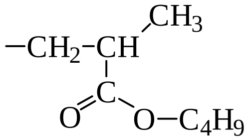
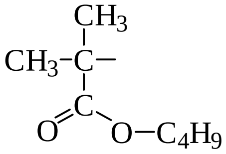
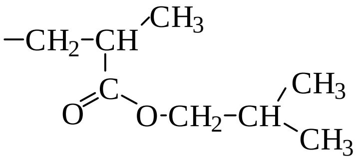
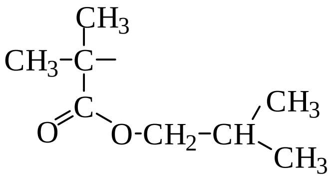
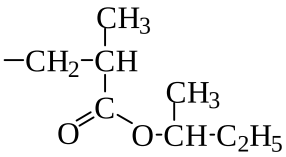
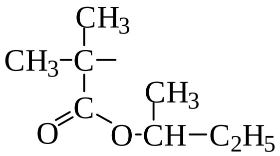
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H13O2-E-2	Isobutyl-methacrylate-E-1	
C8H13O2-E-3	sec-Butyl-methacrylate-E-1	
C8H13O6-E	Cellulose-acetate-E	
C8H14N2O2-R	Hexamethylene-diisocyanate-R	 $-\overset{\overset{\displaystyle O}{\|}}{C}-NH-(CH_2)_6-NH-\overset{\overset{\displaystyle O}{\|}}{C}$
C8H14O2-R-1	n-Butyl-methacrylate-R	
C8H14O2-R-2	Isobutyl-methacrylate-R	
C8H14O2-R-3	sec-Butyl-methacrylate-R	

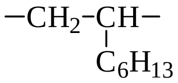
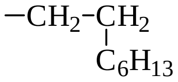
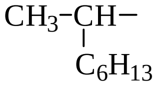
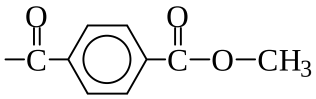
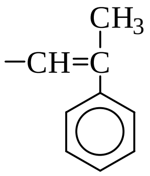
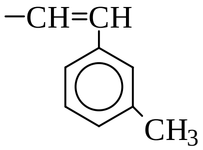
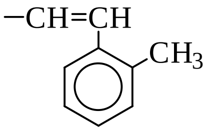
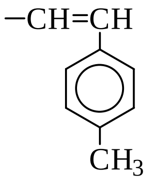
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H15-E	1-Octene-E-1	 $-CH{=}\underset{C_6H_{13}}{\underset{\mid}{CH}}$
C8H15O2-E-1	n-Butyl-methacrylate-E-2	
C8H15O2-E-2	n-Butyl-methacrylate-E-3	
C8H15O2-E-3	Isobutyl-methacrylate-E-2	
C8H15O2-E-4	Isobutyl-methacrylate-E-3	
C8H15O2-E-5	sec-Butyl-methacrylate-E-2	
C8H15O2-E-6	sec-Butyl-methacrylate-E-3	

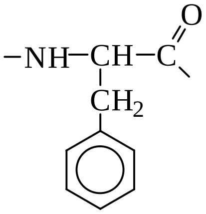
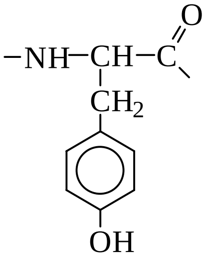
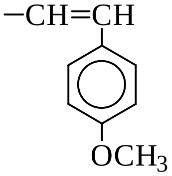
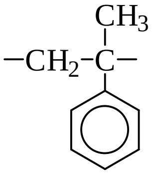
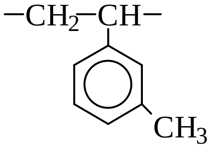
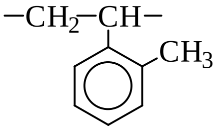
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C8H16-R	1-Octene-R	 $-CH_2-\underset{\substack{\mid\\ C_6H_{13}}}{CH}-$
C8H17-E-1	1-Octene-E-2	 $-CH_2-\underset{\substack{\mid\\ C_6H_{13}}}{CH_2}$
C8H17-E-2	1-Octene-E-3	 $CH_3-\underset{\substack{\mid\\ C_6H_{13}}}{CH}-$
C9H7O3-E	dimethyl-terephthalate-E	
C9H9-E-1	alpha-Methylstyrene-E-1	
C9H9-E-2	m-Methylstyrene-E-1	
C9H9-E-3	o-Methylstyrene-E-1	
C9H9-E-4	p-Methylstyrene-E-1	

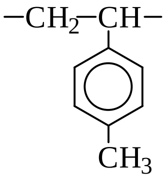
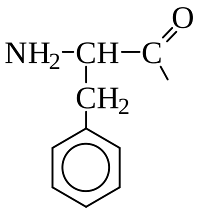
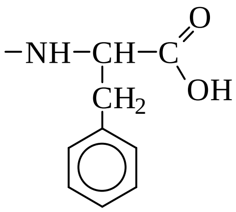
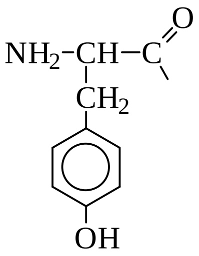
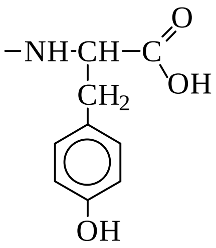
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C9H9NO-R	Phenylalanine-R	
C9H9NO2-R	Tyrosine-R	
C9H9O-E	p-Methoxystyrene-E-1	
C9H10-R-1	alpha-Methylstyrene-R	
C9H10-R-2	m-Methylstyrene-R	
C9H10-R-3	o-Methylstyrene-R	

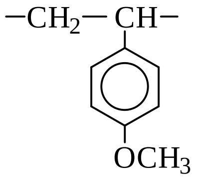
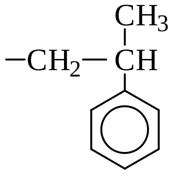
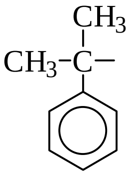
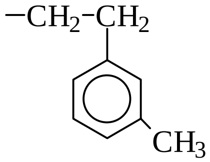
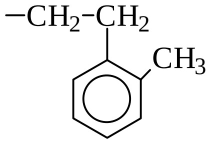
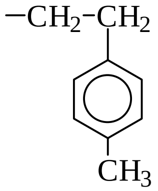
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C9H10-R-4	p-Methylstyrene-R	
C9H10NO-E	Phenylalanine-E-1	
C9H10NO2-E-1	Phenylalanine-E-2	
C9H10NO2-E-2	Tyrosine-E-1	
C9H10NO3-E	Tyrosine-E-2	

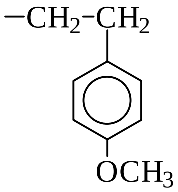
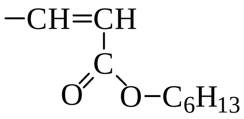
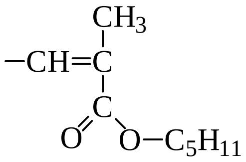
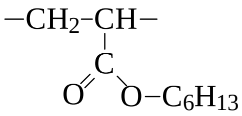
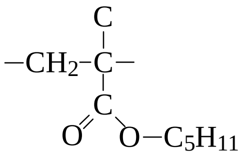
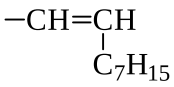
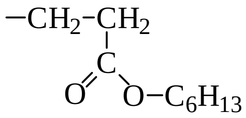
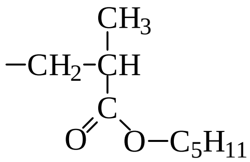
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C9H10O-R	p-Methoxystyrene-R	
C9H11-E-1	alpha-Methylstyrene-E-2	
C9H11-E-2	alpha-Methylstyrene-E-3	
C9H11-E-3	m-Methylstyrene-E-2	
C9H11-E-4	o-Methylstyrene-E-2	
C9H11-E-5	p-Methylstyrene-E-2	

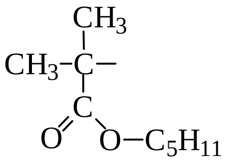
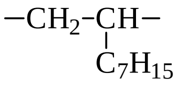
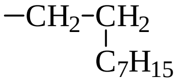
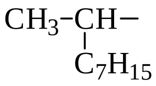
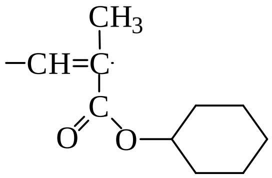
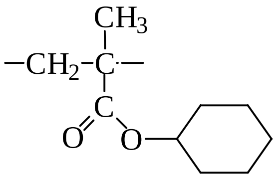
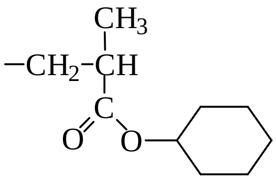
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C9H11O-E	p-Methoxystyrene-E-2	
C9H15O2-E-1	n-Hexyl-acrylate-E-1	
C9H15O2-E-2	n-Pentyl-methacrylate-E-1	
C9H16O2-R-1	n-Hexyl-acrylate-R	
C9H16O2-R-2	n-Pentyl-methacrylate-R	
C9H17-E	1-Nonene-E-1	 $-CH{=}\underset{\underset{C_7H_{15}}{\vert}}{CH}$
C9H17O2-E-1	n-Hexyl-acrylate-E-2	
C9H17O2-E-2	n-Pentyl-methacrylate-E-2	

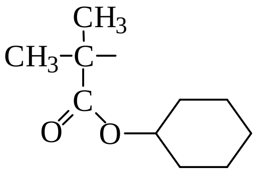
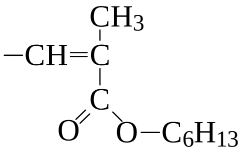
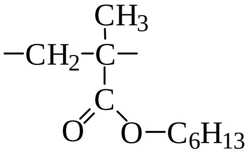
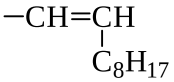
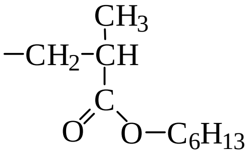
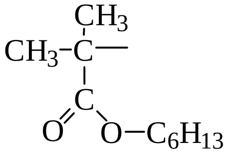
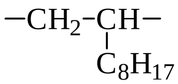
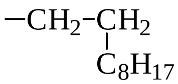
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C9H17O2-E-3	n-Pentyl-methacrylate-E-3	
C9H18-R	1-Nonene-R	 $-CH_2-\underset{C_7H_{15}}{\underset{\mid}{CH}}-$
C9H19-E-1	1-Nonene-E-2	 $-CH_2-\underset{C_7H_{15}}{\underset{\mid}{CH_2}}$
C9H19-E-2	1-Nonene-E-3	 $CH_3-\underset{C_7H_{15}}{\underset{\mid}{CH}}-$
C10H15O2-E	Cyclohexyl-methacrylate-E-1	
C10H16O2-R	Cyclohexyl-methacrylate-R	
C10H17O2-E-1	Cyclohexyl-methacrylate-E-2	

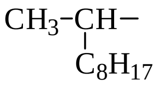
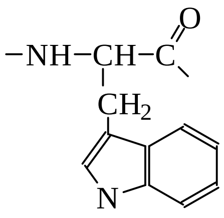
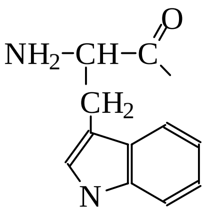
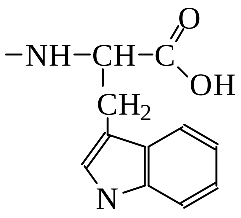
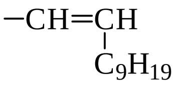
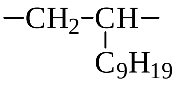
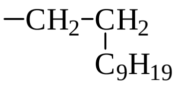
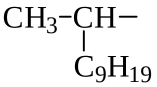
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C10H17O2-E-2	Cyclohexyl-methacrylate-E-3	
C10H17O2-E-3	n-Hexyl-methacrylate-E-1	
C10H18O2-R	n-Hexyl-methacrylate-R	
C10H19-E	1-Decene-E-1	 $-CH{=}\underset{C_8H_{17}}{CH}$
C10H19O2-E-1	n-Hexyl-methacrylate-E-2	
C10H19O2-E-2	n-Hexyl-methacrylate-E-3	
C10H20-R	1-Decene-R	 $-CH_2-\underset{C_8H_{17}}{CH}-$
C10H21-E-1	1-Decene-E-2	 $-CH_2-\underset{C_8H_{17}}{CH_2}$

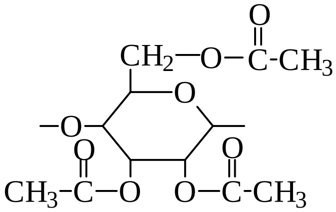
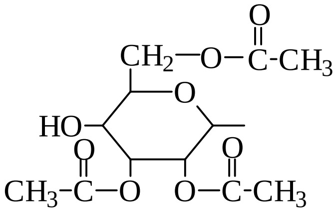
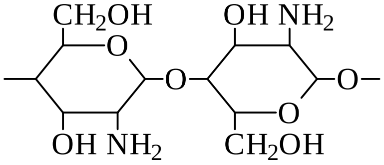
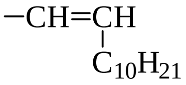
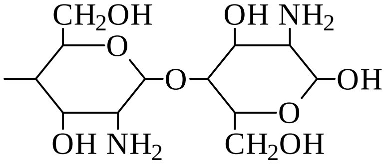
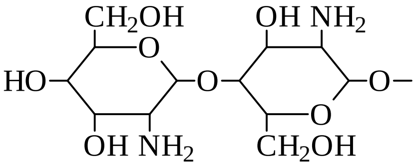
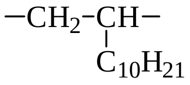
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C10H21-E-2	1-Decene-E-3	 $CH_3-\underset{C_8H_{17}}{CH}-$
C11H10N2O-R	Tryptophan-R	
C11H11N2O-E	Tryptophan-E-1	
C11H11N2O2-E	Tryptophan-E-2	
C11H21-E	1-Undecene-E-1	 $-CH=\underset{C_9H_{19}}{CH}$
C11H22-R	1-Undecene-R	 $-CH_2-\underset{C_9H_{19}}{CH}-$
C11H23-E-1	1-Undecene-E-2	 $-CH_2-\underset{C_9H_{19}}{CH_2}$
C11H23-E-2	1-Undecene-E-3	 $CH_3-\underset{C_9H_{19}}{CH}-$

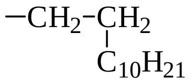
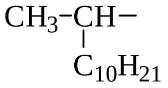
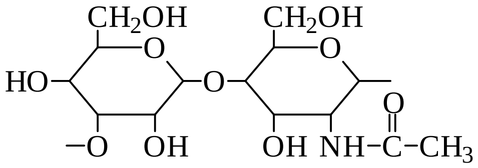
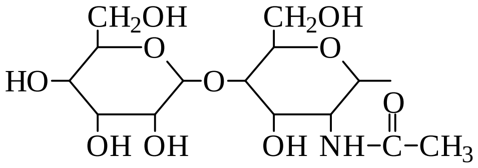
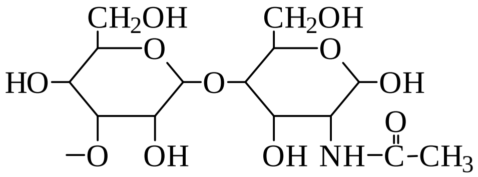
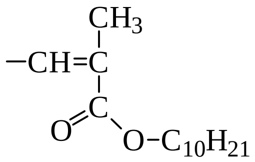
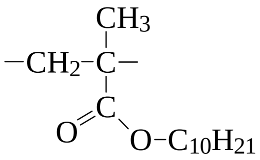
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C12H16O8-R	Cellulose-triacetate-R	
C12H17O8-E	Cellulose-triacetate-E	
C12H22N2O8-R	Chitosan-R	
C12H23-E	1-Dodecene-E-1	 $-CH=\underset{\underset{C_{10}H_{21}}{\vert}}{CH}$
C12H23N2O8-E	Chitosan-E-1	
C12H23N2O9-E	Chitosan-E-2	
C12H24-R	1-Dodecene-R	 $-CH_2-\underset{\underset{C_{10}H_{21}}{\vert}}{CH}-$

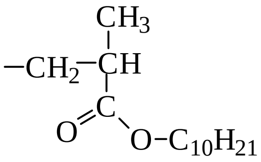
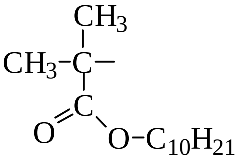
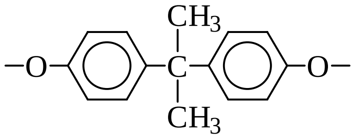
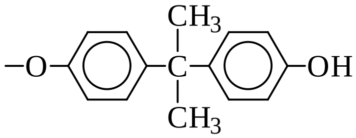
continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C12H25-E-1	1-Dodecene-E-2	 $-CH_2-\underset{C_{10}H_{21}}{\underset{\vert}{CH_2}}$
C12H25-E-2	1-Dodecene-E-3	 $CH_3-\underset{C_{10}H_{21}}{\underset{\vert}{CH}}-$
C14H23NO10-R	Heparin-R	
C14H24NO10-E	Heparin-E-1	
C14H24NO11-E	Heparin-E-2	
C14H25O2-E	Decyl-methacrylate-E-1	
C14H26O2-R	Decyl-methacrylate-R	

continued

Table A.4 SEGMENT Databank Components (cont.)

Alias	Segment Name	Molecular Structure
C14H27O2-E-1	Decyl-methacrylate-E-2	
C14H27O2-E-2	Decylmethacrylate-E-3	
C15H14O2-R	Bisphenol-A-R	
C15H15O2-E	Bisphenol-A-E	

B PHYSICAL PROPERTY METHODS

This appendix documents the physical property route structure for the polymer specific property methods:

For each property method the property models used in the route calculations are described.

Table B.1 POLYFH: Flory-Huggins Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMX01	ESRK	Redlich-Kwong
HVMX	HVMX01	ESRK	Redlich-Kwong
GVMX	GVMX01	ESRK	Redlich-Kwong
SVMX	SVMX01	ESRK	Redlich-Kwong
VVMX	VVMX01	ESRK	Redlich-Kwong
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIV01	ESRK0	Redlich-Kwong
HV	HV02	ESRK0	Redlich-Kwong
GV	GV01	ESRK0	Redlich-Kwong
SV	SV01	ESRK0	Redlich-Kwong
VV	VV01	ERSK0	Redlich-Kwong
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KVOSTLTH	Stiel-Thodos

continued

Table B.1 POLYFH: Flory-Huggins Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMXFH	GMFH, WHENRY, HENRY, PL0XANT, ESRK0, VL0RKT, VL1BROC	Flory-Huggins, HENRY, Extended Antoine, Redlich-Kwong, Rackett, Brevi-O'Connell
HLMX	HLMXFH	GMFH, HL0DVK*	Flory-Huggins, van Krevelen
GLMX	GLMXFH	GMFH, GL0DVK	Flory-Huggins, van Krevelen
SLMX	SLMXFH	GMFH, HL0DVK,* GL0DVK	Flory-Huggins, van Krevelen
VLMX	VLMXDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHIL04	PL0XANT, ESRK0, VL0RKT	Extended Antoine, Redlich-Kwong, Rackett
HL	HLDVK	HL0DVK*	van Krevelen, Aspen
GL	GLDVK	GL0DVK	van Krevelen
SL	SLDVK	HL0DVK,* GL0DVK	van Krevelen
VL	VLDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
DL	DL01	DL0WCA	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal mixing
VSMX	VSMXDVK	VS0DVK,VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

**** Optional van Krevelen/DIPPR model, HL0DVKD, available.***

Table B.2 POLYNRTL: Polymer Non-Random Two-Liquid Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMX01	ESRK	Redlich-Kwong
HVMX	HVMX01	ESRK	Redlich-Kwong
GVMX	GVMX01	ESRK	Redlich-Kwong
SVMX	SVMX01	ESRK	Redlich-Kwong
VVMX	VVMX01	ESRK	Redlich-Kwong
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIV01	ESRK0	Redlich-Kwong
HV	HV02	ESRK0	Redlich-Kwong
GV	GV01	ESRK0	Redlich-Kwong
SV	SV01	ESRK0	Redlich-Kwong
VV	VV01	ERSK0	Redlich-Kwong
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KVOSTLTH	Stiel-Thodos

continued

Table B.2 POLYNRTL: Polymer Non-Random Two-Liquid Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMXP	GMNRTLP,** WHENRY, HENRY, PL0XANT, ESRK0, VL0RKT, VL1BROC	Polymer NRTL, HENRY, Extended Antoine, Redlich-Kwong, Rackett, Brevi-O'Connell
HLMX	HLMXP	GMNRTLP,** HL0DVK*	Polymer NRTL, van Krevelen
GLMX	GLMXP	GMNRTLP,** GL0DVK	Polymer NRTL, van Krevelen
SLMX	SLMXP	GMNRTLP,** HL0DVK,* GL0DVK	Polymer NRTL, van Krevelen
VLMX	VLMXDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrades
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHIL04	PL0XANT, ESRK0, VL0RKT	Extended Antoine, Redlich-Kwong, Rackett
HL	HLDVK	HL0DVK*	van Krevelen, Aspen
GL	GLDVK	GL0DVK	van Krevelen
SL	SLDVK	HL0DVK,* GL0DVK	van Krevelen
VL	VLDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
DL	DL01	DL0WCA	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal mixing
VSMX	VSMXDVK	VS0DVK,VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

**** Optional van Krevelen/DIPPR model, HL0DVKD, available.***

***** Optional Polymer NRTL model with asymmetric alpha, GMRTLPA, available.***

Table B.3 POLYUF: Polymer UNIFAC Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMX01	ESRK	Redlich-Kwong
HVMX	HVMX01	ESRK	Redlich-Kwong
GVMX	GVMX01	ESRK	Redlich-Kwong
SVMX	SVMX01	ESRK	Redlich-Kwong
VVMX	VVMX01	ESRK	Redlich-Kwong
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIV01	ESRK0	Redlich-Kwong
HV	HV02	ESRK0	Redlich-Kwong
GV	GV01	ESRK0	Redlich-Kwong
SV	SV01	ESRK0	Redlich-Kwong
VV	VV01	ERSK0	Redlich-Kwong
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KVOSTLTH	Stiel-Thodos

continued

Table B.3 POLYUF: Polymer UNIFAC Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMPUF	GMPOLUF, WHENRY, HENRY, PL0XANT, ESRK0, VL0RKT, VL1BROC	Polymer UNIFAC, HENRY, Extended Antoine, Redlich-Kwong, Rackett, Brevi-O'Connell
HLMX	HLMXPUF	GMPOLUF, HL0DVK*	Polymer UNIFAC, van Krevelen
GLMX	GLMXPUF	GMPOLUF, GL0DVK	Polymer UNIFAC, van Krevelen
SLMX	SLMXPUF	GMPOLUF, HL0DVK,* GL0DVK	Polymer UNIFAC, van Krevelen
VLMX	VLMXDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHIL04	PL0XANT, ESRK0, VL0RKT	Extended Antoine, Redlich-Kwong, Rackett
HL	HLDVK	HL0DVK	van Krevelen, Aspen
GL	GLDVK	GL0DVK	van Krevelen
SL	SLDVK	HL0DVK,* GL0DVK	van Krevelen
VL	VLDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
DL	DL01	DL0WCA	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal mixing
VSMX	VSMXDVK	VS0DVK,VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

**** Optional van Krevelen/DIPPR model, HL0DVKD, available.***

Table B.4 POLYUFV: Polymer UNIFAC Free Volume Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMX01	ESRK	Redlich-Kwong
HVMX	HVMX01	ESRK	Redlich-Kwong
GVMX	GVMX01	ESRK	Redlich-Kwong
SVMX	SVMX01	ESRK	Redlich-Kwong
VVMX	VVMX01	ESRK	Redlich-Kwong
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIV01	ESRK0	Redlich-Kwong
HV	HV02	ESRK0	Redlich-Kwong
GV	GV01	ESRK0	Redlich-Kwong
SV	SV01	ESRK0	Redlich-Kwong
VV	VV01	ERSK0	Redlich-Kwong
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KVOSTLTH	Stiel-Thodos

continued

Table B.4 POLYUFV: Polymer UNIFAC Free Volume Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMUFV	GMUFFV, WHENRY, HENRY, PL0XANT, ESRK0, VL0RKT, VL1BROC, VL0TAIT	UNIFAC-FV, HENRY, Extended Antoine, Redlich-Kwong, Rackett, Brevi-O'Connell, Tait-van Krevelen
HLMX	HLMXUFV	GMUFFV, HL0DVK*	UNIFAC-FV, van Krevelen
GLMX	GLMXUFV	GMUFFV, GL0DVK	UNIFAC-FV, van Krevelen
SLMX	SLMXUFV	GMUFFV, HL0DVK,* GL0DVK	UNIFAC-FV, van Krevelen
VLMX	VLMXTDVK	VL0TAIT, VL0RKT	Tait-van Krevelen, Rackett
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHIL04	PL0XANT, ESRK0, VL0RKT	Extended Antoine, Redlich-Kwong, Rackett
HL	HLDVK	HL0DVK*	van Krevelen, Aspen
GL	GLDVK	GL0DVK	van Krevelen
SL	SLDVK	HL0DVK, GL0DVK	van Krevelen
VL	VLTDVK	VL0TAIT, VL0RKT	Tait-van Krevelen, Rackett
DL	DL01	DL0WCAV	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal mixing
VSMX	VSMXDVK	VS0DVK,VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

**** Optional van Krevelen/DIPPR model, HL0DVKD, available.***

Table B.5 POLYSL: Sanchez-Lacombe Equation-of-State Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMXSL	ESPLSL	Sanchez-Lacombe
HVMX	HVMXSL	ESPLSL	Sanchez-Lacombe
GVMX	GVMXSL	ESPLSL	Sanchez-Lacombe
SVMX	SVMXSL	ESPLSL	Sanchez-Lacombe
VVMX	VVMXSL	ESPLSL	Sanchez-Lacombe
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIVSL	ESPLSL0	Sanchez-Lacombe
HV	HVSL	ESPLSL0	Sanchez-Lacombe
GV	GVSL	ESPLSL0	Sanchez-Lacombe
SV	SVSL	ESPLSL0	Sanchez-Lacombe
VV	VVSL	ESPLSL0	Sanchez-Lacombe
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KVOSTLTH	Stiel-Thodos

continued

Table B.5 POLYSL: Sanchez-Lacombe Equation-of-State Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMXSL	ESPLSL	Sanchez-Lacombe
HLMX	HLMXSL	ESPLSL	Sanchez-Lacombe
GLMX	GLMXSL	ESPLSL	Sanchez-Lacombe
SLMX	SLMXSL	ESPLSL	Sanchez-Lacombe
VLMX	VLMXSL	ESPLSL	Sanchez-Lacombe
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHILSL	ESPLSL0	Sanchez-Lacombe
HL	HLSL	ESPLSL0	Sanchez-Lacombe
GL	GLSL	ESPLSL0	Sanchez-Lacombe
SL	SLSL	ESPLSL0	Sanchez-Lacombe
VL	VLSL	ESPLSL0	Sanchez-Lacombe
DL	DL01	DL0WCAV	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal mixing
VSMX	VSMXDVK	VS0DVK,VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

Table B.6 POLYSRK: Polymer Soave-Redlich-Kwong Equation-of-State Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMXPS	ESPLRKS	Polymer SRK
HVMX	HVMXPS	ESPLRKS	Polymer SRK
GVMX	GVMXPS	ESPLRKS	Polymer SRK
SVMX	SVMXPS	ESPLRKS	Polymer SRK
VVMX	VVMXPS	ESPLRKS	Polymer SRK
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIVPS	ESPLRKS0	Polymer SRK
HV	HVPS	ESPLRKS0	Polymer SRK
GV	GVPS	ESPLRKS0	Polymer SRK
SV	SVPS	ESPLRKS0	Polymer SRK
VV	VVPS	ESPLRKS0	Polymer SRK
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KV0STLTH	Stiel-Thodos

continued

Table B.6 POLYSRK: Polymer Soave-Redlich-Kwong Equation-of-State Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMXPS	ESPLRKS	Polymer SRK
HLMX	HLMXPS	ESPLRKS	Polymer SRK
GLMX	GLMXPS	ESPLRKS	Polymer SRK
SLMX	SLMXPS	ESPLRKS	Polymer SRK
VLMX	VLMXDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHILPS	ESPLRKS0	Polymer SRK
HL	HLPS	ESPLRKS0	Polymer SRK
GL	GLPS	ESPLRKS0	Polymer SRK
SL	SLPS	ESPLRKS0	Polymer SRK
VL	VLDVK	VL0DVK, VL0RKT	van Krevelen, Rackett
DL	DL01	DL0WCAV	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal Mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal Mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal Mixing
VVMX	VSMXDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

Table B.7 POLYSAFT: Statistical Associating Fluid Theory (SAFT) Equation-of-State Property Method

Vapor

Property Name	Route ID	Model Name	Description
PHIVMX	PHIVMXSF	ESPLSFT	SAFT
HVMX	HVMXSF	ESPLSFT	SAFT
GVMX	GVMXSF	ESPLSFT	SAFT
SVMX	SVMXSF	ESPLSFT	SAFT
VVMX	VVMXSF	ESPLSFT	SAFT
MUVMX	MUVMX02	MUV2DNST	Dean-Stiel
KVMX	KVMX02	KV2STLTH	Stiel-Thodos
DVMX	DVMX02	DV1DKK	Dawson-Khoury-Kobayashi
PHIV	PHIVSF	ESPLSFT0	SAFT
HV	HVSF	ESPLSFT0	SAFT
GV	GVSF	ESPLSFT0	SAFT
SV	SVSF	ESPLSFT0	SAFT
VV	VVSF	ESPLSFT0	SAFT
DV	DV01	DV0CEWL	Chapman-Enskog-Wilke-Lee
MUV	MUV01	MUV0BROK	Chapman-Enskog-Brokaw
KV	KV01	KV0STLTH	Stiel-Thodos

continued

Table B.7 POLYSAFT: Statistical Associating Fluid Theory (SAFT) Equation-of-State Property Method (cont.)

Liquid

Property Name	Route ID	Model Name	Description
PHILMX	PHILMXSF	ESPLSFT	SAFT
HLMX	HLMXSF	ESPLSFT	SAFT
GLMX	GLMXSF	ESPLSFT	SAFT
SLMX	SLMXSF	ESPLSFT	SAFT
VLMX	VLMXSF	ESPLSFT	SAFT
MULMX	MULMXVK	MUL2VK	van Krevelen, Letsou-Stiel
KLMX	KLMX01	KL2SRVR	Sato-Riedel, Vredeveld Mixing
DLMX	DLMX02	DL1WCA	Wilke-Chang-Andrade
SIGLMX	SIGLMX01	SIG2HSS	Hakim-Steinberg-Stiel, Power Law Mixing
PHIL	PHILSF	ESPLSFT0	SAFT
HL	HLSF	ESPLSFT0	SAFT
GL	GLSF	ESPLSFT0	SAFT
SL	SLSF	ESPLSFT0	SAFT
VL	VLSF	ESPLSFT0	SAFT
DL	DL01	DL0WCAV	Wilke-Chang-Andrade
MUL	MULMH	MUL0MH	Modified Mark-Houwink, Andrade
KL	KL01	KL0SR	Sato-Riedel

Solid

Property Name	Route ID	Model Name	Description
HSMX	HSMXDVK	HS0DVK	van Krevelen, Ideal Mixing
GSMX	GSMXDVK	GS0DVK	van Krevelen, Ideal Mixing
SSMX	SSMXDVK	HS0DVK, GS0DVK	van Krevelen, Ideal Mixing
VVMX	VSMXDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial
HS	HSDVK	HS0DVK	van Krevelen
GS	GSDVK	GS0DVK	van Krevelen
SS	SSDVK	HS0DVK, GS0DVK	van Krevelen
VS	VSDVK	VS0DVK, VS0POLY	van Krevelen, Polynomial

C

VAN KREVELEN FUNCTIONAL GROUPS

This appendix lists the van Krevelen functional group parameters. These functional groups used by the van Krevelen property models are listed by category:

CALCULATING SEGMENT PROPERTIES FROM FUNCTIONAL GROUPS

The van Krevelen property models use functional groups to calculate segment property parameters, which are in turn used to calculate polymer property parameters. The functional group parameters listed in Table C.1 are used to calculate segment properties using the following correlations:

Heat Capacity (Liquid or Crystalline)

$$Cp = \sum_k n_k Cp_k$$

Where:

Cp_k = heat capacity for the functional groups in Table C.1

Molar Volume (Liquid, Crystalline, or Glassy)

$$V_k = Vw_k * (A + B * T + C * Tg)$$

$$V = \sum_k n_k V_k$$

Where:

Vw_k = van der Waals volume for the functional groups in Table C.1

A, B, and C = empirical constants and vary by phase

Enthalpy of Formation

$$\Delta H_f^o = \sum_k n_k \Delta G_{fAk}^o$$

$$\Delta S_f^o = \sum_k n_k \Delta G_{fBk}^o$$

Where:

ΔG_{fAk}^o and ΔG_{fBk}^o = heat balance parameters in Table C.1

Glass Transition Temperature

$$T_g = \frac{\sum_k n_k Yg_k}{\sum_k n_k MW_k}$$

Where:

Yg_k = glass transition parameter in Table C.1

MW_k = functional group molecular weight

Melt Transition Temperature

$$T_m = \frac{\sum_k n_k Ym_k}{\sum_k n_k MW_k}$$

Where:

Ym_k = glass transition parameter in Table C.1

MW_k = functional group molecular weight

Viscosity-Temperature Gradient

$$H_\eta = \sum_k n_k H_{\eta k} \; / \; M$$

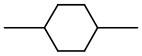
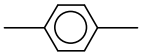
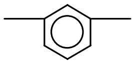
Where:

H_η = viscosity-temperature gradient

$H_{\eta k}$ = viscosity-temperature gradient of group k value listed in Table C.1

n_k = number of occurrence of group k

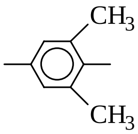
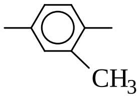
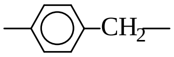
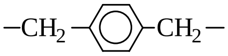
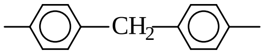
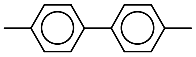
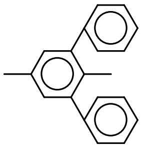
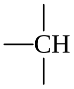
Table C.1 Van Krevelen Functional Groups

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Bifunctional hydrocarbon groups										
$—CH_2—$	100	14.03	10.23	25.35	30.4	-22,000	102	2,700	5,700	420
$—CH(CH_3)—$	101	28.05	20.45	46.5	57.85	-48,700	215	8,000	13,000	1060
	102							8,000	13,000	----
$—CH(C_5H_9)—$	103	82.14	53.28	110.8	147.5	-73,400	548	30,700	----	----
$—CH(C_6H_{11})—$	104	96.17	63.58	121.2	173. 9	-118,400	680	41,300	----	----
$—CH(C_6H_5)—$	105	90.12	52.62	101.2	144.15	84,300	287	36,100	48,000	3600
$—C(CH_3)_2—$	106	42.08	30.67	68.0	81.2	-72,000	330	8,500	12,100	1620
$—C(CH_3)(C_6H_5)—$	107	104.14	62.84	122.7	167.5	61,000	402	51,000	54,000	----
						76,000	76			
$—CH{=}CH—$	108	26.04	16.94	37.3	42.8	70,000	83	3,800	8,000	----
	109							7,400	11,000	----
$—CH{=}C(CH_3)—$	110	40.06	27.16	60.05	74.22	42,000	183	8,100	10,000	----
	111					36,000	190	9,100	13,000	----
$—C{\equiv}C—$	112	24.02	16.1	----	----	230,000	-50	----	----	----
	113	82.14	53.34	103.2	147.5	-96,400	578	19,000	31,000	----
	114					-102,400	585	27,000	45,000	----
	115	76.09	43.32	78.8	113.1	100,000	180	29,000	38,000	3200
	116	76.09	43.32	78.8	113.1	100,000	180	25,000	31,000	----
	117	76.09	43.32	78.8	113.1	100,000	180	----	----	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

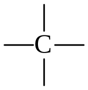
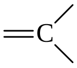
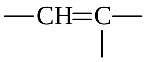
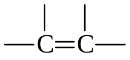
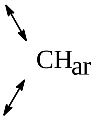
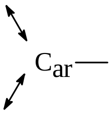
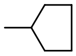
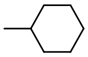
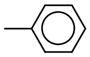
Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Bifunctional hydrocarbon groups (continued)										
	118	104.14	65.62	126.8	166.8	33,000	394	54,000	67,000	----
	119	90.12	54.47	102.75	140.1	66,500	287	35,000	45,000	----
	120	90.12	53.55	104.15	143.5	78,000	282	----	----	----
	121							----	----	----
	122	104.14	63.78	129.5	173.9	56,000	384	25,000	47,000	----
	123	166.21	96.87	182.95	256.6	178,000	462	65,000	85,000	----
	124	152.18	86.64	157.6	226.2	200,000	360	70,000	99,000	---
	125	228.22	130	236	339	299,000	538	118,000	173,000	----
Other hydrogen-containing groups										
$-CH_3$	126	15.03	13.67	30.9	36.9	-46,000	95	----	----	810
$-C_2H_5$	127	29.06	23.90	56.25	67.3	-68,000	197	----	----	----
$-nC_3H_7$	128	43.09	34.13	81.6	97.7	-90,000	299	----	----	----
$-iC_3H_7$	129	43.09	34.12	77.4	94.75	-94,700	310	----	----	----
$-tC_4H_9$	130	57.11	44.34	99.0	118.1	-118,000	425	----	----	----
 $\overset{\vert}{\underset{\vert}{-CH}}$	131	13.02	6.78	15.9	20.95	-2,700	120	----	----	250

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

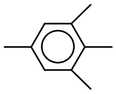
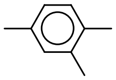
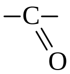
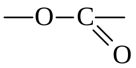
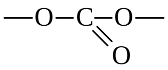
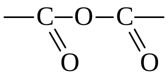
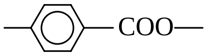
Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Other hydrogen-containing groups (continued)										
 $-\overset{|}{\underset{|}{C}}-$	132	12.01	3.33	6.2	7.4	20,000	140	----	----	0
$=CH_2$	133	14.01	11.94	22.6	21.8	23,000	30	----	----	----
$=CH-$	134	13.02	8.47	18.65	21.8	38,000	38	----	----	380
	135	12.01	5.01	10.5	15.9	50,000	50	----	----	0
$=C=$	136	12.01	6.96	----	----	147,000	-20	----	----	----
 $-CH=\underset{|}{C}-$	137	25.03	13.48	29.15	37.3	88,000	88	----	----	----
$\equiv CH$	138	13.02	8.05	----	----	112,500	-32.5	----	----	----
$\equiv C-$	139	12.01	8.05	----	----	115,000	-25	----	----	----
 $-\overset{|}{C}=\overset{|}{C}-$	140	24.02	10.02	21.0	31.8	100,000	100	----	----	----
						94,000	107	----	----	----
	142	13.02	8.06	15.4	22.2	12,500	26	----	----	----
	143	12.01	5.54	8.55	12.2	25,000	38	----	----	----
	144	69.12	45.56	95.2	126.55	-70,700	428	----	----	----
	145	83.15	56.79	105.6	152.95	-115,700	560	----	----	----
	146	77.10	45.84	85.6	123.2	87,000	167	----	----	3350

***Source:* Van Krevelen, D.W., *Properties of Polymers*, 3rd Ed., Elsevier, Amsterdam (1990).**

continued

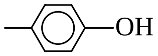
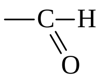
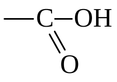
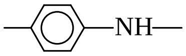
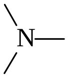
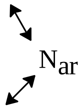
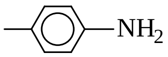
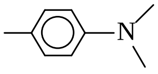
Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Other hydrogen-containing groups (continued)										
	147	74.08	38.28	65.0	93.0	125,000	204	----	----	----
	148	75.08	40.80	71.85	103.2	112,500	192	----	----	----
Bifunctional oxygen-containing groups										
$-O-$	149	16.00	3.71/ [5.1][1]	16.8	35.6	-120,000	70	4,000	13,500	480
 $-\underset{\underset{O}{\Vert}}{C}-$	150	28.01	11.7	23.05	52.8	-132,000	40	9,000	12,000	----
 $-O-\underset{\underset{O}{\Vert}}{C}-$	151	44.01	15.2/ [17.0][1]	(46)	65.0	-337,000	116	12,500	30,000	1450
	152							12,500	30,000	----
 $-O-\underset{\underset{O}{\Vert}}{C}-O-$	153	60.01	18.9/ [23.0][1]	(63)	----	----	----	20,000	30,000	3150
 $-\underset{\underset{O}{\Vert}}{C}-O-\underset{\underset{O}{\Vert}}{C}-$	154	72.02	(27)	(63)	(114)	----	----	22,000	35,000	----
— CH(OH)—	155	30.03	14.82	32.6	65.75	-178,700	170	13,000	----	----
— CH(COOH)—	156	58.04	----	(65.6)	119.85	-395,700	238	----	----	----
— CH(HC=O)—	157	42.14	21.92	----	----	-127,700	146	----	----	----
	158	120.10	58.52	(124.8)	178.1	-237,000	296	----	----	----
$-O-CH_2-O-$	159	46.03	17.63	58.95	101.6	-262,000	242	10,700	32,700	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

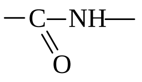
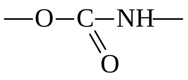
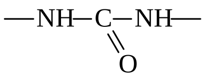
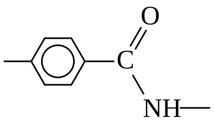
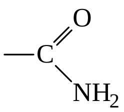
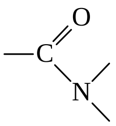
Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Other oxygen-containing groups										
$-OH$	160	17.01	8.04	17.0	44.8	-176,000	50	---	----	----
	161	93.10	51.36	95.8	157.9	-76,000	230	----	----	----
 $-\underset{\overset{\|}{O}}{C}-H$	162	29.02	15.14	----	----	-125,000	26	----	----	----
 $-\underset{\overset{\|}{O}}{C}-OH$	163	45.02	----	(50)	98.9	-393,000	118	----	----	----
Bifunctional nitrogen-containing groups										
$-NH-$	164	15.02	8.08	14.25	(31.8)	58,000	120	7,000	18,000	----
$-CH(CN)-$	165	39.04	21.48	(40.6)	----	120,300	91.5	----	----	----
$-CH(NH_2)-$	166	29.04	17.32	36.55	----	8,800	222.5	----	----	----
	167	91.11	(51.4)	93.05	(144.9)	158,000	300	----	----	----
Other nitrogen-containing groups										
$-NH_2$	168	16.02	10.54	20.95	----	11,500	102.5	----	----	----
	169	14.01	4.33	17.1	(44.0)	97,000	150	----	----	----
	170	14.01	----	----	----	69,000	50	----	----	----
$-C\equiv N$	171	26.02	14.7	(25)	----	123,000	-28.5	----	----	----
	172	92.12	53.86	99.75	----	111,500	282.5	----	----	----
	173	90.10	47.65	95.9	(157.1)	197,000	330	----	----	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Bifunctional nitrogen- and oxygen-containing groups										
 $-\underset{\underset{O}{\|}}{C}-NH-$	174	43.03	19.56 [18.1][1]	(38/54)	(90.1)	-74,000	160	15,000	45,000	1650
 $-O-\underset{\underset{O}{\|}}{C}-NH-$	175	59.03	(23)	(58)	----	-279,000	-240	20,000	43,500	----
 $-NH-\underset{\underset{O}{\|}}{C}-NH-$	176	58.04	(27.6)	(50)	----	-16,000	280	20,000	60,000	----
$-CH(NO_2)-$	177	59.03	23.58	57.5	----	-44,200	263	----	----	----
	178	119.12	62.88	(116.8/1 32.8)	(203.2)	----	----	7,000	98,000	----
Other nitrogen- and oxygen-containing groups										
	179	44.03	(22.2)	----	----	----	----	----	----	----
	180	42.02	(16.0)	----	----	----	----	----	----	----
$-NO_2$	181	46.01	16.8	41.9	----	-41,500	143	----	----	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

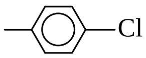
Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Bifunctional sulphur-containing groups										
$-\mathrm{S}-$	182	32.06	10.8	24.05	44.8	40,000	-24	8,000	22,500	----
$-\mathrm{S}-\mathrm{S}-$	183	64.12	22.7	(48.1)	(89.6)	46,000	-28	16,000	30,000	----
$-\mathrm{SO_2}-$	184	64.06	20.3	(50)	----	-282,000	152	32,000	56,000	----
$-\mathrm{S}-\mathrm{CH_2}-\mathrm{S}-$	185	78.15	31.8	73.45	120.0	58,000	54	----	----	----
Other sulphur-containing groups										
$-\mathrm{SH}$	186	33.07	14.81	46.8	52.4	13,000	-33	----	----	----
Bifunctional halogen-containing groups										
$-\mathrm{CHF}-$	187	32.02	13.0	(37.0)	(41.95)	-197,700	114	12,400	17,400	----
$-\mathrm{CF_2}-$	188	50.01	15.3	(49.0)	(49.4)	-370,000	128	10,500	25,500	----
$-\mathrm{CHCl}-$	189	48.48	19.0	42.7	(60.75)	-51,700	111	19,400	27,500	2330
$-\mathrm{CCl_2}-$	190	82.92	27.8	60.4	(87.0)	-78,000	122	22,000	29,000	----
$-\mathrm{CH}{=}\mathrm{CCl}-$	191	60.49	25.72	56.25	(77.1)	39,000	79	15,200	32,000	----
$-\mathrm{CFCl}-$	192	66.47	21.57	(54.7)	(68.2)	-224,000	125	28,000	32,000	----
$-\mathrm{CHBr}-$	193	92.93	21.4	41.9	----	-16,700	106	----	----	----
$-\mathrm{CBr_2}-$	194	171.84	32.5	58.8	----	-8,000	112	----	----	----
$-\mathrm{CHI}-$	195	139.93	27.1	38.0	----	37,300	79	----	----	----
$-\mathrm{CI_2}-$	196	265.83	44.0	51.0	----	100,000	58	----	----	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

continued

Table C.1 Van Krevelen Functional Groups (cont.)

Functional Group	Group No.	MW g/mol	V_w cm^3 / mol	Cp_c J/mol-K	Cp_l J/mol-K	ΔG^o_{fA}	ΔG^o_{fB}	Y_g	Y_m	H_η
Other halogen-containing groups										
—F	197	19.00	6.0	(21.4)	(21.0)	-195,000	-6	----	----	----
—CF_3	198	69.01	21.33	(70.4)	(70.4)	-565,000	122	----	----	----
—CHF_2	199	51.02	18.8	(58.4)	(62.95)	-392,700	108	----	----	----
—CH_2F	200	33.03	16.2	(46.75)	(51.4)	-217,000	96	----	----	----
—Cl	201	35.46	12.2	27.1	(39.8)	-49,000	-9	----	----	2080
—CCl_3	202	118.38	(40)	87.5	(126.8)	-127,000	113	----	----	----
—$CHCl_2$	203	83.93	31.3	69.8	(100.55)	100,700	102	----	----	----
—CH_2Cl	204	49.48	22.5	52.45	(70.2)	-71,700	93	----	----	----
	205	111.55	55.3	105.9	(152.9)	51,000	171	----	----	----
—Br	206	79.92	14.6	26.3	----	-14,000	-14	----	----	----
—CBr_3	207	251.76	(47.1)	85.1	----	-22,000	98	----	----	----
—$CHBr_2$	208	172.85	36.0	68.2	----	-30,700	92	----	----	----
—CH_2Br	209	93.94	24.8	51.65	----	-36,000	88	----	-----	----
—I	210	126.91	20.4	22.4	----	40,000	-41	----	----	----
—CI_3	211	392.74	(64.4)	73.4	----	140,000	17	----	----	----
—CHI_2	212	266.84	47.5	60.4	----	77,300	38	----	----	----
—CH_2I	213	140.94	30.6	47.75	----	18,000	61	----	-----	----

Source: Van Krevelen, D.W., Properties of Polymers, 3rd Ed., Elsevier, Amsterdam (1990).

D TAIT MODEL COEFFICIENTS

This appendix lists parameters available for the Tait molar volume calculations for selected polymers.

These parameters are available in POLYMER for the polymers listed in Table D.1.

Table D.1 Tait Model Coefficients

Polymer	A_0 (m^3 / kg)	A_1 (m^3 / kg.K)	A_2 (m^3 / kg.K^2)	B_0 (Pa)	B_1 (1/K)	P Range † (Mpa)	T Range † (K)
BR	1.0969E-03	7.6789E-07	-2.2216E-10	1.7596E+08	4.3355E-03	0.1-283	277-328
HDPE	1.1567E-03	6.2888E-07	1.1268E-09	1.7867E+08	4.7254E-03	0.1-200	415-472
I-PB	1.1561E-03	6.1015E-07	8.3234E-10	1.8382E+08	4.7833E-03	0.0-196	407-514
I-PMMA	7.9770E-04	5.5274E-07	-1.4503E-10	2.9210E+08	4.1960E-03	0.1-200	328-463
I-PP	1.2033E-03	4.8182E-07	7.7589E-10	1.4236E+08	4.0184E-03	0.0-196	447-571
LDPE	1.1004E-03	1.4557E-06	-1.5749E-09	1.7598E+08	4.6677E-03	0.1-200	398-471
LLDPE	1.1105E-03	1.2489E-06	-4.0642E-10	1.7255E+08	4.4256E-03	0.1-200	420-473
PAMIDE	7.8153E-04	3.6134E-07	2.7519E-10	3.4019E+08	3.8021E-03	0.0-177	455-588
PBMA	9.3282E-04	5.7856E-07	5.7343-10	2.2569E+08	5.3116E-03	0.1-200	295-473
PC	7.9165E-04	4.4201E-07	2.8583E-10	3.1268E+08	3.9728E-03	0.0-177	430-610
PCHMA	8.7410E-04	4.9035E-07	3.2707E-10	3.0545E+08	5.5030E-03	0.1-200	383-472
PDMS	1.0122E-03	7.7266E-07	1.9944E-09	8.7746E+07	6.2560E-03	0.0-100	298-343
PHENOXY	8.3796E-04	3.6449E-07	5.2933E-10	3.5434E+08	4.3649E-03	0.0-177	349-574
PIB	1.0890E-03	2.5554E-07	2.2682E-09	1.9410E+08	3.9995E-03	0.0-100	326-383
PMMA	8.2396E-04	3.0490E-07	7.0201E-10	2.9803E+08	4.3789E-03	0.1-200	387-432
PMP	1.2078E-03	5.1461E-07	9.7366E-10	1.4978E+08	4.6302E-03	0.0-196	514-592
POM	8.3198E-04	2.7550E-07	2.2000E-09	3.1030E+08	4.4652E-03	0.0-196	462-492
POMS	9.3905E-04	5.1288E-07	5.9157E-11	2.4690E+08	3.6633E-03	0.1-180	413-471
PS-1	9.3805E-04	3.3086E-07	6.6910E-10	2.5001E+08	4.1815E-03	0.1-200	389-469
PTFE	4.6867E-04	1.1542E-07	1.1931E-09	4.0910E+08	9.2556E-03	0.0-392	604-646
PVAC	8.2832E-04	4.7205E-07	1.1364E-09	1.8825E+08	3.8774E-03	0.0-100	337-393

† Range of experimental data used in the determination of equation constants.

Source: Danner, R.P. and High, M.S. "Handbook of Polymer Solution Thermodynamics." Design Institute for Physical Property Data, American Institute of Chemical Engineers. 3B-5.

E MASS BASED PROPERTY PARAMETERS

The Aspen Plus convention is to use mole based parameters for property model calculations. However, polymer property parameters are often more conveniently obtained on a mass basis. To satisfy the needs of users who may prefer the use of mass based parameters, in Polymers Plus there is a corresponding mass based parameter for selected mole based parameters.

Table E.1 shows a list of model parameters and their mass-based counterparts. Note that the mass based parameters should only be used for polymers and oligomers, and not for segments.

Table E.1 Mass Based Property Parameters

Mole Based Parameter	Mass Based Parameter	Description
CPCVK	CPCVKM	Crystalline heat capacity
CPLVK	CPLVKM	Liquid heat capacity
DGCON	DGCONM	Standard free energy of condensation
DGFORM	DGFVKM	Standard free energy on formation at 25°C
DBSUB	DGSUBM	Standard free energy of sublimation
DHCON	DHCONM	Standard enthalpy of condensation
DHFVK	DHFVKM	Standard enthalpy of formation at 25°C
DHSUB	DHSUBM	Standard enthalpy of sublimation
DNCVK	DNCVKM	Crystalline density
DNGVK	DNGVKM	Glass density
DNLVK	DNLVKM	Liquid density

F

EQUATION-OF-STATE PARAMETERS

This appendix lists unary parameters available for use with the Sanchez-Lacombe equation of state model (POLYSL property method) and the SAFT (POLYSAFT) equation of state model. Parameters are available for selected:

• Polymers (SL)	(F•2)
• Monomers (SL)	(F•3)
• Solvents (SL)	(F•3)
• Solvents and Polymers (SAFT)	(F•6)

The parameters are not automatically retrieved from databanks. These parameters are not unique in any way. Users may generate them through experimental data regression for the components of interest.

Table F.1 lists the Sanchez-Lacombe unary parameters for polymers while Table F.2 lists the Sanchez-Lacombe unary parameters for monomers and solvents. Table F.3 lists the SAFT unary parameters for solvents, monomers and polymers.

Table F.1 Sanchez-Lacombe Unary Parameters for Polymers

Polymer	T*, K	P*, bar	ρ*, kg / m^3	T range, K	P, up to bar
HDPE	649	4250	904	426-473	1000
LDPE	673	3590	887	408-471	1000
PDMS	476	3020	1104	298-343	1000
PBMA	627	4310	1125	307-473	2000
PHMA	697	4260	1178	398-472	2000
PIB	643	3540	974	326-383	1000
PMMA	696	5030	1269	397-432	2000
POMS	768	3780	1079	412-471	1600
PS	735	3570	1105	388-468	2000
PVAC	590	5090	1283	308-373	800

Source:* *Sanchez, I. C. and R. H. Lacombe, Statistical Thermodynamics of Polymer Solutions, Macromolecules,* 11*(6), 1145-1156 (1978).

Table F.2 Sanchez-Lacombe Unary Parameters for Monomers and Solvents

Formula	**Component**	**T*, K**	**P*, bar**	**ρ*, kg / m^3**
CCl4	Carbon Tetrachloride	535	8126	1788
CHCl3	Chloroform	512	4560	1688
CH3Cl	Methyl chloride	487	5593	1538
CO2	Carbon dioxide †	277	7436	1629
CS2	Carbon disulfide	567	5157	1398
CH4	Methane	224	2482	500
CH4O	Methanol	468	12017	922
C3H3N	Acrilonitrile †	527	5930	868
C3H6O	Acetone	484	5330	917
C3H6O2	Ethyl formate	466	4965	1076
C6H7N	Aniline	614	6292	1115
C3H8O	Propanol	420	8856	972
C3H8O	Isopropyl alcohol	399	8532	975
CH3NO2	Nitromethane	620	9251	1490
C2HCl3	1,1,1-Trichloroethylene	516	3779	1518
C2H2Cl2	1,1-Dichloroethylene	488	5117	1722
C2H4	Ethylene †	291	3339	660
C2H4O2	Acetic acid	562	8613	1164
C2H6	Ethane	315	3273	640
C2H6O	Ethanol	413	10690	963
C3H8	Propane	371	3131	690
C4H8O	Methyl ethyl ketone	513	4468	913
C4H8O2	Ethyl acetate	468	4580	1052
C4H10	n-Butane	403	3222	736
C4H10	Isobutane	398	2878	720

† Evaluated from vapor-pressure and liquid-density data regression

***Source:** Sanchez, I. C. and R. H. Lacombe, Statistical Thermodynamics of Polymer Solutions, Macromolecules, 11(6), 1145-1156 (1978).*

continued

Table F.2 Sanchez-Lacombe Unary Parameters for Monomers and Solvents (cont.)

Formula	Component	T*, K	P*, bar	ρ*, kg / m^3
C4H10O	Tert-butyl alcohol	448	6931	952
C4H10O	Diethyl ether	431	3627	870
C5H5N	Pyridine	566	5492	1079
C5H10	Cyclopentane	491	3800	867
C5H12	n-Pentane	441	3101	755
C5H12	Isopentane	424	3080	765
C5H12	Neopentane	415	2655	744
C6H5Cl	Chlorobenzene	585	4367	1206
C6H6	Benzene	523	4438	994
C6H6O	Phenol	530	7934	1192
C6H14	n-Hexane	476	2979	775
C6H12	Cyclohexane	497	3830	902
C6H12O2	n-Butyl acetate	498	3942	1003
C7H8	Toluene	543	4023	966
C7H16	n-Heptane	487	3090	800
C8H8	Styrene †	563	3684	870
C8H10	p-Xylene	561	3810	949
C8H10	m-Xylene	560	3850	952
C8H10	o-Xylene	571	3942	965
C8H18	n-Octane	502	3080	815
C9H20	n-Nonane	517	3070	828
C10H18	trans-Decalin	621	3151	935
C10H18	cis-Decalin	631	3334	960

† Evaluated from vapor-pressure and liquid-density data regression

Source:* *Sanchez, I. C. and R. H. Lacombe, Statistical Thermodynamics of Polymer Solutions, Macromolecules, 11(6), 1145-1156 (1978).

continued

Table F.2 Sanchez-Lacombe Unary Parameters for Monomers and Solvents (cont.)

Formula	Component	T*, K	P*, bar	ρ*, kg / m^3
C10H22	n-Decane	530	3040	837
C11H24	n-Undecane	542	3030	846
C12H26	n-Dodecane	552	3009	854
C13H28	n-Tridecane	560	2989	858
C14H10	Phenanthrene	801	3769	1013
C14H30	n-Tetradecane	570	2959	864
C17H36	n-Heptadecane	596	2867	880
C20H42	n-Eicosane †	617	3067	961
H2O	Water	623	26871	1105
H2S	Hydrogen Sulfate	382	6129	1095

† Evaluated from vapor-pressure and liquid-density data regression

Source: Sanchez, I. C. and R. H. Lacombe, Statistical Thermodynamics of Polymer Solutions, Macromolecules, *11* ***(6), 1145-1156 (1978).***

Table F.3 SAFT Unary Parameters for Various Non-Associating Fluids

Formula	**Component**	**T range, K**	**v^{oo}, cm^3 / mol**	**m**	**u^o / k, K**
N2	Nitrogen	---	19.457	1.0	123.53
AR	Argon	---	16.29	1.0	150.86
CO	Carbon Monoxide	72-121	15.776	1.221	111.97
CO2	Carbon Dioxide	218-288	13.578	1.417	216.08
CL2	Chlorine	180-400	22.755	1.147	367.44
CS2	Carbon Disulfide	278-533	23.622	1.463	396.05
SO2	Sulfur Dioxide	283-413	22.611	1.133	335.84
CH4	Methane	92-180	21.576	1.0	190.29
C2H6	Ethane	160-300	14.460	1.941	191.44
C3H8	Propane	190-360	13.457	2.696	193.03
C4H10	n-Butane	220-420	12.599	3.458	195.11
C5H12	n-Pentane	233-450	12.533	4.091	200.02
C6H14	n-Hexane	243-493	12.475	4.724	202.72
C7H16	n-Heptane	273-523	12.282	5.391	204.61
C8H18	n-Octane	303-543	12.234	6.045	206.03
C9H20	n-Nonane	303-503	12.240	6.883	203.56
C10H22	n-Decane	313-573	11.723	7.527	205.46
C12H26	n-Dodecane	313-523	11.864	8.921	205.93
C14H30	n-Tetradecane	313-533	12.389	9.978	209.40
C16H34	n-Hexadecane	333-593	12.300	11.209	210.65
C20H42	n-Eicosane	393-573	12.0	13.940	211.25
C28H58	n-Octacosane	449-704	12.0	19.287	209.96
C36H74	n-Hexatriacontane	497-768	12.0	24.443	208.74
C44H90	n-Tetratetracontane	534-725	12.0	29.252	207.73
C5H10	Cyclopentane	252-483	12.469	3.670	226.70
C6H12	Methyl-cyclopentane	263-503	13.201	4.142	223.25
C7H14	Ethyl-cyclopentane	273-513	13.766	4.578	229.04
C8H16	Propyl-cyclopentane	293-423	14.251	5.037	232.18

continued

Source: Huang, S. H. and M. Radosz, Equation of State for Small, Large, Polydisperse, and Associating Molecules, Ind., Eng. Chem. Res., 29, 2284-2294 (1990).

Table F.3 SAFT Unary Parameters for Various Non-Associating Fluids (cont.)

Formula	Component	T range, K	v^{∞}, cm^3 / mol	m	u^o / k, K
C9H18	Butyl-cyclopentane	314-578	14.148	5.657	230.61
C10H20	Pentyl-cyclopentane	333-483	13.460	6.503	225.56
C6H12	Cyclohexane	283-513	13.502	3.970	236.41
C7H14	Methylcyclohexane	273-533	15.651	3.954	248.44
C8H16	Ethylcyclohexane	273-453	15.503	4.656	243.16
C9H18	Propylcyclohexane	313-453	15.037	5.326	238.51
C10H20	Butylcyclohexane	333-484	14.450	6.060	234.30
C11H22	Pentylcyclohexane	353-503	14.034	6.804	230.91
C6H6	Benzene	300-540	11.421	3.749	250.19
C7H8	Methyl-benzene	293-533	11.789	4.373	245.27
C8H10	Ethyl-benzene	293-573	12.681	4.719	248.79
C9H12	n-Propyl-benzene	323-573	12.421	5.521	238.66
C10H14	n-Butyl-benzene	293-523	12.894	6.058	238.19
C8H10	m-Xylene	309-573	12.184	4.886	245.88
C12H10	Biphenyl	433-653	12.068	6.136	280.54
C10H8	Naphthalene	373-693	13.704	4.671	304.80
C11H10	1-Methyl-naphthalene	383-511	13.684	5.418	293.45
C12H12	1-Ethyl-naphthalene	393-563	12.835	6.292	276.18
C13H14	1-n-Propyl-naphthalene	403-546	13.304	6.882	266.82
C14H16	1-n-Butyl-naphthalene	413-566	13.140	7.766	252.11
C14H10	Phenanthrene	373-633	16.518	5.327	352.00
C14H10	Anthracene	493-673	16.297	5.344	352.65
C16H10	Pyrene	553-673	18.212	5.615	369.38
C2H6O	Dimethyl-ether	179-265	11.536	2.799	207.83
C3H8O	Methyl-ethyl-ether	266-299	10.065	3.540	203.54
C4H10O	methyl-n-propyl-ether	267-335	10.224	4.069	208.13

continued

Source: ***Huang, S. H. and M. Radosz, Equation of State for Small, Large, Polydisperse, and Associating Molecules, Ind., Eng. Chem. Res., 29, 2284-2294 (1990).***

Table F.3 SAFT Unary Parameters for Various Non-Associating Fluids (cont.)

Formula	**Component**	**T range, K**	**v^{oo}, cm^3 / mol**	**m**	**u^o / k, K**
C4H10O	Diethyl-ether	273-453	10.220	4.430	191.92
C12H10O	Phenyl-ether	523-633	12.100	6.358	276.13
C3H9N	Trimethylamine	193-277	14.102	3.459	196.09
C12H10O	Triethylamine	323-368	11.288	5.363	201.31
C3H6O	Acetone	273-492	7.765	4.504	210.92
C4H8O	Methy-ethyl ketone	257-376	11.871	4.193	229.99
C5H10O	Methyl-n-propyl ketone	274.399	11.653	4.644	230.40
C5H10O	Diethyl-ketone	275-399	10.510	4.569	235.24
C2H4	Ethylene	133-263	18.157	1.464	212.06
C3H6	Propylene	140-320	15.648	2.223	213.90
C4H8	1-Butene	203-383	13.154	3.162	202.49
C6H12	1-Hexene	213-403	12.999	4.508	204.71
CH3CL	Chloromethane	213-333	10.765	2.377	238.37
CH2CL2	Dichloromethane	230-333	10.341	3.114	253.03
CHCL3	Trichloromethane	244-357	10.971	3.661	240.31
CCL4	Tetrachloromethane	273-523	13.730	3.458	257.46
C2H5CL	Chloroethane	212-440	11.074	3.034	229.58
C3H7CL	1-Chloropropane	238-341	11.946	3.600	229.14
C4H7CL	1-Chlorobutane	262-375	12.236	4.207	227.88
C6H11CL	1-Chlorohexane	306-435	12.422	5.458	225.82
C6H5CL	Chlorobenzene	273-543	13.093	3.962	276.72
PE	Polyethylene (MW=25000)	413-473	12.0	1165.77	210.0
P(E&P)	Polypropylene	263-303	12.0	822.68	210.0

G

KINETIC RATE CONSTANT PARAMETERS

This appendix provides decomposition rate parameters for commonly used initiators. Within each group the initiators are arranged by increasing total number of carbon atoms.

The parameters are grouped as follows:

INITIATOR DECOMPOSITION RATE

Table G.1 lists decomposition rate parameters for free-radical initiators. These parameters assume first-order kinetics.†

The initiator concentration I (in SI units) is related to decomposition rate $k_d[1/\mathrm{sec}]$ as follows:

$$\frac{d[I]}{dt} = -k_d[I]$$

The reaction half-life is related to the decomposition rate by the equation:

$$t_{1/2} = \frac{\ln 2}{k_d}$$

Where:

A = frequency factor

E_a = Arrhenius activation energy

T = absolute temperature

R = universal gas constant

The decomposition rate, k_d, can be found with the equation:

$$k_d = A * \exp(-E_a / RT)$$

Assuming the activation energy, E_a, and the decomposition rate, k_o, at some absolute temperature, T_o, are known, the decomposition rate at another temperature can be found using:

$$\ln k = \ln k_o - \frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T_o}\right)$$

Where:

R = universal gas constant equal to 8314.39 J/kmol*K

---

† Masson, J.C, "Decomposition Rates of Organic Free Radical Initiators," Polymer Handbook, 3rd ed., Wiley, II/1 (1989).

The values for Table G.1 were obtained from a logarithmic plot of temperature vs. half-life. The half-life values were determined in benzene or in chlorobenzene in concentrations of 0.1 to 0.2 M, and they were correlated with polymerization results in some major monomers.

Two data points were chosen from the graph for each initiator usually by reading the temperatures for the half-lives of 10s and 10hrs: (half-life1$(t_1) = 10\text{s}$, temperature1(T_1)) and (half-life2$(t_2) = 10\text{hrs}$, temperature2(T_2)). The two points were used to find the slope $(-E_a / R)$ and the y-intercept (ln A) for the linear regression curve below:

$$\ln k_d = \ln A - \frac{E_a}{R} * \frac{1}{T}$$

After some rearrangement, the above equation yields the following two equations that determine E_a and A for each initiator:

$$E_a = R * \left(\ln \frac{t_1}{t_2} \right) * \left(\frac{T_2 * T_1}{T_2 - T_1} \right)$$

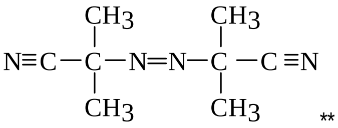
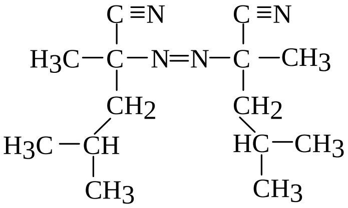
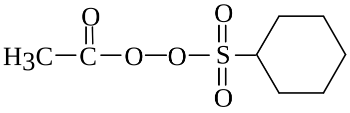
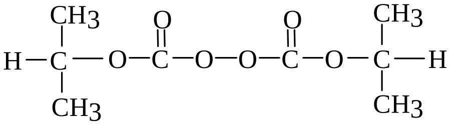
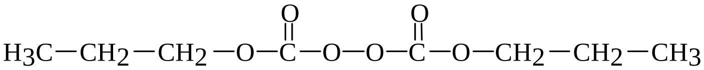
$$A = \frac{\ln 2}{t_2} * \exp \frac{E}{RT_2}$$

Where:

t_1 = initiator half-life at temperature T_1

t_2 = initiator half-life at temperature T_2

Table G.1 Initiator Decomposition Rate Parameters

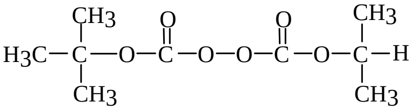
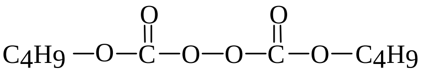
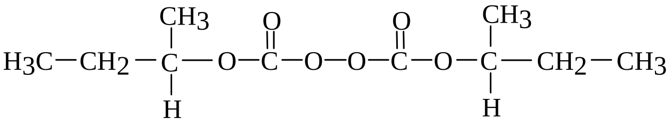
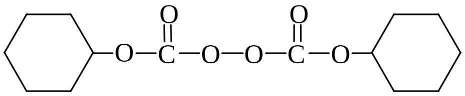
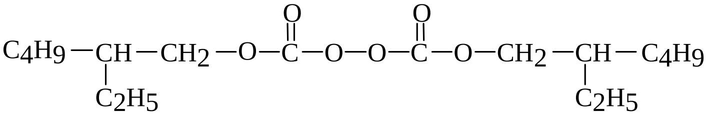
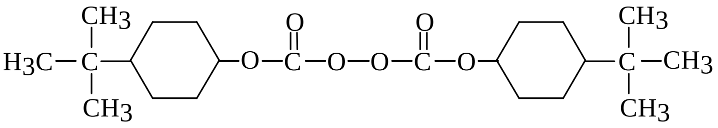
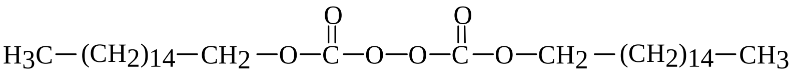
Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Azo-Nitriles							
	2,2'-azo-bis-isobutyronitrile  $N{\equiv}C-C(CH_3)_2-N{=}N-C(CH_3)_2-C{\equiv}N$ **	164.212	128.8	1.82E+15	391	355	337
	2,2'-azo-bis(2,4-dimethylvaleronitrile) 	248.374	116.3	1.88E+14	375	338	320
Sulfonyl Peroxides							
	acetyl cyclohexane sulphonyl peroxide 	222.261	133.9	7.14E+17	354	324	310
Peroxydicarbonates							
	di-isopropyl peroxydicarbonate  $H-C(CH_3)_2-O-C(=O)-O-O-C(=O)-O-C(CH_3)_2-H$	206.194	125.0	7.11E+15	367	334	318
	di-n-propyl peroxydicarbonate  $H_3C-CH_2-CH_2-O-C(=O)-O-O-C(=O)-O-CH_2-CH_2-CH_3$	206.194	125.9	9.44E+15	367	334	318

continued

** ***Budavari, S., "931. 2,2'-Azobisisobutyronitrile," The Merck Index: An Encyclopedia of Chemicals, Drugs, and Biologicals, Eleventh Ed., Merck & Co., 146 (1989).***

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

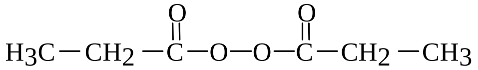
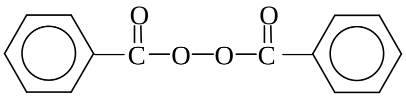
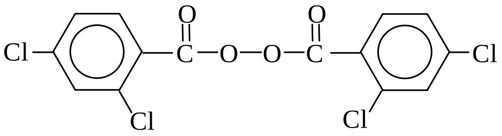
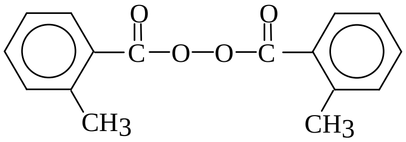
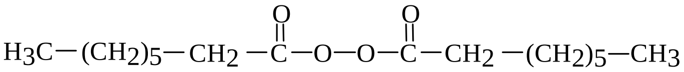
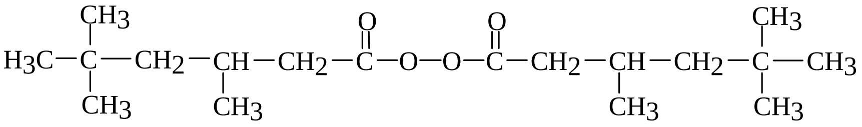
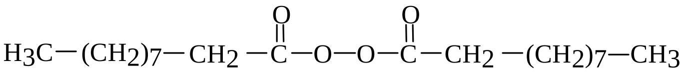
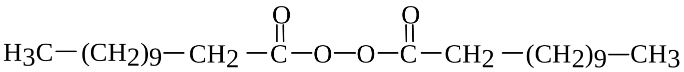
Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Peroxydicarbonates (continued)							
	tert-butyl peroxy isopropylcarbonate  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-O-\overset{O}{\overset{\Vert}{C}}-O-O-\overset{O}{\overset{\Vert}{C}}-O-\overset{CH_3}{\underset{CH_3}{C}}-H$	220.221	143.1	3.14E+15	429	389	370
	di-n-butyl peroxydicarbonate  $C_4H_9-O-\overset{O}{\overset{\Vert}{C}}-O-O-\overset{O}{\overset{\Vert}{C}}-O-C_4H_9$	234.248	125.9	9.44E+15	367	334	318
	di-sec-butyl peroxydicarbonate  $H_3C-CH_2-\overset{CH_3}{\underset{H}{C}}-O-\overset{O}{\overset{\Vert}{C}}-O-O-\overset{O}{\overset{\Vert}{C}}-O-\overset{CH_3}{\underset{H}{C}}-CH_2-CH_3$	234.248	125.0	7.11E+15	367	334	318
	di-cyclohexyl peroxydicarbonate 	286.324	125.1	8.26E+15	366	333	317
	bis(2-ethylhexyl) peroxydicarbonate  $C_4H_9-\underset{C_2H_5}{CH}-CH_2-O-\overset{O}{\overset{\Vert}{C}}-O-O-\overset{O}{\overset{\Vert}{C}}-O-CH_2-\underset{C_2H_5}{CH}-C_4H_9$	346.464	124.5	7.45E+15	365	332	316
	bis(4-tert-butyl cyclohexyl) peroxydicarbonate 	398.540	124.5	7.45E+15	365	332	316
	di-cetyl peroxydicarbonate  $H_3C-(CH_2)_{14}-CH_2-O-\overset{O}{\overset{\Vert}{C}}-O-O-\overset{O}{\overset{\Vert}{C}}-O-CH_2-(CH_2)_{14}-CH_3$	570.896	125.9	9.44E+15	367	334	318

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

continued

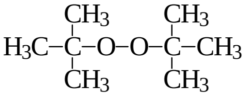
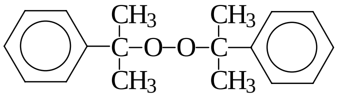
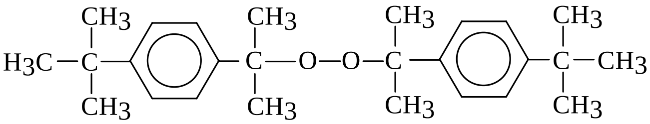
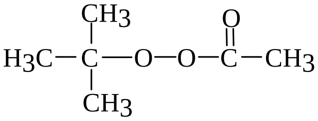
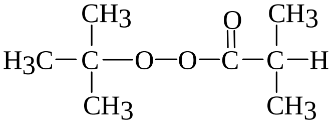
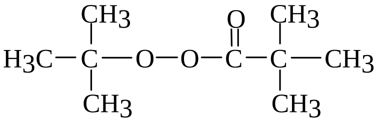
Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Acyl Peroxides							
	di-propionyl peroxide  $H_3C-CH_2-\overset{O}{\overset{\|}{C}}-O-O-\overset{O}{\overset{\|}{C}}-CH_2-CH_3$	146.142	126.6	7.35E+14	394	356	338
	di-benzoyl peroxide 	242.230	127.3	3.71E+14	403	364	345
	bis(2,4-dichlorobenzoyl) peroxide 	380.010	129.4	4.59E+15	384	349	332
	bis(ortho-methylbenzoyl) peroxide 	270.284	119.1	1.39E+14	387	349	330
	di-octanoyl peroxide  $H_3C-(CH_2)_5-CH_2-\overset{O}{\overset{\|}{C}}-O-O-\overset{O}{\overset{\|}{C}}-CH_2-(CH_2)_5-CH_3$	286.412	133.8	1.28E+16	388	353	336
	bis(3,5,5-trimethylhexanoyl) peroxide  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-CH_2-\underset{CH_3}{CH}-CH_2-\overset{O}{\overset{\|}{C}}-O-O-\overset{O}{\overset{\|}{C}}-CH_2-\underset{CH_3}{CH}-CH_2-\overset{CH_3}{\underset{CH_3}{C}}-CH_3$	314.466	124.4	6.43E+14	388	351	333
	di-decanoyl peroxide  $H_3C-(CH_2)_7-CH_2-\overset{O}{\overset{\|}{C}}-O-O-\overset{O}{\overset{\|}{C}}-CH_2-(CH_2)_7-CH_3$	342.520	127.4	1.44E+15	389	353	335
	di-lauroyl peroxide  $H_3C-(CH_2)_9-CH_2-\overset{O}{\overset{\|}{C}}-O-O-\overset{O}{\overset{\|}{C}}-CH_2-(CH_2)_9-CH_3$	398.628	127.4	1.44E+15	389	353	335

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

continued

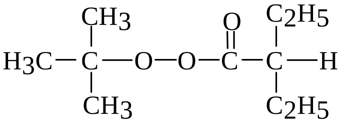
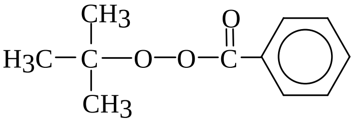
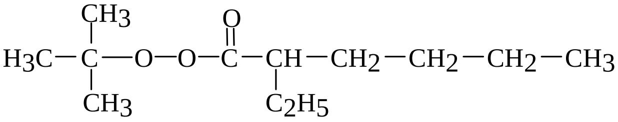
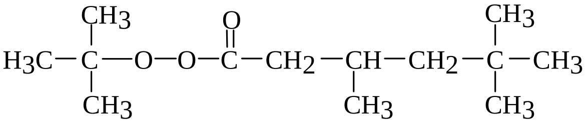
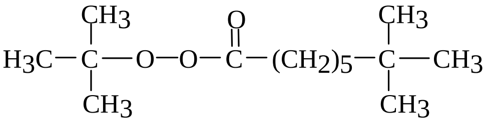
Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Alkyl Peroxides							
	di-tert-butyl peroxide  $H_3C-\underset{CH_3}{\overset{CH_3}{C}}-O-O-\underset{CH_3}{\overset{CH_3}{C}}-CH_3$	146.230	148.9	6.16E+14	465	421	399
	di-cumyl peroxide 	270.372	152.7	5.65E+15	451	410	390
	di-tert-butyl cumyl peroxide 	382.588	152.6	3.33E+15	457	415	394
Per-Esters							
	tert-butyl peroxyacetate  $H_3C-\underset{CH_3}{\overset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-CH_3$	132.159	144.3	2.23E+15	436	396	376
	tert-butyl peroxyisobutyrate  $H_3C-\underset{CH_3}{\overset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-\underset{CH_3}{\overset{CH_3}{C}}-H$	160.213	138.7	8.61E+15	406	369	351
	tert-butyl peroxypivalate  $H_3C-\underset{CH_3}{\overset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-\underset{CH_3}{\overset{CH_3}{C}}-CH_3$	174.240	117.8	1.11E+14	385	347	328

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

continued

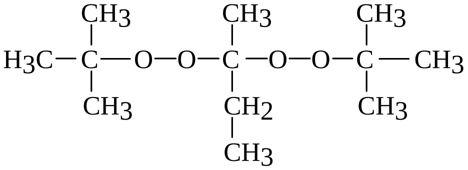
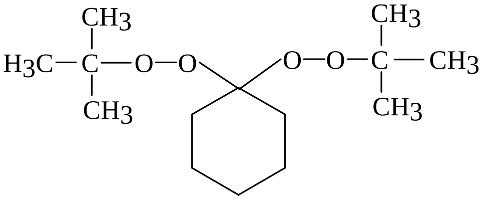
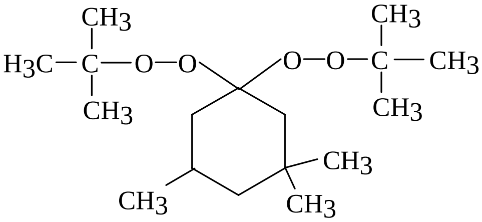
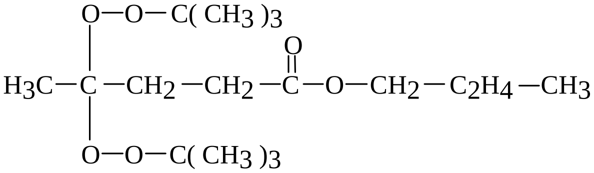
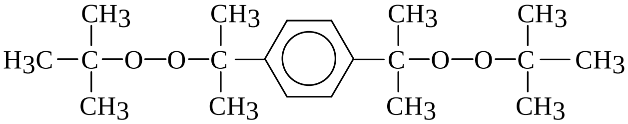
Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Per-Esters (continued)							
	tert-butyl peroxydiethylacetate  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-\overset{C_2H_5}{\underset{C_2H_5}{C}}-H$	188.267	126.7	2.07E+14	407	367	348
	tert-butyl peroxybenzoate 	194.230	138.8	2.96E+14	442	399	378
	tert-butyl peroxy-2-ethylhexanoate  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-\underset{C_2H_5}{CH}-CH_2-CH_2-CH_2-CH_3$	216.321	127.3	2.87E+14	406	366	347
	tert-butyl peroxy-3,5,5-trimethyl- hexanoate  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-CH_2-\underset{CH_3}{CH}-CH_2-\overset{CH_3}{\underset{CH_3}{C}}-CH_3$	230.348	143.0	2.38E+15	159	391	372
	tert-butyl peroxyneodecanoate  $H_3C-\overset{CH_3}{\underset{CH_3}{C}}-O-O-\overset{O}{\overset{\parallel}{C}}-(CH_2)_5-\overset{CH_3}{\underset{CH_3}{C}}-CH_3$	244.375	121.3	1.26E+15	372	337	320

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

continued

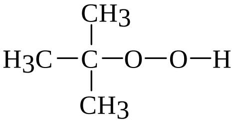
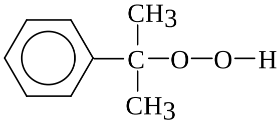
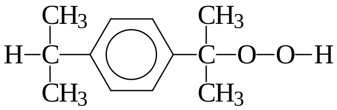
Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	Name / Structure	MW g/mol	E_a 1E6 J/kmol	A s^{-1}	T, $t_{1/2} = 60sec$ K	T, $t_{1/2} = 1hr$ K	T, $t_{1/2} = 10hrs$ K
Per-Ketals							
	2,2-di-tert-butyl peroxybutane  $H_3C-C(CH_3)_2-O-O-C(CH_3)(CH_2CH_3)-O-O-C(CH_3)_2-CH_3$	234.36	141.5	1.15E+15	435	394	374
	1,1-di-tert-butyl peroxycyclohexane 	260.374	143.1	3.14E+15	429	389	370
	1,1-di-tert-butyl peroxy-3,5,5- trimethylcyclohexane 	302.455	144.8	6.96E+15	426	387	368
	4,4-di-tert-butyl peroxy n-butyl valerate  $H_3C-C(O-O-C(CH_3)_3)_2-CH_2-CH_2-C(=O)-O-CH_2-C_2H_4-CH_3$	334.453	139.5	2.10E+14	448	404	383
	1,4-bis(tert-butyl peroxyisopropyl) benzene 	338.488	152.6	3.33E+15	457	415	394

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

continued

Table G.1 Initiator Decomposition Rate Parameters (cont.)

Formula	**Name / Structure**	**MW** **g/mol**	**E_a** **1E6 J/kmol**	**A** **s^{-1}**	**T, $t_{1/2} = 60sec$ K**	**T, $t_{1/2} = 1hr$ K**	**T, $t_{1/2} = 10hrs$ K**
Hydroperoxides							
	tert-butyl hydroperoxide 	90.122	142.8	1.23E+12	532	472	444
	cumene hydroperoxide 	152.193	113.0	9.05E+08	542	466	432
	di-isopropyl benzene monohydroperoxide 	194.274	115.4	3.51E+09	525	455	423

Source: Warson, H., Per-Compounds and Per-Salts in Polymer Processes, Solihull Chemical Services, England, 5-17 (1980).

H

FORTRAN UTILITIES

This appendix describes the input and output arguments for various Fortran utilities useful for writing user kinetic subroutines. For each utility a list of variables in the argument list is given along with their I/O status, their type, and a brief description. These utilities are available to you in addition to those documented in the *Aspen Plus User Models Reference Manual*.

The utilities documented in this appendix are:

- Component Attribute Handling Utilities
- Component Handling Utilities
- General Stream Handling Utilities
- Other Utilities

A list of utilities contained in each category follows:

Component Attribute Handling Utilities

Component Handling Utilities

General Stream Handling Utilities

Other Utilities

COMPONENT ATTRIBUTE HANDLING UTILITIES

CAELID

Utility Description

This utility finds a component attribute element ID given the attribute ID and the element number.

Argument List

```
SUBROUTINE SHS_CAELID ( IDCAT, IELEM, IDCAEL )
```

Variable	I/O	Type	Dimension	Description
IDCAT	I	INTEGER	2	Comp attr. ID in two integer words
IELEM	I	INTEGER	---	Comp attr. element no.
IDCAEL	O	INTEGER	2	Comp attr. element ID in two integer words

Calling Sequence in User Routine

```
INTEGER IDCAT(2), IELEM, IDCAEL(2)
DATA IDCAT / 'DPN ','    ' /
IELEM=1
.
.
.
CALL SHS_CAELID ( IDCAT, IELEM, IDCAEL )
```

CAID

Utility Description

This utility finds a component attribute ID given the component sequence number, the attribute type number, and the substream structure.

Argument List

```
SUBROUTINE SHS_CAID ( ISSCNC, NCSEQ, J, IDCAT )
```

Variable	I/O	Type	Dimension	Description
ISSCNC	I	INTEGER	---	Flag: 1 = conventional substream 2 = nonconventional substream
NCSEQ	I	INTEGER	---	Attributed component sequence number
J	I	INTEGER	---	Comp attr. type no.
IDCAT	O	INTEGER	2	Comp attr. ID in two integer words

Calling Sequence in User Routine

```
INTEGER ISSCNC, NCSEQ, J, IDCAT(2)
ISSCNC=1             !'Conventional substream'
J=1
.
.
.
CALL SHS_CAID ( ISSCNC, NCSEQ, J, IDCAT )
```

CAMIX

Utility Description

This is the component attribute mixing utility. It mixes the attributes from two inlet streams into an outlet stream.

Argument List

```
SUBROUTINE SHS_CAMIX ( IP, SS1, SS2, SSO )
```

Variable	I/O	Type	Dimension	Description
IP	I	I	---	Substream type 1 or 2 = conventional 3 = nonconventional
SS1	I	R	1	1st input substream
SS2	I	R	1	2nd input substream
SSO	O	R	1	Output substream

Calling Sequence in User Routine

```
INTEGER IP
REAL*8 SS1(1), SS2(1), SS0(1)
IP=1            !'Conventional substream'
.
.
.
CALL SHS_CAMIX ( IP, SS1, SS2, SSO )
```

CASPLT

Utility Description

This is the component attribute splitting utility. It calculates the attribute values in a product stream for a given feed stream.

Argument List

```
SUBROUTINE SHS_CASPLT ( FEED, PROD, NSUBS, IDXSUB, ITYPE )
```

Variable	I/O	Type	Dimension	Description
FEED	I	DBL	(1)	Combined feed stream
PROD	O	DBL	(1)	Given outlet stream
NSUBS	I	INT	---	Number of substreams
ITYPE	I	INT	NSUB	Substream type vector
IDXSUB	I	INT	NSUB	Substream index vector

Calling Sequence in User Routine

```
INTEGER NSUBS, ITYPE(1), IDXSUB(1)
REAL*8 FEED(1), PROD(1)
IDXSUB(1)=1             !'First substream'
ITYPE(1)=1              !'Conventional'
.
.
.
CALL SHS_CASPLT ( FEED, PROD, NSUBS, IDXSUB, ITYPE )
```

CASPSS

Utility Description

This is the component attribute splitting utility for substreams. It calculates the attribute values in a product substream for a given feed substream.

Argument List

```
SUBROUTINE SHS_CASPSS ( FEED, PROD, ITYPE )
```

Variable	I/O	Type	Dimension	Description
FEED	I	DBL	(1)	Feed substream
PROD	O	DBL	(1)	Outlet substream
ITYPE	I	INT	NSUB	Substream type

Calling Sequence in User Routine

```
INTEGER ITYPE(1)
REAL*8 FEED(1), PROD(1)
ITYPE(1)=1       !'Conventional'
.
.
.
CALL SHS_CASPSS ( FEED, PROD, ITYPE )
```

CAUPDT

Utility Description

This utility calculates class zero component attribute values in a product stream based on class 2 component attributes.

Argument List

```
SUBROUTINE SHS_CAUPDT ( STREAM, NSUBS, IDXSUB, IPHASE )
```

Variable	I/O	Type	Dimension	Description
STREAM	R*8	I	(1)	Stream vector
NSUBS	I	I	---	Number of substreams
IDXSUB	I	I	NSUBS	Substream index vector
IPHASE	I	I	NSUBS	Substream type vector 1= MIXED 2= CISOLID 3= NCSOLID

Calling Sequence in User Routine

```
INTEGER NSUBS, IDXSUB(1), IPHASE(1)
REAL*8 STREAM(1)
IPHASE(1)=1          !'First phase'
.
.
.
CALL SHS_CAUPDT ( STREAM, NSUBS, IDXSUB, IPHASE )
```

COPYCA

Utility Description

This utility copies all attribute values from one stream into another stream of the same type.

Argument List

```
SUBROUTINE SHS_COPYCA( LDIN, LD, LVRIN, LVR, IPHIN, IPH )
```

Variable	I/O	Type	Dimension	Description
LDIN	I	I	---	Location of the descriptor bead for the input stream
LD	I	I	---	Location of the descriptor bead for the output stream
LVRIN	I	I	---	Location of the real stream variable data for the input stream
LVR	I	I	---	Location of the real stream variable
IPHIN	I	I	---	Phase no. in the input stream
IPH	I	I	---	Phase no. in the output stream data for the output stream

Calling Sequence in User Routine

```
INTEGER IPHIN, IPH
INTEGE LDIN, LD, LVRIN, LVR              !'Use LOCATS utility model
                                           to obtain these values'
IPHIN=1
IPH=1
.
.
.
CALL SHS_COPYCA( LDIN, LD, LVRIN, LVR, IPHIN, IPH )
```

GETCRY

Utility Description

This utility returns the crystallinity of a list of components.

Argument List

```
SUBROUTINE POLY_GETCRY ( NCNC, NCP, IDX, CRY )
```

Variable	I/O	Type	Dimension	Description
NCNC	I	I	---	1 = conventional substream 2 = non-conventional substream
NCP	I	I	---	Number of components
IDX	I	I	NCP	Component index vector
CRY	O	I	NCP	Crystalline fraction

Calling Sequence in User Routine

```
INTEGER NCNC, NCP, IDXP, CRY(1)
NCNC=1             !'Conventional substream'
NCP=1
IDXP=2             !'Polymer is 2nd component'
.
.
.
CALL POLY_GETCRY( NCNC, NCP, IDX, CRY )
```

GETDPN

Utility Description

This utility returns the number average degree of polymerization.

Argument List

```
SUBROUTINE POLY_GETDPN ( NCNC, NCP, IDX, DPN )
```

Variable	I/O	Type	Dimension	Description
NCNC	I	I	---	1 = conventional substream 2 = non-conventional substream
NCP	I	I	---	Number of components
IDX	I	I	NCP	Component index vector
DPN	O	R	NCP	Degree of polymerization

Calling Sequence in User Routine

```
INTEGER NCNC, NCP, IDXP
REAL*8 DPN
NCNC=1             !'Conventional substream'
NCP=1
IDXP=2            !'Polymer is 2nd component'
.
.
.
CALL POLY_GETDPN( NCNC, NCP, IDX, DPN )
```

GETMWN

Utility Description

This utility calculates the true molecular weight of a polymer, from the degree of polymerization and the average segment molecular weight.

Argument List

```
SUBROUTINE POLY_GETMWN ( NCNC, NCP, IDX, XMWTRU )
```

Variable	I/O	Type	Dimension	Description
NCNC	I	INT	---	1 = conventional substream 2 = non-conventional substream
NCP	I	INT	---	Number of components
IDX	I	INT	NCP	Component index vector
XMWTRU	O	DBL	NCP	True number average molecular weight

Calling Sequence in User Routine

```
INTEGER NCNC, NCP, IDXP
REAL*8 XMWTRU(1)
NCNC=1             !'Conventional substream'
NCP=1
IDXP=2             !'Polymer is 2nd component'
.
.
.
CALL POLY_GETMWN( NCNC, NCP, IDX, XMWTRU )
```

GETMWW

Utility Description

This utility returns the weight-average molecular weight vector for all of the polymer components present. For standard components, the component molecular weight is returned.

Argument List

```
SUBROUTINE POLY_GETMWW ( NCNC, NCP, IDX, MWW )
```

Variable	I/O	Type	Dimension	Description
NCNC	I	INTEGER	---	1 = conventional substream 2 = non-conventional substream
NCP	I	INTEGER	---	Number of components present
IDX	I	INTEGER	NCP	Component index vector
MWW	O	DBL	NCP	Weight-average molecular weight

Calling Sequence in User Routine

```
REAL*8 MWW
INTEGER NCNC, NCP, IDXI
NCNC=1             !'Conventional substream'
NCP=1              !'Only one component (polymer)'
IDXI=2             !'Polymer is 2nd component'
.
.
.
CALL POLY_GETMWW( NCNC, NCP, IDXI, MWW )
```

LCAOFF

Utility Description

This utility finds the offset of a component attribute from the beginning of the substream given the structure of the substream, the component sequence number and the attribute type number.

Argument List

```
FUNCTION SHS_LCAOFF (ISSCNC, NCSEQ, J)
```

Variable	I/O	Type	Dimension	Description
ISSCNC	I	INTEGER	---	Flag: 1 = Conventional substream 2 = non conventional substream
NCSEQ	I	INTEGER	---	Attributed component sequence number
J	I	INTEGER	---	Comp. attribute type no.
LCAOFF	O	INTEGER	---	Attribute offset from substream

Calling Sequence in User Routine

```
INTEGER ISSCNC, NCSEQ, J, SHS_LCAOFF
ISSCNC=1       !'Conventional substream'
NCSEQ=2        !'Second attributed component'
J=1            !'First component attribute in the list for this component'
.
.
.
N = SHS_LCAOFF (ISSCNC, NCSEQ, J)
```

LCATT

Utility Description

This utility finds the offset of a component attribute from the substream given the structure of the substream, attributed component sequence number and the attribute ID.

Argument List

```
FUNCTION SHS_LCATT (ISSCNC, NCSEQ, IDCATT)
```

Variable	I/O	Type	Dimension	Description
ISSCNC	I	INTEGER	---	Flag: 1 = conventional substream 2 = non conventional substream
NCSEQ	I	INTEGER	---	Attributed component sequence number
IDCATT	I	INTEGER	2	Component attribute ID in two integer words
LCATT	O	INTEGER	---	Attribute offset

Calling Sequence in User Routine

```
INTEGER ISSCNC, NCSEQ, IAID(2), SHS_LCATT
DATA IAD / DPN ', / '    ' /
ISSCNC=1
NCSEQ=1
.
.
.
N = SHS_LCATT (ISSCNC, NCSEQ, IAID)
```

NCAVAR

Utility Description

This utility finds the number of elements in a component attribute given the attribute type index, the component index, and the substream type.

Argument List

```
FUNCTION SHS_NCAVAR (ISSCNC, NCSEQ, J)
```

Variable	I/O	Type	Dimension	Description
ISSCNC	I	INTEGER	---	Flag: 1 = conventional substream 2 = non conventional substream
NCSEQ	I	INTEGER	---	Attributed component sequence number
J	I	INTEGER	---	Comp attribute type no.
NCAVAR	O	INTEGER	---	Attribute length

Calling Sequence in User Routine

```
INTEGER ISSCNC, NCSEQ, J, SHS_NCAVAR
ISSCNC=1       !'Conventional substream'
NCSEQ=1
J=1
.
.
.
N = SHS_NCAVAR (ISSCNC, NCSEQ, J)
```

COMPONENT HANDLING UTILITIES

CPACK

Utility Description

This utility packs a list of conventional components into an output vector which contains components whose mole fraction is greater than a minimun value XMIN.

Argument List

```
SUBROUTINE SHS_CPACK (PHASE, NCP, IDX, X, FLOW)
```

Variable	I/O	Type	Dimension	Description
PHASE	I	REAL	NVCP	Vector of component flows
NCP	O	INTEGER	---	Number of components actually present
IDX	O	INTEGER	NCC	Index vector of components actually present
X	O	REAL	NCC	Mole fraction vector of components actually present
FLOW	O	REAL	---	Total molar flow of the phase

Calling Sequence in User Routine

```
INTEGER NCP, IDX(1)
REAL*8 X(1), FLOW, PHASE(1)
.
.
.
CALL SHS_CPACK (PHASE, NCP, IDX, X, FLOW)
```

IFCMNC

Utility Description

This utility is used to retrieve parameter values from the plex.

Argument List

```
FUNCTION DMS_IFCMNC ( NAME )
```

Variable	I/O	Type	Dimension	Description
NAME	I	CHARACTER*8	---	Parameter name as a character string

Calling Sequence in User Routine

```
C
    INTEGER TC, I
    .
    .
    .
    TC(I)=DMS_IFCMNC('TC')+I
```

ISCAT

Utility Description

This utility determines whether a component is a catalyst.

Argument List

```
FUNCTION ISCAT ( ICOMP )
```

Variable	I/O	Type	Dimension	Description
ICOMP	I	I	---	Component index
ISCAT	O	L	---	True for catalysts

Calling Sequence in User Routine

```
INTEGER ICOMP1
LOGICAL  PPUTL_ISCAT
ICOMP=2        !'Catalyst is 2nd component'
.
.
.
N = PPUTL_ISCAT( ICOMP )
```

ISOLIG

Utility Description

This utility determines whether a component is an oligomer.

Argument List

```
FUNCTION SHS_ISOLIG ( ICOMP )
```

Variable	I/O	Type	Dimension	Description
ICOMP	I	I	---	Component index
ISOLIG	O	L	---	True for oligomers

Calling Sequence in User Routine

```
INTEGER ICOMP
LOGICAL SHS_ISOLIG
ICOMP=3         !'Oligomer is 3rd component'
.
.
.
N = SHS_ISOLIG( ICOMP )
```

ISPOLY

Utility Description

This utility determines whether a component is a polymer.

Argument List

```
FUNCTION SHS_ISPOLY ( ICOMP )
```

Variable	I/O	Type	Dimension	Description
ICOMP	I	I	---	Component index
ISPOLY	O	L	---	True for polymers

Calling Sequence in User Routine

```
INTEGER ICOMP
LOGICAL SHS_ISPOLY
ICOMP=2         !'Polymer is 2nd component'
.
.
.
N = ISPOLY( ICOMP )
```

ISSEG

Utility Description

This utility determines whether a component is a segment.

Argument List

```
FUNCTION PPUTL_ISSEG ( ICOMP )
```

Variable	I/O	Type	Dimension	Description
ICOMP	I	I	---	Component index
ISSEG	O	L	---	True for segments

Calling Sequence in User Routine

```
INTEGER ICOMP
LOGICAL PPUTL_ISSEG
ICOMP=5         !'Segment is 5th component'
.
.
.
N = PPUTL_ISSEG( ICOMP )
```

SCPACK

Utility Description

This utility packs lists of conventional component flow rates from several phases into an output array which contains mole fractions of components that are present in at least one of the phases.

Argument List

```
SUBROUTINE SHS_SCPACK (NPHASE, LPHASE, NCP, IDX, X, FLOW)
```

Variable	I/O	Type	Dimension	Description
NPHASE	I	INTEGER	---	Number of phases to be packed
LPHASE	I	INTEGER	NPHASE	Vector of phase plex addresses
NCP	O	INTEGER	---	Number of components actually present
IDX	O	INTEGER	NCC	Index vector of components actually present
X	O	REAL	NCC, NPHASE	Mole fraction array of components actually present
FLOW	O	REAL	NPHASE	Vector of phase flow rates

Calling Sequence in User Routine

```
INTEGER NPHASE, LPHASE(1), NCP, IDX(1)
REAL*8 X(2), FLOW(2)
.
.
.
CALL SHS_SCPACK (NPHASE, LPHASE, NCP, IDX, X, FLOW)
```

XATOWT

Utility Description

This utility returns the true weight fractions for all of the components present.

Argument List

```
SUBROUTINE POLY_XATOWT ( X, NCP, IDX, WT )
```

Variable	I/O	Type	Dimension	Description
X	I	DBL	NCP	Apparent mole fraction vector
NCP	I	INTEGER	---	Number of components present
IDX	I	INTEGER	NCP	Component index vector
WT	O	DBL	NCP	True weight fraction vector

Calling Sequence in User Routine

```
C    Dimension of WT can be set high enough to avoid overwriting;
C    Alternatively, the proper work space can be assigned to the
C    array WT (preferred method)
C
     REAL*8 WT(10)
     .
     .
     .
     CALL POLY_XATOWT (X, NCP, IDX, WT)
```

XATOXT

Utility Description

This utility returns the true mole fractions for all of the components present.

Argument List

```
SUBROUTINE POLY_XATOXT ( NCP, IDX, XMW, X, XTRUE )
```

Variable	I/O	Type	Dimension	Description
NCP	I	INTEGER	---	Number of components present
IDX	I	INTEGER	NCP	Component index vector
XMW	I	DBL	NCC	Molecular weight vector
NCC	I	INTEGER	---	Number of components listed
X	I	IDNL	NCP	Apparent mole fraction vector
XTRUE	O	DBL	NCP	True mole fraction vector

Calling Sequence in User Routine

```
C    Dimension of XTRUE can be set high enough to avoid overwriting;
C    Alternatively, the proper work space can be assigned to the
C    array XTRUE (preferred method)
C
     REAL*8 XTRUE(10)
     .
     .
     .
     CALL POLY_XATOXT (NCP, IDX, XMW, X, XTRUE)
```

GENERAL STREAM HANDLING UTILITIES

IPTYPE

Utility Description

This utility returns the substream type number from the stream class descriptor bead.

Argument List

```
FUNCTION SHS_IPTYPE (LD, I)
```

Variable	I/O	Type	Dimension	Description
LD	I	INTEGER	---	Address of stream class descriptor bead
I	I	INTEGER	---	Substream number

Calling Sequence in User Routine

```
INTEGER I, SHS_IPTYPE
INTEGR LD          !'Use LOCATS utility model to obtain LD'
I=1                !'Substream 1'
.
.
.
N =  SHS_IPTYPE (LD, I)
```

LOCATS

Utility Description

This utility finds the integer and real plex addresses of a stream, the stream class bead location and bead number, given the stream bead number.

Argument List

```
SUBROUTINE SHS_LOCATS (NB, LVI, LVR, LD, NBD)
```

Variable	I/O	Type	Dimension	Description
NB	I	INTEGER	---	Stream bead number
LVI	O	INTEGER	---	Integer plex location of the stream bead
LVR	O	INTEGER	---	Real plex location of the real portion of the bead
LD	O	INTEGER	---	Integer plex location of the descriptor bead
NBD	O	INTEGER	---	Descriptor bead number
LOFFDB	---	INTEGER	---	Offset to the descriptor bead number in the stream bead

Calling Sequence in User Routine

```
INTEGER LVI, LVR, LD, NBD, LOFFDB
INTEGER NB         !'NB is obtained from the argument list'
.
.
.
CALL SHS_LOCATS (NB, LVI, LVR, LD, NBD)
```

LPHASE

Utility Description

This utility finds the offset of a substream from the beginning of a stream structure.

Argument List

```
FUNCTION SHS_LPHASE (LD, I)
```

Variable	I/O	Type	Dimension	Description
LD	I	INTEGER	---	Address of stream class descriptor bead
LPHASE	O	INTEGER	---	Offset of substream I in the stream structure
I	I	INTEGER	---	Substream number

Calling Sequence in User Routine

```
INTEGER I, SHS_LPHASE
INTEGER LD         !'Use LOCATS utility to obtain LD'
I=1
.
.
.
N = SHS_LPHASE (LD, I)
```

NPHASE

Utility Description

This utility finds the number of substreams from the stream class descriptor bead.

Argument List

```
FUNCTION SHS_NPHASE (LD)
```

Variable	I/O	Type	Dimension	Description
LD	I	INTEGER	---	Address of stream
NPHASE	O	INTEGER	---	No. of substreams

Calling Sequence in User Routine

```
INTEGER SHS_NPHASE
INTEGER         !'Use LOCATS utility model to obtain LD'
.
.
.
N = SHS_NPHASE (LD)
```

NSVAR

Utility Description

This utility returns the number of stream variables.

Argument List

```
FUNCTION SHS_NSVAR (LD)
```

Variable	I/O	Type	Dimension	Description
LD	I	INTEGER	---	Address of stream class descriptor bead
NSVAR	O	INTEGER	---	Length of stream variable

Calling Sequence in User Routine

```
INTEGER SHS_NSVAR
INTEGER LD        !'Use LOCATS utility model to obtain LD'
.
.
.
N = SHS_NSVAR (LD)
```

SSCOPY

Utility Description

This utility copies substream information from one stream to another.

Argument List

```
SUBROUTINE  SHS_SSCOPY( LD, S1, S2, IDX, I)
```

Variable	I/O	Type	Dimension	Description
LD	I	I	---	Locator of descriptor bead
S1	I	R	(1)	Stream vector to copy from
S2	O	R	(1)	Stream vector to copy into
IDX	I	I	---	Location of substream within stream vector
I	I	I	---	Substream number

Calling Sequence in User Routine

```
INTEGER LD   !'Use LOCATS utility model to obtain LD'
INTEGER IDX(1), I
REAL*8 S(1), S2(1)
I=1         !'Substream 1'
.
.
.
CALL  SHS_SSCOPY( LD, S1, S2, IDX, I)
```

OTHER UTILITIES

VOLL

Utility Description

This utility calculates the mixture molar volume of liquid.

Argument List

```
SUBROUTINE  PPMON_VOLL ( T, P, X, N, IDX, NBOPST, KDIAG, KKV, V, DV, KER )
```

Variable	I/O	Type	Dimension	Description
T	I	REAL*8	---	Temperature (K)
P	I	REAL*8	---	Pressure (N / m^2)
N	I	INTEGER	---	Number of components present
IDX	I	INTEGER	N	Component index vector
X	I	REAL*8	N	Liquid mole fraction vector
NBOPST	I	INTEGER	6	Physical property method vector
KDIAG	I	INTEGER	---	Diagnostic level code
KKV	I	INTEGER	---	Mole volume calculation code
V	O	REAL*8	---	Mixture molar volume $(m^3 / kgmole)$
DV	O	REAL*8	---	Partial derivatives of mixture molar volume with respect to temperature $(m^3 / kgmole - K)$
KER	O	INTEGER	---	Error return code (=0 if an error or warning condition occurred in any physical property model; =0 otherwise)

Calling Sequence in User Routine

```
INTEGER N, IDX(N), NBOPST(6)   !'These variables are obtained from
                                 the argument list'
INTEGER KDIAG, KKV
REAL*8 T, P, X(N)              !'X is obtained from the argument list'
KDIAG=4
KKV=1
.
.
.
CALL  PPMON_VOLL (T, P, X, N, IDX, NBOPST, KDIAG, KKV, V, DV, KER)
```

INPUT LANGUAGE REFERENCE

This section describes the input language for:

SPECIFYING COMPONENTS

Naming Components

The input language used to name components is described below.

Input Language for Components

```
COMPONENTS    cid    [cname]    [outid]    / ...
```

Input Language Description for Components

COMPONENTS	*cid*	Component ID. Used to refer to the component in all subsequent input and is also used to identify the component in the simulation report. Aspen Plus input language conventions and naming guidelines apply to this keyword.
	cname	The databank name or alias used for that component. Refer to the documentation for the desired databank to find out the correct databank name or alias for the desired component. Place an asterisk (*) in the cname position if you do not wish to retrieve the component from the databank. Note that in this case you are required to provide all necessary physical property parameters.
	outid	Eight-character name used for the component in reports. (Default=*cid*)

Input Language Example for Components

```
DATABANKS PURECOMP / POLYMER / SEGMENT
COMPONENTS
  INI1   STYRENE          INIT      /  ;  INITIATOR
  STY    STYRENE          STYRENE   /  ;  MONOMER
  CAN    ACRYLONITRILE    CAN       /  ;  MONOMER
  XYLENE P-XYLENE         XYLENE    /  ;  SOLVENT
  STYSEG STYRENE-R        STY-SEG   /  ;  STYRENE SEGMENT
  ACNSEG ACRYLONITRILE-R  ACN-SEG   /  ;  ACN SEGMENT
  SAN    SAN              SAN          ;  COPOLYMER
```

Specifying Component Characterization Inputs

A POLYMERS paragraph is used to define polymers, their segments, oligomers, and heterogeneous catalysts, if any, involved in the polymerization. This paragraph is also used to define the polymer and catalyst component attributes desired in the simulation. Only the names of the attributes need to be specified in the POLYMERS paragraph. Initial values for the component attributes may be entered for the polymer and catalyst components in each stream via the STREAM paragraph. The input language for the POLYMERS paragraph is given below.

Input Language for Polymers, Oligomers and Catalysts

```
POLYMERS
   PARAM           kwd=value
   SEGMENTS        seg-id  seg-type / ...
   OLIGOMERS       olig-id  seg-id  number / ...
   POLYMERS        poly-id / ...
   CATALYSTS       cat-id  mol-site / ...
   ATTRIBUTES      comp-id  attr-list / ...
   DISTRIBUTION    polyid  disttype  NPOINTS=value  FUNCLOG=YES/NO
                   UPPER=value
```

Input Language Description for Polymers Oligomers and Catalysts

PARAM	Used to enter special parameters. Keywords are as follows.	
	NSITE	Number of catalyst site types
SEGMENTS	Used to specify all the segments used in the simulation. The information entered through this keyword is used by the system to pass segment property information.	
	seg-id	Name of the segment (must be a valid component ID)
	seg-type	Segment type. This information is used to differentiate segment types. The options are END, REPEAT, BRANCH3, or BRANCH4. The default value is REPEAT
OLIGOMERS	Used to specify the structure of oligomers present in the simulation.	
	olig-id	Oligomer component ID
	seg-id	ID for segment contained in that oligomer. All the segment names must be valid component IDs (Optional)
	number	Number of this segment in the oligomer (Default=1)
POLYMERS	Used to identify all polymers present in the simulation.	
	poly-id	Name of the polymer (must be a valid component ID)

CATALYSTS Used to identify all the heterogeneous polymerization catalysts present in the simulation and to specify the moles of catalytic sites per mole of catalyst.

cat-id	Catalyst component ID
mol-site	Moles of catalytic sites per unit mass of that catalyst

ATTRIBUTES Used to specify all the polymer/catalyst component attributes desired for each polymer/catalyst in the simulation. Only the attribute names need to be specified here. Values for the component attributes are entered in the COMP-ATTR sentence of the STREAM paragraph.

comp-id	Polymer or catalyst component ID
attr-list	List of component attributes. The component attributes specific to polymers listed in Tables 2.5-2.10 while those for catalysts are listed in Table 2.12

DISTRIBUTION Used to request polymer property distribution plots.

polyid	Polymer ID
disttype	Distribution type
NPOINTS	Number of points
FUNCLOG	Calculate distribution as rW(r) vs. r on a log scale. Default is NO
upper	Upper limit

Since component attributes represent a significant feature in Polymers Plus, a complete subsection has been devoted to their use in the simulator. For more detailed information regarding component attributes see Section 2.2.

Input Language Example for Polymers, Oligomers and Catalysts

```
POLYMERS
  POLYMERS   SAN                   ; DEFINE SEGMENTS IN POLYSTYRENE
  SEGMENTS   STYSEG REPEAT/
             ACNSEG REPEAT         ; DEFINE TYPE OF SEGMENTS PRESENT
; DEFINE ATTRIBUTES FOR POLYMERS
  ATTRIBUTES SAN  DPN DPW PDI MWN MWW ZMOM FMOM SMOM SFLOW SFRAC    &
                 LDPN LZMOM LFMOM LSFLOW LSFRAC LEFLOW LEFRAC LPFRAC
  DISTRIBUTION  PS  CHAIN-SIZE  NPOINTS=100  UPPER=9999
```

SPECIFYING COMPONENT ATTRIBUTES

Specifying Characterization Attributes

See Specifying Component Characterization Inputs.

Specifying Conventional Component Attributes

To assign user component attributes to a conventional component use the ATTR-COMPS paragraph as follows:

Input Language for Catalyst Component Attributes

```
ATTR-COMPS   comp-id attr-list  CLASS=CV  /  ...
```

Input Language Description for Catalyst Component Attributes

comp-id	Standard component ID.
attr-list	List of attributes. Valid attributes were given in Table 2.14.

Initializing Attributes in Streams

Following is the input language used to enter attribute values in streams.

Input Language for Material Streams

```
STREAM    sid
   SUBSTREAM    ssid    keyword=value
   basis-FLOW    cid    flow  / . . .
   basis-FRAC    cid    frac  / . . .
   COMP-ATTR    cname    cattrname    (value-list)  / . . .
```

Keywords:

TEMP PRES basis-FLOW

Optional Keywords:

NPHASE PHASE

Input Language Description for Material Streams

SUBSTREAM	Used to enter state and flash specifications for substreams.	
	ssid	Substream ID
	TEMP	Temperature
	PRES	Pressure
	basis-FLOW	Flow rate on a MOLE, MASS, or VOLUME basis
	NPHASE	Number of phases
	PHASE	Used to specify the phase when NPHASE=1 PHASE=V (vapor), L (liquid), or S (solid)
basis-FLOW	Used to enter component flows.	
	cid	Component ID
	flow	Component mole or mass flow
basis-FRAC	Used to enter component fractions.	
	cid	Component ID
	frac	Component mole or mass fraction
COMP-ATTR	Used to enter component attribute values.	
	cname	Component name
	cattrname	Component attribute name. For polymer attributes, values must be entered for at least SFRAC or SFLOW, and DPN or both ZMOM and FMOM

value-list	List of values for each element in the attribute. Use "*" to skip entries

Input Language Example for Material Streams

```
STREAM FEED
  SUBSTREAM MIXED TEMP=70 PRES=1
  MASS-FLOW STY 13.5 /ACN 7.27 /XYLENE 79 /SAN 0.1E-5/INI1 0.23
  COMP-ATTR SAN    DPN (3000)              /
                   DPW (6000)              /
                   PDI (2)                 /
                   MWN (312450)            /
                   MWW (624900)            /
                   ZMOM (0.39E-10)         /
                   FMOM (1.17E-7)          /
                   SMOM (7.02E-4)          /
                   SFLOW (0.55E-7 0.55E-7)      /
                   SFRAC (0.5 0.5)              /
                   LSFLOW (0.  0.)         /
                   LEFLOW (0.  0.)
```

REQUESTING DISTRIBUTION CALCULATIONS

See Specifying Component Characterization Inputs.

CALCULATING END USE PROPERTIES

Input Language for Prop-Set

PROP-SET *propsetid* *propname-list* *keyword=value*

Optional Keywords:

COMPS PHASE UNITS TEMP PRES

Input Language Description for Prop-Set

Use the Prop-Set paragraph to define a property set. A property set is a collection of thermodynamic, transport, and other properties. Each property set you define is identified by an ID you supply.

propsetid	Property set ID.	
propname-list	List of property names. (See *Aspen Plus Physical Property Data Reference Manual*.)	
COMPS	List of component IDs (applies to all properties listed in *Aspen Plus Physical Property Data Reference Manual*). (Default=all components actually present when the property is calculated.)	
PHASE	PHASE=V	Vapor
	PHASE=L	Total liquid
	PHASE=L1	First-liquid
	PHASE=L2	Second-liquid
	PHASE=T	Total mixture
	PHASE=S	Solid
	Phase compositions are determined at stream conditions. (Default=T, if listed as a valid phase for the property in *Aspen Plus Physical Property Data Reference Manual*; otherwise no default.)	
UNITS	Units options selected for the units keywords that are listed for the property in *Aspen Plus Physical Property Data Reference Manual*. (Default=IN-UNITS if Prop-Set is specified for design specifications, Fortran blocks, optimization paragraphs and constraint paragraphs. Default=OUT-UNITS if Prop-Set is specified for reports. If a property has mole, mass, or flow units, the default will be mole units.)	

TEMP	Temperatures for property calculations. (Default=stream temperature. For VVSTD and VVSTDMX, Default=25°C.)
PRES	Pressures for property calculations. (Default=stream pressure. For VVSTD and VVSTDMX, Default=1 atm.)

Input Language for USER-PROPERTY

```
USER-PROPERTY   userpropid   propname-list   keyword=value
```

Keyword:

SUBROUTINE

Optional Keywords:

FLASH UNIT-TYPE UNIT-LABEL COMP-DEP LVPCT-DEP CURVE-PROP DEFAULT-PROP BLEND-METHOD BLEND-OPT EXTRAPOLATE

Input Language Description for USER-PROPERTY

Use the USER-PROPERTY paragraph to define the property. This property can be referenced in the Prop-Set paragraph in the same way as built-in properties. You must supply a Fortran subroutine to calculate the value of the user Prop-Set properties.

userpropid	User property set ID. This property must be different from built-in properties. (See *Aspen Plus Physical Property Data Reference Manual*.)
SUBROUTINE	Name of user-supplied subroutine for calculating the property. For details on writing the user-supplied subroutine, see *Aspen Plus User Models Reference Manual*.

FLASH	FLASH=NO	Does not flash the stream before the user-supplied subroutine is called (Default)
	FLASH=NOCOMPOSITE	Does not flash the stream for total stream properties (When PHASE=T in the Prop-Set paragraph), but flashes for any other phase specification
	FLASH=YES	Always flashes stream before the user-supplied subroutine is called
UNIT-TYPE	Units keyword for the property. If not entered, unit conversion is not performed on property values returned from the user-supplied subroutine.	
UNIT-LABEL	Unit label for the property printed in the report. A unit label is used only when unit conversion is performed by the user-supplied subroutine (that is, when UNIT-TYPE is not given).	
COMP-DEP	COMP-DEP=YES	Property is component property
	COMP-DEP=NO	Property is a mixture property (Default)

SPECIFYING PHYSICAL PROPERTY INPUTS

Specifying Property Methods

To input language used to specify property methods is described below.

Input Language for Physical Property Methods

```
PROPERTIES    opsetname    keyword=value  /
    opsetname    [sectionid-list]    keyword=value  /  . . .
```

Optional keywords:

FREE-WATER   SOLU-WATER   HENRY-COMPS

```
HENRY-COMPS    henryid    cid-list
```

Input Language Description for Property Methods

The PROPERTIES paragraph is used to specify the property method(s) to be used in your simulation. In this paragraph properties may be specified for the entire flowsheet, for a flowsheet section, or for an individual unit operation block. Depending on the component system used, additional information may be required such as Henry's law information, water solubility correlation, free-water phase properties. The input language for specifying property methods is as follows.

opsetname	Primary property method name (See Table 3.2).	
sectionid-list	List of flowsheet section IDs.	
FREE-WATER	Free water phase property method name (Default=STEAM-TA).	
SOLU-WATER	Method for calculating the K-value of water in the organic phase.	
	SOLU-WATER=0	Water solubility correlation is used, vapor phase fugacity for water calculated by free water phase property method
	SOLU-WATER=1	Water solubility correlation is used, vapor phase fugacity for water calculated by primary property method
	SOLU-WATER=2	Water solubility correlation is used with a correction for unsaturated systems, vapor phase fugacity for water calculated by primary property method
	SOLU-WATER=3	Primary property method is used. This method is not recommended for water-hydrocarbon systems unless water-hydrocarbon interaction parameters are available. (Default)
HENRY-COMPS	Henry's constant component list ID.	

The HENRY-COMPS paragraph identifies lists of components for which Henry's law and infinite dilution normalization are used. There may be any number of HENRY-COMPS paragraphs since different lists may apply to different blocks or sections of the flowsheet.

henryid	Henry's constant component list ID
cid-list	List of component IDs

Input Language Example for Property Methods

```
HENRY-COMPS  HC  INI1
PROPERTIES  POLYNRTL  HENRY-COMPS=HC
```

Specifying Property Data

To input language used to specify property data is described below.

Input Language for Property Data

```
PROP-DATA
   PROP-LIST    paramname    [setno]  /  . . .
   PVAL    cid    value-list  /  value-list  /  . . .
   PROP-LIST    paramname    [setno]  /  . . .
   BPVAL    cid1    cid2    value-list  /  value-list  /  . . .
   COMP-LIST    cid-list
   CVAL    paramname    setno    1    value-list
   COMP-LIST    cid2-list
   BCVAL    paramname    setno    1    cid1    value-list  /
                         1    cid1    value-list  /  . . .
```

Physical property models require data in order to calculate property values. Once you have selected the property method(s) to be used in your simulation, you must determine the parameter requirements for the models contained in the property method(s), and ensure that they are available in the databanks. If the model parameters are not available from the databanks, you may estimate them using the Property Constant Estimation System, or enter them using the PROP-DATA or TAB-POLY paragraphs. The input language for the PROP-DATA paragraphs is as follows. Note that only the general structure is given, the format for the input parameters required by polymer specific models is given in the Chapter 3 subsection where the model is described.

PROP-LIST	Used to enter parameter names and data set numbers.
PVAL	Used to enter the PROP-LIST parameter values.
BPVAL	Used to enter the PROP-LIST binary parameter values.
COMP-LIST	Used to enter component IDs.
CVAL	Used to enter the COMP-LIST parameter values.
BCVAL	Used to enter the COMP-LIST binary parameter values.

paramname	Parameter name

setno	Data set number. For CVAL and BCVAL the data set number must be entered. For setno > 1, the data set number must also be specified in a new property method defined using the PROP-REPLACE paragraph. (For PROP-LIST, Default=1)
cid	Component ID
cid1	Component ID of first component of binary pair
cid2	Component ID of second component of binary pair
value-list	List of parameter values. For PROP-LIST, enter one value for each element of the property; for COMP-LIST, enter one value for each component in the cid-list.
cid-list	List of component ID

Input Language Example for Property Data

```
PROP-DATA
  IN-UNITS SI
  PROP-LIST       PLXANT                       / TB
  PVAL HOPOLY     -40.0 0 0 0 0 0 0 0 1D3      / 2000.0
  PVAL COPOLY     -40.0 0 0 0 0 0 0 0 1D3      / 2000.0

PROP-DATA
  IN-UNITS SI
  PROP-LIST       MW
  PVAL HOPOLY       1.0
  PVAL COPOLY       1.0
  PVAL ABSEG      192.17
  PVAL ASEG        76.09
  PVAL BSEG       116.08

PROP-DATA
  IN-UNITS SI
  PROP-LIST        DHCONM    /   DHSUB     /  TMVK  / TGVK
  PVAL HOPOLY   -3.64261D4  /  8.84633D4  /  1.0    / 0.0
  PVAL COPOLY   -3.64261D4  /  8.84633D4  /  1.0    / 0.0

PROP-DATA
  IN-UNITS SI
  PROP-LIST GMRENB      /  GMRENC
  BPVAL MCH ASEG -92.0  /    0.2
  BPVAL ASEG MCH 430.0  /    0.2
```

Estimating Property Parameters

To input language used to estimate property parameters is described below.

Input Language for Property Parameter Estimation

```
ESTIMATE [option]

STRUCTURES
  method    SEG-id    groupno    nooccur / groupno    nooccur / . . . . . .
```

Input Language Description for Property Parameter Estimation

The main keywords for specifying property parameter estimation inputs are the ESTIMATE and the STRUCTURES paragraphs. A brief description of the input language for these paragraphs follows. For more detailed information please refer to the *Aspen Plus Physical Property Data Reference Manual.*

option	Option=ALL	Estimate all missing parameters (default)
method	Polymer property estimation method name	
SEG-id	Segment ID defined in the component list	
groupno	Functional group number (group ID taken from Appendix C)	
nooccur	Number of occurrences of the group	

Input Language Example for Property Parameter Estimation

```
ESTIMATE ALL

STRUCTURES
 VANKREV  ABSEG   115 1                    ;-(C6H4)-
 VANKREV  BSEG             151 2 / 100 2   ;       -COO-CH2-CH2-COO-
 VANKREV  ABSEG   115 1  / 151 2 / 100 2   ;-(C6H4)-COO-CH2-CH2-COO-
```

SPECIFYING STEP-GROWTH POLYMERIZATION KINETICS

The input language for the STEP-GROWTH REACTIONS paragraph is described below.

Input Language

```
REACTIONS    rxnid    STEP-GROWTH
   DESCRIPTION    '...'
   REPORT  REPORT=yes/no RXN-SUMMARY=yes/no  RXN-DETAILS=yes/noI
   STOIC     reactionno     compid     coeff  /  ...
   RATE-CON  reactionno  pre-exp  act-energy  [T-exp]  [T-ref] [USER-RC=number]
   POWLAW-EXP     reactionno     compid     exponent  /
   SPECIES POLYMER=polymerid    OLIGOMER=oligomer-list
   REAC-GRP     groupid     type  /…
   SPEC-GROUP     compid     groupid     number  /  groupid     number  /  ...
   RXN-SET rxn-setno
        [A-NUCL-SPEC=compid]    [A-ELEC-GRP=groupid]  &
        [V-ELEC-SPEC=compid ]    [V-NUCL-GRP=groupid]  &
        [V-NUCL-SPEC=compid]    [V-ELEC-GRP=groupid]   &
        RC-SETS=rc-setno-list
   SG-RATE-CON rc-setno
        [CAT-SPEC=compid]     [CAT-GRP=groupid]  &
        sgpre-exp     [sgact-energy]  [sgt-exp]  [sgt-ref]  [USER-RC=number]
   SUBROUTINE KINETICS=kinname   RATECON=rcname  MASSTRANS=mtname
   USER-VECS   NINTK=nintk   NREALK=nrealk   NINTRC=nintrc &
        NREALRC=nrealc   NINTMT=nintmt   NREALMT=nrealmt &
        NIWORK=niwork   NWORK=nwork NURC=nurc
   INTK   value-list
   REALK   value-list
   INTRC    value-list
   REALRC   value-list
   INTMT   value-list
   REALMT   value-list
   INCL-COMPS  compid-list
```

The keywords for specifying rate constant parameters for the built-in reactions, and for specifying user reactions are described below.

Input Language Description

rxnid	Unique paragraph ID.	
DESCRIPTION	Up to 64 characters between double quotes.	
REPORT	Reaction report options- controls writing of reaction report in .REP file.	
	REPORT=*YES*	Print reaction report
	REPORT=*NO*	Do not print reaction report
	RXN-SUMMARY=*YES*	Print stoichiometry for each model-generated and user-specified reaction. (Default).
	RXN-SUMMARY=*NO*	Do not print this summary.
	RXN-DETAILS=*YES*	Print stoichiometry, rate constants, and probability factors for each model-generated and user-specified reaction.
	RXN-DETAILS=*NO*	Do not print this detailed summary.
STOIC	Used to specify stoichiometry for user reactions.	
	Reactionno	Reaction number
	compid	Component ID
	coeff	Stoichiometric coefficient (positive for products, negative for reactants)
RATE-CON	Used to specify rate constants for user reactions.	
	Reactionno	Reaction number
	pre-exp	Pre-exponential factor in inverse-time units
	act-energy	Activation energy in mole enthalpy units
	T-exp	Temperature exponent
	T-ref	Reference temperature
	number	User rate constant flag
POWLAW-EXP	Used to specify power-law exponents for user reactions.	
	reactionno	Reaction number
	compid	Component ID

exponent	Power law exponent

SPECIES Used to specify key components involved in the reactions.

polymerid	Component ID for polymer product
oligomer-list	List of oligomers to be tracked

REAC-GRP Used to identify the names and types of reacting functional groups participating in the reaction network.

groupid	Functional group ID
type	Functional group type
EE-GRP	Electrophilic repeat unit
NN-GRP	Nucleophilic repeat unit
EN-GRP	Mixed electrophilic/nucleophilic repeat unit
E-GRP	Electrophilic leaving group
N-GRP	Nucleophilic leaving group
EX-GRP	Electrophilic modifier (end cap)
NX-GRP	Nucleophilic modifier (end cap)

SPEC-GROUP Used to characterize the reacting functional group composition of the components (segments and monomers) participating in the step-growth reaction network.

compid	Component ID
groupid	Reactive functional group ID
number	Number of occurrences of group in species

SG-RATE-CON Used to specify rate constants for model-generated step-growth reactions and to specify which catalyst they apply to (if any).

setno	Rate constant set number
CAT-SPEC=*compid*	Component ID of catalyst species
CAT-GRP=*groupid*	Group ID of catalyst group
USER-RC=*number*	User rate expression flag
sgpre-exp	Pre-exponential factor in inverse-time units
sgact-energy	Activation energy in mole-enthalpy units

sgt-exp	Temperature exponent
sgt-ref	Reference temperature in temperature units

RXN-SET Used to assign sets of rate constants to model-generated reactions.

A-NUCL-SPEC= *compid*	Component ID of reactant which acts as the attacking nucleophile
A-ELEC-GRP= *groupid*	Group ID of electrophilic leaving group in attacking nucleophilic reactant
V-ELEC-SPEC= *compid*	Component ID of reactant which acts as the nucleophile. When reactions occur inside polymer molecules, this may be a segment.
V-ELEC-GRP= *groupid*	Group ID of electrophilic group in victim species (attached to V-NUCL-GRP)
V-NUCL-SPEC= *compid*	Component ID of nucleophilic reactant attached to the victim electrophilic reactant at the reacting site
V-NUCL-GRP= *groupid*	Group ID of nucleophilic group in victim species (attached to V-ELEC-GRP)
RC-SETS = *rcsetno-list*	List of rate constants (from SG-RATE-CON) which apply to the set of reactions identified by the previous keywords

SUBROUTINE Used to provide the names of user-supplied Fortran subroutines. The subroutine argument lists are documented in Tables 4.38-4.40.

KINETICS=*kinname*	User kinetic subroutine name
RATECON=*rcname*	User rate constant subroutine name
MASSTRAN= *mtname*	User concentration basis / mass-transfer subroutine name

USER-VECS Used to specify the size of vectors for user subroutines.

NINTK=*nintk*	Length of integer array for kinetics
NREALK=*nrealk*	Length of real array for kinetics
NINTRC=*nintrc*	Length of integer array for rate constants
NREALRC=*nrealrc*	Length of real array for rate constants
NINTMT=*nintmt*	Length of integer array for mass transfer

	NREALMT=*nrealmt*	Length of real array for mass transfer
	NIWORK=*niwork*	Total length of integer workspace
	NWORK=*nwork*	Total length of real workspace
INTK	Used to enter integer parameter for kinetics.	
REALK	Used to enter real parameters for kinetics.	
INTRC	Used to enter integer parameters for rate constants.	
REALRC	Used to enter real parameters for rate constants.	
INTMT	Used to enter integer parameters for mass transfer.	
REALMT	Used to enter real parameters for mass transfer.	
INCL-COMPS	Used to list components which participate in reactions in the user kinetics model, but which do not appear in model-generated or user-specified reactions.	
	Compid-list	List of additional components to include in the mass-balance calculations

Input Language Example

```
REACTIONS NYLON STEP-GROWTH
    DESCRIPTION “NYLON-6 KINETICS:  SIMPLE MODEL WITHOUT CYCLICS”
    REPORT RXN-DETAILS=YES
    SPECIES POLYMER=NYLON6
    REAC-GROUP TNH2 E-GRP / TCOOH N-GRP / BCAP EN-GRP
    SPECIES-GRP T-NH2 TNH2 1 / T-NH2 BCAP 1 / T-COOH TCOOH 1 /  &
                T-COOH BCAP 1 / ACA TNH2 1 / ACA TCOOH 1 /  &
                ACA BCAP 1 / B-ACA BCAP 1 / H2O TNH2 1 / H2O TCOOH 1
    SG-RATE-CON 1                TREF=260 PRE-EXP= 5.461 ACT-
ENERGY=23.271
    SG-RATE-CON 2 CAT-SPEC=ACA    TREF=260 PRE-EXP=40.678 ACT-
ENERGY=20.670
    SG-RATE-CON 3 CAT-SPEC=T-COOH TREF=260 PRE-EXP=40.678 ACT-
ENERGY=20.670
    SG-RATE-CON 4                TREF=260 PRE-EXP=0.0124 ACT-
ENERGY=29.217
    SG-RATE-CON 5 CAT-SPEC=ACA    TREF=260 PRE-EXP=0.0924 ACT-
ENERGY=26.616
    SG-RATE-CON 6 CAT-SPEC=T-COOH TREF=260 PRE-EXP=0.0924 ACT-
ENERGY=26.616
    RXN-SET 1 ELECTRO-GRP=TNH2 NUCLEO-GRP=TCOOH RC-SETS= 1 2 3
    RXN-SET 2 NUCLEOPHILE=H2O                   RC-SETS= 4 5 6
```

Input Language Example (cont.)

```
STOIC 1 CL -1.0 / H2O -1.0 / ACA 1.0
STOIC 2 CL -1.0 / H2O -1.0 / ACA 1.0
STOIC 3 CL -1.0 / H2O -1.0 / ACA 1.0
STOIC 4 ACA -1.0 / CL 1.0 / H2O 1.0
STOIC 5 ACA -1.0 / CL 1.0 / H2O 1.0
STOIC 6 ACA -1.0 / CL 1.0 / H2O 1.0
STOIC 7 CL -1.0 / B-ACA 1.0
STOIC 8 CL -1.0 / B-ACA 1.0
STOIC 9 CL -1.0 / B-ACA 1.0
STOIC 10 B-ACA -1.0 / CL 1.0
STOIC 11 B-ACA -1.0 / CL 1.0
STOIC 12 B-ACA -1.0 / CL 1.0
STOIC 13 CL -1.0 / ACA -1.0 / T-NH2 1.0 / T-COOH 1.0
STOIC 14 CL -1.0 / ACA -1.0 / T-NH2 1.0 / T-COOH 1.0
STOIC 15 CL -1.0 / ACA -1.0 / T-NH2 1.0 / T-COOH 1.0
STOIC 16 T-NH2 -1.0 / T-COOH -1.0 / ACA 1.0 / CL 1.0
STOIC 17 T-NH2 -1.0 / T-COOH -1.0 / ACA 1.0 / CL 1.0
STOIC 18 T-NH2 -1.0 / T-COOH -1.0 / ACA 1.0 / CL 1.0
STOIC 19 CL -1.0 / B-ACA 1.0
STOIC 20 CL -1.0 / B-ACA 1.0
STOIC 21 CL -1.0 / B-ACA 1.0
RATE-CON 1  PRE-EXP=0.00424  ACT-ENERGY=19.880  TREF=260
RATE-CON 2  PRE-EXP=0.840712 ACT-ENERGY=18.806  TREF=260
RATE-CON 3  PRE-EXP=0.840712 ACT-ENERGY=18.806  TREF=260
RATE-CON 4  PRE-EXP=1.370519 ACT-ENERGY=17.962  TREF=260
RATE-CON 5  PRE-EXP=271.7817 ACT-ENERGY=16.888  TREF=260
RATE-CON 6  PRE-EXP=271.7817 ACT-ENERGY=16.888  TREF=260
RATE-CON 7  PRE-EXP=1.23117  ACT-ENERGY=22.845  TREF=260
RATE-CON 8  PRE-EXP=93.61226 ACT-ENERGY=20.107  TREF=260
RATE-CON 9  PRE-EXP=93.61226 ACT-ENERGY=20.107  TREF=260
RATE-CON 10 PRE-EXP=0.893159 ACT-ENERGY=26.888  TREF=260
RATE-CON 11 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
RATE-CON 12 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
RATE-CON 13 PRE-EXP=1.23117  ACT-ENERGY=22.845  TREF=260
RATE-CON 14 PRE-EXP=93.61226 ACT-ENERGY=20.107  TREF=260
RATE-CON 15 PRE-EXP=93.61226 ACT-ENERGY=20.107  TREF=260
RATE-CON 16 PRE-EXP=0.893159 ACT-ENERGY=26.888  TREF=260
RATE-CON 17 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
RATE-CON 18 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
RATE-CON 19 PRE-EXP=0.893159 ACT-ENERGY=26.888  TREF=260
RATE-CON 20 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
RATE-CON 21 PRE-EXP=67.83767 ACT-ENERGY=24.151  TREF=260
POWLAW-EXP 1 CL 1.0 / H2O 1.0
POWLAW-EXP 2 CL 1.0 / H2O 1.0 / T-COOH 1.0
POWLAW-EXP 3 CL 1.0 / H2O 1.0 / ACA 1.0
POWLAW-EXP 4 ACA 1.0
POWLAW-EXP 5 ACA 1.0 / T-COOH 1.0
POWLAW-EXP 6 ACA 2.0
POWLAW-EXP 7 CL 1.0 / T-NH2 1.0
POWLAW-EXP 8 CL 1.0 / T-NH2 1.0 / T-COOH 1.0
POWLAW-EXP 9 CL 1.0 / T-NH2 1.0 / ACA 1.0
POWLAW-EXP 10 T-NH2 1.0
```

Input Language Example (cont.)

```
POWLAW-EXP 11 T-NH2 1.0 / T-COOH 1.0
POWLAW-EXP 12 T-NH2 1.0 / ACA 1.0
POWLAW-EXP 13 CL 1.0 / ACA 1.0
POWLAW-EXP 14 CL 1.0 / ACA 1.0 / T-COOH 1.0
POWLAW-EXP 15 CL 1.0 / ACA 2.0
POWLAW-EXP 16 ACA 1.0
POWLAW-EXP 17 T-COOH 1.0 / ACA 1.0
POWLAW-EXP 18 ACA 2.0
POWLAW-EXP 19 ACA 1.0
POWLAW-EXP 20 ACA 1.0 / T-COOH 1.0
POWLAW-EXP 21 ACA 2.0
```

SPECIFYING FREE-RADICAL POLYMERIZATION KINETICS

The input language for the FREE-RAD REACTIONS paragraph is described below. The reaction keywords and rate coefficient parameters for free-radical polymerization are given. Users may select a subset of the built-in reactions for a given simulation.

Input Language

```
REACTIONS    reacid    FREE-RAD
  PARAM    QSSA=yes/no   QSSAZ=yes/no QSSAF=yes/no    RAD-INTENS=value
  SPECIES    INITIATOR=cid    MONOMER=cid    INHIBITOR=cid    &
             SOLVENT=cid . . .
  INIT-DEC    cid   idpre-exp    idact-energy    idact-volume    ideffic    idnrad
  INIT-CAT    cid1    cid2    icpre-exp    icact-energy    icact-volume    iceffic    icnrad
  INIT-SP    cid1    cid2    ispre-exp    isact-energy    isact-volume
  INIT-SP-EFF    cid    coeffa    coeffb    coeffc
  CHAIN-INI    cid    cipre-exp    ciact-energy    ciact-volume
  PROPAGATION    cid1    cid2    prpre-exp    pract-energy    pract-volume
  CHAT-MON    cid1    cid2    cmpre-exp    cmact-energy    cmact-volume
  CHAT-POL    cid1    cid2    cppre-exp    cpact-energy    cpact-volume
  CHAT-AGENT    cid1    cid2    capre-exp    caact-energy    caact-volume
  CHAT-SOL    cid1    cid2    cspre-exp    csact-energy    csact-volume
  B-SCISSION    cid    bspre-exp    bsact-energy    bsact-volume
  TERM-DIS    cid1    cid2    tdpre-exp    tdact-energy    tdact-volume
  TERM-COMB    cid1    cid2    tcpre-exp    tcact-energy    tcact-volume
  INHIBITION    cid1    cid2    inpre-exp    inact-energy    inact-volume
  SC-BRANCH    cid1    cid2    scpre-exp    scact-energy    scact-volume
  GEL-EFFECT    GETYPE=reactiontype    CORR-NO=corrno  &
    MAX-PARAMS=maxparams  GE-PARAMS=paramlist / ...
  SUBROUTINE    GEL-EFFECT=subname
```

Input Language Description

reacid	Paragraph ID.	
PARAM	Used to specify polymerization mechanism, radiation intensity, and request the Quasi-Steady-State Approximation (QSSA).	
	RAD-INTENS=*value*	Used to specify a value for the radiation intensity to be used for the induced initiation reaction (default is 1.0)
	QSSA=*YES/NO*	Used to request QSSA for all moments (default is NO)
	QSSAZ=*YES/NO*	Used to request QSSA for the zeroth moment only (default is NO)
	QSSAF=*YES/NO*	Used to request QSSA for the first moment only (default is NO)
	QSSAS=*YES/NO*	Used to request QSSA for the second moment only (default is NO)
SPECIES	Reacting species identification. This sentence is used to associate components in the simulation with reactive species in the built-in free-radical kinetic scheme. The following species keywords are currently valid	
	INITIATOR CATALYST COINITIATOR MONOMER CHAINTAG SOLVENT INHIBITOR POLYMER	
MON-RSEG	Specifies the pairing between monomers and their corresponding repeat segments in a polymer.	
	Monomer	Monomer ID
	R-Seg	Corresponding repeat segment ID
INIT-DEC	Identifier for initiator decomposition reaction.	
	cid1	Initiator ID
	idpre-exp	Preexponential factor
	idact-energy	Activation energy
	idact-volume	Activation volume (default is 0.0)
	ideffic	Initiator efficiency (default is 1.0)

	idnrad	Number of radicals from one initiator molecule (default is 2.0)
INIT-CAT	Identifier for catalyzed initiator decomposition reaction.	
	cid1	Initiator ID
	cid2	Catalyst ID
	icpre-exp	Preexponential factor
	icact-energy	Activation energy
	icact-volume	Activation volume (default is 0.0)
	iceffic	Initiator efficiency (default is 1.0)
	icnrad	Number of radicals from one initiator molecule (default is 2.0)
INIT-SP	Identifier for thermal and radiation induced initiation reaction.	
	cid1	Monomer ID
	cid2	Co-initiator ID
	ispre-exp	Preexponential factor
	isact-energy	Activation energy
	isact-volume	Activation volume (default is 0.0)
INIT-SP-EFF	Parameters for thermal and radiation induced initiation reaction.	
	cid	Monomer ID
	coeffa	Exponent for coinitiator concentration (default is 0.0)
	coeffb	Exponent for monomer concentration (default is 0.0)
	coeffc	Exponent for radiation intensity (default is 0.0)
CHAIN-INI	Identifier for chain initiation reaction.	
	cid1	Monomer ID
	cipre-exp	Preexponential factor
	ciact-energy	Activation energy
	ciact-volume	Activation volume (default is 0.0)

PROPAGATION Identifier for chain propagation reaction.

cid1	Monomer corresponding to polymer active segment ID
cid2	Monomer ID
prpre-exp	Preexponential factor
pract-energy	Activation energy
pract-volume	Activation volume (default is 0.0)

CHAT-MON Identifier for chain transfer to monomer reaction.

cid1	Monomer corresponding to polymer active segment ID
cid2	Monomer ID
cmpre-exp	Preexponential factor
cmact-energy	Activation energy
cmact-volume	Activation volume (default is 0.0)

CHAT-POL Identifier for chain transfer to polymer reaction.

cid1	Monomer corresponding to polymer active segment ID
cid2	Monomer corresponding to polymer reacting segment ID on dead chain
cppre-exp	Preexponential factor
cpact-energy	Activation energy
cpact-volume	Activation volume (default is 0.0)

CHAT-AGENT Identifier for chain transfer to transfer agent reaction.

cid1	Monomer corresponding to polymer active segment ID
cid2	Transfer agent ID
capre-exp	Preexponential factor
caact-energy	Activation energy
caact-volume	Activation volume (default is 0.0)

CHAT-SOL	Identifier for chain transfer to solvent reaction.	
	cid1	Monomer corresponding it polymer active segment ID
	cid2	Solvent ID
	cspre-exp	Preexponential factor
	csact-energy	Activation energy
	csact-volume	Activation volume (default is 0.0)
B-SCISSION	Identifier for beta-scission reaction.	
	cid1	Monomer corresponding to polymer active segment ID
	bspre-exp	Preexponential factor
	bsact-energy	Activation energy
	bsact-volume	Activation volume (default is 0.0)
TERM-DIS	Identifier for chain termination by disproportionation reaction.	
	cid1	Monomer corresponding to first polymer active segment ID
	cid2	Monomer corresponding to second polymer active segment ID
	tdpre-exp	Preexponential factor
	tdact-energy	Activation energy
	tdact-volume	Activation volume (default is 0.0)
TERM-COMB	Identifier for chain termination by combination reaction.	
	cid1	Monomer corresponding to first polymer active segment ID
	cid2	Monomer corresponding to second polymer active segment ID
	tcpre-exp	Preexponential factor
	tcact-energy	Activation energy
	tcact-volume	Activation volume (default is 0.0)

INHIBITION	Identifier for chain inhibition reaction.

cid1	Monomer corresponding to polymer active segment ID
cid2	Inhibitor ID
inpre-exp	Preexponential factor
inact-energy	Activation energy
inact-volume	Activation volume (default is 0.0)

SC-BRANCH	Identifier for short chain branching reaction.

cid1	Monomer corresponding to reactant polymer active segment ID
cid2	Monomer corresponding to product active segment ID
scpre-exp	Preexponential factor
scact-energy	Activation energy
scact-volume	Activation volume (default is 0.0)

GEL-EFFECT	Gel effect switch and correlation selection. This sentence is used to

– Include a gel effect for any reactions in the built-in kinetic scheme and for the initiator efficiency

– Select a gel effect correlation from a list of built-in and user specified gel effect correlations

– Specify the maximum number of parameters

– Specify the parameter values for the selected correlation

The default action is to not include a gel effect.

GETYPE= *reactiontype*	Used to identify the type of reaction to apply gel effect to. A list of valid reaction types follows
INITIATION	Initiator decomposition
INIT-EFF	Initiator efficiency
PROPAGATION	Propagation, chain initiation and induced initiation reactions
CHAT-MON	Chain transfer to monomer

	CHAT-POL	Chain transfer to polymer
	CHAT-AGENT	Chain transfer to agent
	CHAT-SOL	Chain transfer to solvent
	TERMINATION	Termination
	CORR-NO=*corrno*	Used to select a correlation number. If a correlation number greater than the number of built-in correlations (currently 2) is specified then the user should supply a Fortran subroutine containing the user gel effect correlation.
	MAX-PARAMS= *maxparams*	Used to enter the maximum number of gel effect parameters for the correlation selected.
	GE-PARAMS= *paramlist*	Used to enter a list of parameters for the correlation selected.
SUBROUTINE	User subroutines sentence.	
	GEL-EFFECT=*subname*	Used to specify the name of the subroutine containing user gel effect correlations. The user gel-effect subroutine argument list was shown in Table 4.42. A Fortran template called USRGEL.F is available for your use.

Input Language Example

```
REACTIONS R1 FREE-RAD

   DESCRIPTION 'EXAMPLE FREE-RADICAL INPUT'
   SPECIES INITIATOR=INI1  MONOMER=STY ACN SOLVENT=XYLENE POLYMER=SAN
;
;  RATE COEFFICIENT   PARAMETERS FOR MODIFIED ARRHENIUS EXPRESSION
;                     FREQUENCY    ACTIVATION  ACTIVATION  EFF   #RAD
                      FACTOR       ENERGY      VOLUME
                                (J/KMOL)       (CUM)
   INIT-DEC       INI1  3.71D-5  0.0           0.D0        1.0     2
   CHAIN-INI       STY  4820.    0.0           0.D0
   CHAIN-INI       ACN  225.     0.0           0.D0
   PROPAGATION STY STY  4820.    0.0           0.D0
   PROPAGATION STY ANC  10277.   0.0           0.D0
   PROPAGATION ACN STY  7165.6   0.0           0.D0
   PROPAGATION ACN ACN  225.     0.0           0.D0
   TERM-COMB   STY STY  1.39D7   0.0           0.D0
   TERM-COMB   STY ACN  1.79D9   0.0           0.D0
   TERM-COMB   ACN STY  1.79D9   0.0           0.D0
   TERM-COMB   ACN ACN  1.02D8   0.0           0.D0
   CHAT-MON    STY STY  0.289    0.0           0.D0
   CHAT-MON    STY ACN  0.289    0.0           0.D0
   CHAT-MON    ACN STY  0.043    0.0           0.D0
   CHAT-MON    ACN ACN  0.043    0.0           0.D0
   GEL-EFFECT  GETYPE=TERMINATION CORR-NO=2 MAX-PARAMS=10   &
     GE-PARAMS=1 0 2.57 -5.05E-3 9.56 -1.76E-2 -3.03 7.85E-3 0.0 2
```

SPECIFYING EMULSION POLYMERIZATION KINETICS

The input language for the EMULSION REACTIONS paragraph is described below. Users are able to select the phases in which the reactions are occurring and also define the kinetics of particle absorption, desorption, and termination.

Input Language

```
REACTIONS   reacid   EMULSION
  PARAM   KBASIS=mole/mass
  SPLIT-PM   spm-cid kll
  SPECIES   INITIATOR=cid   MONOMER=cid   INHIBITOR=cid   &
            DISPERSANT=cid . . .
  INIT-DEC   phasid  cid  idpre-exp  idact-energy  [idact-volume]  ideffic  idnrad
  INIT-CAT  phasid  cid1  cid2  icpre-exp  icact-energy  [icact-volume]  iceffic  icnrad
  INIT-SP   phasid  cid1  cid2  ispre-exp  isact-energy  [isact-volume]
  INIT-ACT  phasid  cid1  cid2  iapre-exp  iaact-energy  [iaact-volume]  iaeffic  ianrad
  CHAIN-INI   phasid   cid   cipre-exp   ciact-energy   [ciact-volume]
  PROPAGATION  phasid  cid1   cid2   prpre-exp   pract-energy   [pract-volume]
  CHAT-MON   phasid  cid1   cid2   cmpre-exp   cmact-energy   [cmact-volume]
  CHAT-POL   phasid  cid1  cid2   cppre-exp   cpact-energy   [cpact-volume]
  CHAT-AGENT  phasid   cid1  cid2   capre-exp   caact-energy   [caact-volume]
  B-SCISSION   phasid   cid   bspre-exp   bsact-energy   [bsact-volume]
  TERM-DIS  phasid  cid1   cid2   tdpre-exp   tdact-energy   [tdact-volume]
  TERM-COMB  phasid  cid1   cid2   tcpre-exp   tcact-energy   [tcact-volume]
  INHIBITION   phasid  cid1   cid2   inpre-exp   inact-energy   [inact-volume]
  SC-BRANCH   phasid  cid1   cid2   scpre-exp   scact-energy   [scact-volume]
  REDUCTION phasid cid1 cid2 rdpre-exp rdact-energy [rdact-volume] rdeffic rdnrad
  OXIDATION  phasid  cid1  cid2  oxpre-exp  oxact-energy  [oxact-volume
  INIT-SP-EFF   cid   coeffa   coeffb   coeffc
  GEL-EFFECT   GETYPE=reactiontype   CORR-NO=corrno  &
    MAX-PARAMS=maxparams  GE-PARAMS=paramlist / ...
  SUBROUTINE   GEL-EFFECT=subname
  ABS-MIC   ampre-exp   amact-energy
  ABS-PART   appre-exp   apact-energy
  DES-PART   dppre-exp   dpact-energy
  EMUL-PARAMS   emulid   cmc-conc   area
```

Input Language Description

reacid	Paragraph ID.	
PARAM	Use to enter basis parameters.	
	KBASIS=*mole/mass*	Basis for phase split ratios
SPLIT-PM	Used to enter homosaturation solubility of species in the polymer phase.	
	spm-cid	Component ID of the species partitioning into the polymer phase
	kll	Ratio of fraction of species in polymer phase to fraction in monomer phase. KBASIS determines whether amount is in moles or in mass
SPECIES	Reacting species identification. This sentence is used to associate components in the simulation with species in the built-in free-radical kinetic scheme. The following species keywords are currently valid	
	INITIATOR COINITIATOR CATALYST MONOMER CHAINTAG DISPERSANT INHIBITOR POLYMER	
INIT-DEC	Identifier for initiator decomposition reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid	Initiator ID
	idpre-exp	Preexponential factor
	idact-energy	Activation energy
	idact-volume	Activation volume (optional)
	ideffic	Initiator efficiency
	idnrad	Number of radicals from one initiator molecule
INIT-CAT	Identifier for catalyzed initiator decomposition reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Initiator ID
	cid2	Catalyst ID
	icpre-exp	Preexponential factor
	icact-energy	Activation energy
	icact-volume	Activation volume (optional)

	iceffic	Initiator efficiency
	icnrad	Number of radicals from one initiator molecule
INIT-SP	Identifier for thermal and radiation induced initiation reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Monomer ID
	cid2	Co-initiator ID
	ispre-exp	Preexponential factor
	isact-energy	Activation energy
	isact-volume	Activation volume (optional)
INIT-ACT	Identifier for initiation by activator and initiator.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Initiator ID
	cid2	Activator ID
	iapre-exp	Preexponential factor
	iaact-energy	Activation energy
	iaact-volume	Activation volume (optional)
	iaeffic	Initiator activation efficiency
	ianrad	Initiator activation number of radicals
CHAIN-INI	Identifier for chain initiation reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid	Monomer ID
	cipre-exp	Preexponential factor
	ciact-energy	Activation energy
	ciact-volume	Activation volume (optional)

PROPAGATION Identifier for chain propagation reaction.

phasid	Reaction phase (POLYMER or DISPERSANT)
cid1	Monomer corresponding to active polymer segment ID
cid2	Monomer ID
prpre-exp	Preexponential factor
pract-energy	Activation energy
pract-volume	Activation volume (optional)

CHAT-MON Identifier for chain transfer to monomer reaction.

phasid	Reaction phase (POLYMER or DISPERSANT)
cid1	Monomer corresponding to active polymer segment ID
cid2	Monomer ID
cmpre-exp	Preexponential factor
cmact-energy	Activation energy
cmact-volume	Activation volume (optional)

CHAT-POL Identifier for chain transfer to polymer reaction.

phasid	Reaction phase (POLYMER or DISPERSANT)
cid1	Monomer corresponding to active polymer segment ID
cid2	Monomer corresponding to reacting polymer segment ID or dead chain
cppre-exp	Preexponential factor
cpact-energy	Activation energy
cpact-volume	Activation volume (optional)

CHAT-AGENT Identifier for chain transfer to transfer agent reaction.

phasid	Reaction phase (POLYMER or DISPERSANT)
cid1	Monomer corresponding to active polymer segment ID
cid2	Transfer agent ID

	capre-exp	Preexponential factor
	caact-energy	Activation energy
	caact-volume	Activation volume (optional)
B-SCISSION	Identifier for beta-scission reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid	Active polymer segment ID
	bspre-exp	Preexponential factor
	bsact-energy	Activation energy
	bsact-volume	Activation volume (optional)
TERM-DIS	Identifier for chain termination by disproportionation reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	First active polymer segment ID
	cid2	Second active polymer segment ID
	tdpre-exp	Preexponential factor
	tdact-energy	Activation energy
	tdact-volume	Activation volume (optional)
TERM-COMB	Identifier for chain termination by combination reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	First active polymer segment ID
	cid2	Second active polymer segment ID
	tcpre-exp	Preexponential factor
	tcact-energy	Activation energy
	tcact-volume	Activation volume (optional)
INHIBITION	Identifier for chain inhibition reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Active polymer segment ID
	cid2	Inhibitor ID
	inpre-exp	Preexponential factor

	inact-energy	Activation energy
	inact-volume	Activation volume (optional)
SC-BRANCH	Identifier for short chain branching reaction.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Reactant active polymer segment ID
	cid2	Product active segment ID
	scpre-exp	Preexponential factor
	scact-energy	Activation energy
	scact-volume	Activation volume (optional)
REDUCTION	Identifier for reduction step of redox initiation.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Activator ID
	cid2	Redox agent (catalyst) ID
	rdpre-exp	Preexponential factor
	rdact-energy	Activation energy
	rdact-volume	Activation volume (optional)
	rdeffic	Reduction activation efficiency
	rdnrad	Reduction activation number of radicals
OXIDATION	Identifier for oxidation step of redox initiation.	
	phasid	Reaction phase (POLYMER or DISPERSANT)
	cid1	Initiator ID
	cid2	Redox agent (catalyst) ID
	oxpre-exp	Preexponential factor
	oxact-energy	Activation energy
	oxact-volume	Activation volume (optional)

INIT-SP-EFF	Parameters for thermal and radiation induced initiation reaction.	
	cid1	Monomer ID
	coeffa	Exponent for coinitiator concentration (default is 1.0)
	coeffb	Exponent for monomer concentration (default is 1.0)
	coeffc	Exponent for radiation intensity (default is 1.0)

GEL-EFFECT Gel effect switch and correlation selection. This sentence is used to

- Include a gel effect for any reactions in the built-in kinetic scheme and for the initiator efficiency
- Select a gel effect correlation from a list of built-in and user specified gel effect correlations
- Specify the maximum number of parameters
- Specify the parameter values for the selected correlation

The default action is to not include a gel effect.

GETYPE= *reactiontype*	Used to identify the type of reaction to apply gel effect to. A list of valid reaction types follows
INITIATION	Initiator decomposition
INIT-EFF	Initiator efficiency
PROPAGATION	Propagation, chain initiation and induced initiation reactions
CHAT-MON	Chain transfer to monomer
CHAT-POL	Chain transfer to polymer
CHAT-AG	Chain transfer to agent
TERMINATION	Termination
CORR-NO=*corrno*	Used to select a correlation number. If a correlation number greater than the number of built-in correlations (currently 2) is specified then the user should supply a Fortran subroutine containing the user gel effect correlation.

	MAX-PARAMS= *maxparams*	Used to enter the maximum number of gel effect parameters for the correlation selected.
	GE-PARAMS= *paramlist*	Used to enter a list of parameters for the correlation selected.
SUBROUTINE	User subroutines sentence.	
	GEL-EFFECT= *subname*	Used to specify the name of the subroutine containing user gel effect correlations. The user gel-effect subroutine argument list was shown in Table 4.42. A Fortran template called USRGEL.F is available for your use.
ABS-MIC	Used to specify rate of radical absorption by micelles.	
	ampre-exp	Preexponential factor
	amact-energy	Activation energy
ABS-PART	Used to specify rate of radical absorption by particles.	
	appre-exp	Preexponential factor
	apact-energy	Activation energy
DES-PART	Identifier for radical desorption.	
	dppre-exp	Preexponential factor
	dpact-energy	Activation energy
EMUL-PARAMS	Used to specify emulsion parameters for micellar nucleation.	
	emulid	Emulsifier ID
	cmc-conc	Critical micelle concentration
	area	Surface coverage or area per unit mole of emulsifier

Input Language Example

```
REACTIONS EMLRXN EMULSION
    DESCRIPTION "EXAMPLE EMULSION INPUT"
    PARAM KBASIS=MASS QSSA=YES
    SPECIES INITIATOR=APS MONOMER=STY NBA EMULSIFIER=EMUL  &
        DISPERSANT=H2O POLYMER=POLYMER
    INIT-DEC DISPERSANT APS 1.0000E+16 1.4020E+08 0.0  &
        EFFIC=.80 NRADS=2
    PROPAGATION POLYMER STY STY 2341450.0 2.6000E+07
    PROPAGATION POLYMER STY NBA 3265600.0 2.6000E+07
    PROPAGATION POLYMER NBA NBA 1909530.0 2.2400E+07
    PROPAGATION POLYMER NBA STY 1.4918E+07 2.2400E+07
    CHAT-MON POLYMER STY STY 3310000.0 5.3020E+07
    CHAT-MON POLYMER STY NBA 3310000.0 5.3020E+07
    CHAT-MON POLYMER NBA NBA 438.90 2.7600E+07
    CHAT-MON POLYMER NBA STY 438.90 2.7600E+07
    TERM-COMB POLYMER STY STY 1.6125E+09 7000000.0
    TERM-COMB POLYMER STY NBA 7.3204E+09 1.4600E+07
    TERM-COMB POLYMER NBA NBA 3.3217E+10 2.2200E+07
    TERM-COMB POLYMER NBA STY 7.3204E+09 1.4600E+07
    ABS-MIC 1.0000E-07 0.0
    ABS-PART 1.0000E-07 0.0
    DES-PART 0.0 0.0
    EMUL-PARAMS EMUL 0.0 5.0000E+08
    SPLIT-PM STY .40
    SPLIT-PM NBA .40
```

SPECIFYING ZIEGLER-NATTA POLYMERIZATION KINETICS

The input language for the part of the polymerization REACTIONS paragraph specific to Ziegler-Natta kinetics is described below. Ziegler-Natta inputs may be used to define the reaction kinetics for a wide variety of homo- and co-polymers produced by catalyzed polymerization, including HDPE. A subset of the built-in kinetics can be defined for a simulation by including the reaction keywords for the desired reactions and specifying the rate coefficient parameters for these reactions. The reaction keywords and rate coefficient parameters for Ziegler-Natta polymerization are described below. Currently for two phase systems the polymerization reactions are applied to the liquid phase in the reactor. For gas phase polymerization systems the solid polymer, or the amorphous part of the polymer, is modeled as a liquid.

Input Language

```
REACTIONS     reacid    ZIEGLER-NAT
  SPECIES     CATALYST=cid        COCATALYST=cid      MONOMER=cid  &
              CHAINTAG=cid        SOLVENT=cid         POISON=cid  &
              BYPRODUCT=cid       HYDROGEN=cid        POLYMER=cid  &
              ELECDONOR=cid       TDBSEGMENT=cid
  ACT-SPON        site-id  cid1         aspre-exp  asact-energy  asorder
  ACT-COCAT       site-id  cid1  cid2   acpre-exp  acact-energy  acorder
  ACT-EDONOR      site-id  cid1  cid2   aepre-exp  aeact-energy  aeorder
  ACT-H2          site-id  cid1  cid2   ahpre-exp  ahact-energy  ahorder
  ACT-MON         site-id  cid1  cid2   ampre-exp  amact-energy  amorder
  CHAIN-INI       site-id  cid1         cipre-exp  ciact-energy  ciorder
  PROPAGATION     site-id  cid1  cid2   prpre-exp  pract-energy  prorder
  CHAT-MON        site-id  cid1  cid2   cmpre-exp  cmact-energy  cmorder  cmtdb-frac
  CHAT-AGENT      site-id  cid1  cid2   capre-exp  caact-energy  caorder  catdb-frac
  CHAT-SOL        site-id  cid1  cid2   cspre-exp  csact-energy  csorder  cstdb-frac
  CHAT-COCAT      site-id  cid1  cid2   ccpre-exp  ccact-energy  ccorder  cctdb-frac
  CHAT-H2         site-id  cid1  cid2   chpre-exp  chact-energy  chorder  chtdb-frac
  CHAT-EDONOR     site-id  cid1  cid2   cepre-exp  ceact-energy  ceorder  cetdb-frac
  CHAT-SPON       site-id  cid1  cid2   cnpre-exp  cnact-energy  cnorder  cntdb-frac
  DEACT-POISON    site-id  cid1         dppre-exp  dpact-energy  dporder
  DEACT-COCAT     site-id  cid1         dcpre-exp  dcact-energy  dcorder
  DEACT-MON       site-id  cid1         dmpre-exp  dmact-energy  dmorder
  DEACT-EDONOR    site-id  cid1         depre-exp  deact-energy  deorder
  DEACT-H2        site-id  cid1         dhpre-exp  dhact-energy  dhorder
  DEACT-SPON      site-id               dspre-exp  dsact-energy  dsorder
  COCAT-POISON             cid1  cid2   copre-exp  coact-energy  coorder
  FSINH-H2        site-id  cid1         fhpre-exp  fhact-energy  fhorder
  RSINH-H2        site-id  cid1         rhpre-exp  rhact-energy  rhorder
  FSINH-POISON    site-id  cid1         fppre-exp  fpact-energy  fporder
  RSINH-POISON    site-id  cid1         rppre-exp  rpact-energy  rporder
  TDB-POLY        site-id  cid1  cid2   tdpre-exp  tdact-energy  tdorder
  ATACT-PROP      site-id  cid1  cid2   atpre-exp  atact-energy  atorder
```

Input Language Description

reacid	Reaction paragraph ID.

SPECIES — Reacting species identification. This sentence is used to associate components in the simulation with the reactive species in the built-in kinetic scheme. The following species keywords are currently valid

CATALYST
CHAINTAG
BYPRODUCT
ELECDONOR
COCATALYST
SOLVENT
HYDROGEN
TDBSEGMENT
MONOMER
POISON
POLYMER

MON-RSEG — Specifies the pairing between monomers and their corresponding repeat segments in a polymer.

monomer	Monomer ID
R-Seg	Corresponding repeat segment ID

ACT-SPON — Reaction identifier for spontaneous site activation of a catalyst potential site to a vacant active site of type k.

site-id	Site type identifier for active site formed (k = 1, 2, ... , NSITE)
cid1	Component ID of catalyst
aspre-exp	Preexponential factor (default is 0.0)
asact-energy	Activation energy (default is 0.0)
asorder	Reaction order for potential site concentration (default is 0.0)

ACT-COCAT — Reaction identifier for site activation by cocatalyst of a catalyst potential site to a vacant active site of type k.

site-id	Site type identifier for active site (k = 1, 2, ... , NSITE)
cid1	Component ID of catalyst
cid2	Component ID of cocatalyst
acpre-exp	Preexponential factor (default is 0.0)
acact-energy	Activation energy (default is 0.0)
acorder	Reaction order for cocatalyst concentration (default is 0.0)

ACT-EDONOR	Reaction identifier for site activation by electron donor of a catalyst potential site to a vacant active site of type k.	
	site-id	Site type identifier for active site formed (k = 1, 2, ... , NSITE)
	cid1	Component ID of catalyst
	cid2	Component ID of electron donor
	aepre-exp	Preexponential factor (default is 0.0)
	aeact-energy	Activation energy (default is 0.0)
	aeorder	Reaction order for electron donor concentration (default is 0.0)
ACT-H2	Reaction identifier for site activation by hydrogen of a catalyst potential site to a vacant active site of type k.	
	site-id	Site type identifier for active site formed (k = 1, 2, ... , NSITE)
	cid1	Component ID of catalyst
	cid2	Component ID of hydrogen
	ahpre-exp	Preexponential factor (default is 0.0)
	ahact-energy	Activation energy (default is 0.0)
	ahorder	Reaction order for hydrogen concentration (default is 0.0)
ACT-MON	Reaction identifier for site activation by monomer of a catalyst potential site to a vacant active site of type k.	
	site-id	Site type identifier for active site formed (k = 1, 2, ... , NSITE)
	cid1	Component ID of catalyst
	cid2	Component ID of monomer
	ampre-exp	Preexponential factor (default is 0.0)
	amact-energy	Activation energy (default is 0.0)
	amorder	Reaction order for monomer concentration (default is 0.0)

CHAIN-INI Reaction identifier for polymer chain initiation on a vacant active site of type k. The vacant site becomes a propagation site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of monomer
cipre-exp	Preexponential factor (default is 0.0)
ciact-energy	Activation energy (default is 0.0)
ciorder	Reaction order for monomer concentration (default is 0.0)

PROPAGATION Reaction identifier for polymer chain propagation on an active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of monomer
prpre-exp	Preexponential factor (default is 0.0)
pract-energy	Activation energy (default is 0.0)
prorder	Reaction order for monomer concentration (default is 0.0)

CHAT-MON Reaction identifier for chain transfer to monomer on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of monomer
cmpre-exp	Preexponential factor (default is 0.0)
cmact-energy	Activation energy (default is 0.0)
cmorder	Reaction order for monomer concentration (default is 0.0)
cmtdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-AGENT Reaction identifier for chain transfer to agent on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of chain transfer agent
capre-exp	Preexponential factor (default is 0.0)
caact-energy	Activation energy (default is 0.0)
caorder	Reaction order for agent concentration (default is 0.0)
catdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-SOL Reaction identifier for chain transfer to solvent on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of solvent
cspre-exp	Preexponential factor (default is 0.0)
csact-energy	Activation energy (default is 0.0)
csorder	Reaction order for solvent concentration (default is 0.0)
cstdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-COCAT Reaction identifier for chain transfer to cocatalyst on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of cocatalyst
ccpre-exp	Preexponential factor (default is 0.0)
ccact-energy	Activation energy (default is 0.0)

ccorder	Reaction order for cocatalyst concentration (default is 0.0)
cctdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-H2	Reaction identifier for chain transfer to hydrogen on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of hydrogen
chpre-exp	Preexponential factor (default is 0.0)
chact-energy	Activation energy (default is 0.0)
chorder	Reaction order for hydrogen concentration (default is 0.0)
chtdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-EDONOR	Reaction identifier for chain transfer to electron donor on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cid2	Component ID of electron donor
cepre-exp	Preexponential factor (default is 0.0)
ceact-energy	Activation energy (default is 0.0)
ceorder	Reaction order for electron donor concentration (default is 0.0)
cetdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

CHAT-SPON	Reaction identifier for spontaneous chain transfer on active site of type k.

site-id	Site type identifier (k = 1, 2, ... , NSITE)

cid1	Component ID of active segment (specified in terms of the corresponding monomer ID)
cnpre-exp	Preexponential factor (default is 0.0)
cnact-energy	Activation energy (default is 0.0)
cnorder	Reaction order (not used)
cntdb-frac	Fraction of generated dead polymer chains with terminal double bonds (default is 0.0)

DEACT-POISON Reaction identifier for site deactivation by poison of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of poison
dppre-exp	Preexponential factor (default is 0.0)
dpact-energy	Activation energy (default is 0.0)
dporder	Reaction order for poison concentration (default is 0.0)

DEACT-COCAT Reaction identifier for site deactivation by cocatalyst of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of cocatalyst
dcpre-exp	Preexponential factor (default is 0.0)
dcact-energy	Activation energy (default is 0.0)
dcorder	Reaction order for cocatalyst concentration (default is 0.0)

DEACT-MON Reaction identifier for site deactivation by monomer of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of monomer
dmpre-exp	Preexponential factor (default is 0.0)
dmact-energy	Activation energy (default is 0.0)
dmorder	Reaction order for monomer concentration (default is 0.0)

DEACT-EDONOR Reaction identifier for site deactivation by electron donor of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of electron donor
depre-exp	Preexponential factor (default is 0.0)
deact-energy	Activation energy (default is 0.0)
deorder	Reaction order for electron donor concentration (default is 0.0)

DEACT-H2 Reaction identifier for site deactivation by hydrogen of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of hydrogen
dhpre-exp	Preexponential factor (default is 0.0)
dhact-energy	Activation energy (default is 0.0)
dhorder	Reaction order for hydrogen concentration (default is 0.0)

DEACT-SPON Reaction identifier for spontaneous site deactivation of a catalyst active site of type k to a dead site.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
dspre-exp	Preexponential factor (default is 0.0)
dsact-energy	Activation energy (default is 0.0)
dsorder	Reaction order (not used)

COCAT-POISON Reaction identifier for cocatalyst poisoning reaction.

cid1	Component ID of cocatalyst
cid2	Component ID of poison
copre-exp	Preexponential factor (default is 0.0)
coact-energy	Activation energy (default is 0.0)
coorder	Reaction order (not used)

FSINH-H2 Reaction identifier for site inhibition by hydrogen-forward reaction.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of hydrogen
fhpre-exp	Preexponential factor (default is 0.0)
fhact-energy	Activation energy (default is 0.0)
fhorder	Reaction order for hydrogen concentration (default is 0.0)

RSINH-H2 Reaction identifier for site inhibition by hydrogen-reverse reaction.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of hydrogen
rhpre-exp	Preexponential factor (default is 0.0)
rhact-energy	Activation energy (default is 0.0)
rhorder	Reaction order for inhibited site concentration (default is 0.0)

FSINH-POISON Reaction identifier for site inhibition by poison-forward reaction.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of poison
fppre-exp	Preexponential factor (default is 0.0)
fpact-energy	Activation energy (default is 0.0)
fporder	Reaction order for poison concentration (default is 0.0)

RSINH-POISON Reaction identifier for site inhibition by poison-reverse reaction.

site-id	Site type identifier (k = 1, 2, ... , NSITE)
cid1	Component ID of poison
rppre-exp	Preexponential factor (default is 0.0)
rpact-energy	Activation energy (default is 0.0)
rporder	Reaction order for inhibited site concentration (default is 0.0)

TDB-POLY	Reaction identifier for terminal double bond propagation reaction.	
	site-id	Site type identifier (k = 1, 2, ... , NSITE)
	cid1	Component ID of active segment (specified in terms of the corresponding monomer)
	cid2	Component ID of TDB segment
	tdpre-exp	Preexponential factor (default is 0.0)
	tdact-energy	Activation energy (default is 0.0)
	tdorder	Reaction order (not used)
ATACT-PROP	Reaction identifier for atactic propagation reaction.	
	site-id	Site type identifier (k = 1, 2, ... , NSITE)
	cid1	Component ID of active segment (specified in terms of the corresponding monomer)
	cid2	Component ID of monomer
	atpre-exp	Preexponential factor (default is 0.0)
	atact-energy	Activation energy (default is 0.0)
	atorder	Reaction order for monomer concentration (default is 0.0)

Input Language Example

```
REACTIONS ZN-R2 ZIEGLER-NAT
    DESCRIPTION "ZIEGLER-NATTA KINETIC SCHEME"
    SPECIES CATALYST=CAT COCATALYST=CCAT MONOMER=E2  &
            SOLVENT=HEXANE HYDROGEN=H2 POLYMER=HDPE
    ACT-SPON    1 CAT      .080  0.0  1.0
    ACT-SPON    2 CAT      .080  0.0  1.0
    ACT-SPON    3 CAT      .080  0.0  1.0
    ACT-SPON    4 CAT      .080  0.0  1.0
    ACT-COCAT   1 CAT CCAT .150  0.0  1.0
    ACT-COCAT   2 CAT CCAT .150  0.0  1.0
    ACT-COCAT   3 CAT CCAT .150  0.0  1.0
    ACT-COCAT   4 CAT CCAT .150  0.0  1.0
    CHAIN-INI   1 E2       255.0 0.0  1.0
    CHAIN-INI   2 E2       90.0  0.0  1.0
    CHAIN-INI   3 E2       255.0 0.0  1.0
    CHAIN-INI   4 E2       90.0  0.0  1.0
    PROPAGATION 1 E2 E2    255.0  0.0  1.0
    PROPAGATION 2 E2 E2    90.0   0.0  1.0
    PROPAGATION 3 E2 E2    255.0  0.0  1.0
    PROPAGATION 4 E2 E2    90.0   0.0  1.0
    CHAT-MON    1 E2 E2   .090   0.0  1.0
    CHAT-MON    2 E2 E2   .240   0.0  1.0
    CHAT-MON    3 E2 E2   .090   0.0  1.0
    CHAT-MON    4 E2 E2   .240   0.0  1.0
    CHAT-H2     1 E2 H2   5.550  0.0  1.0
    CHAT-H2     2 E2 H2   18.50  0.0  1.0
    CHAT-H2     3 E2 H2   5.550  0.0  1.0
    CHAT-H2     4 E2 H2   18.50  0.0  1.0
    CHAT-SPON   1 E2      .0040  0.0  1.0
    CHAT-SPON   2 E2      .0120  0.0  1.0
    CHAT-SPON   3 E2      .0040  0.0  1.0
    CHAT-SPON   4 E2      .0120  0.0  1.0
    DEACT-SPON  1         .00010 0.0  1.0
    DEACT-SPON  2         .00060 0.0  1.0
    DEACT-SPON  3         .00010 0.0  1.0
    DEACT-SPON  4         .00060 0.0  1.0
```

SPECIFYING IONIC POLYMERIZATION KINETICS

The input language for the IONIC REACTIONS paragraph is described below.

Input Language

```
REACTIONS   reacid   IONIC
  SPECIES   ASSO-INI=cid        INIT=cid            CATALYST=cid  &
            EX-AGENT=cid        CT-AGENT=cid        TM-AGENT=cid  &
            POLYMERS
  MON-SEG   cid   segid   /  cid   segid  / …
  INIT-DISSOC  cid1  cid2  idpre-exp-f  idact-ener-f  idpre-exp-r  idact-ener-r  idasso-no &
               idref-temp
  ACT-CATALYST  site-id cid1   cid2   acpre-exp-f   acact-ener-f   acpre-exp-r   acact-ener-r &
                accoefb   accoefd acref-temp
  CHAIN-INI-1   site-id   cid   i1pre-exp-f   i1act-ener-f   i1ref-temp
  CHAIN-INI-2   site-id   cid1   cid2   i2pre-exp-f   i2act-ener-f   i2coefd
  CHAIN-INI-T   site-id   cid   itpre-exp-f   itact-ener-f   itref-temp
  PROPAGATION   site-id   cid1   cid2   prpre-exp-f   pract-ener-f   prref-temp
  ASSOCIATION   site-id   cid   aspre-exp-f     asact-ener-f   aspre-exp-r   asact-ener-r
  EXCH-GENERAL  rxn id  site-id1  cid1  site-id2  cid2  egpre-exp-f  egact-ener-f  egref-temp
  EXCH-AGENT   rxn id   site-id1   cid1   site-id2   cid2   eapre-exp-f   eaact-ener-f   &
               eapre-expr  eaact-ener-r  eacoefd  earef-temp
  EQUILIB-CION   site-id1   cid1   site-id2   eqpre-exp-f   eqact-ener-f   eqpre-exp-r   &
                 eqexp-ener-r  eqcoefd  eqref-temp
  CHAT-SPON    site-id    cid    cspre-exp-f    csact-ener-f    csref-temp
  CHAT-MONOMER    site-id    cid1    cid2    cmpre-exp-f    cmact-ener-f    cmref-temp
  CHAT-DORM-P   rxn id   site-id1   cid1   site-id2   cid2  cdpre-exp-f  cdact-ener-f  cdref-temp
  CHAT-AGENT    site-id   cid1   cid2   capre-exp-f   caact-ener-f   caorder   caref-temp
  TERM-C-ION   site-id    cid    tcpre-exp    tcact-energy    tccoefb    tcref-temp
  TERM-AGENT   site-id   cid1   cid2   tapre-exp-f   taact-ener-f   taorder   taref-temp
```

Input Language Description

reacid	Reaction paragraph ID.	
SPECIES	Reacting species identification. This sentence is used to associate components in the simulation with the reactive species in the built-in kinetic scheme. The following species keywords are currently valid: ASSOC-INIT MONOMER TERM-AGENT INITIATOR EXCH-AGENT POLYMER CATALYST CHAT-AGENT	
MON-RSEG	Identifying the reacting monomer and the corresponding repeat segment associated with it.	
	cid1	Component ID of monomer
	cid2	Component ID of corresponding repeat segment
INIT-DISSOC	Reaction identifier for initiator dissociation reaction. Associated initiator of type m dissociates into type p initiator.	
	cid1	Component ID of associated initiator
	cid2	Component ID of catalyst
	idpre-exp-f	Preexponential factor for forward reaction
	idact-ener-f	Activation energy for forward reaction
	idpre-exp-r	Preexponential factor for reverse reaction
	idact-ener-r	Activation energy for reverse reaction
	idasso-no	Initiator Association number
	idref-temp	Reference temperature
ACT-CATALYST	Reaction identifier for active species activation by catalyst of type n of an initiator of type m to form active species and/or counter-ion of type i.	
	site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
	cid1	Component ID of initiator
	cid2	Component ID of catalyst
	acpre-exp-f	Preexponential factor for forward reaction
	acact-ener-f	Activation energy for forward reaction

acpre-exp-r	Preexponential factor for reverse reaction
acact-ener-r	Activation energy for reverse reaction
accoefb	0 if cid2 does not participate in the reaction. 1 if cid2 participates in the reaction
accoefd	0 if counter-ion is absent. 1 if counter-ion is present
acref-temp	Reference temperature

CHAIN-INI-1	Reaction identifier for chain initiation of active species of type i by monomer of type j.

site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
cid	Component ID of monomer
i1pre-exp-f	Preexponential factor
i1act-ener-f	Activation energy
i1ref-temp	Reference temperature

CHAIN-INI-2	Reaction identifier for chain initiation of active species of type i by monomer of type j reacting with initiator of type m.

site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
cid1	Component ID of initiator
cid2	Component ID of monomer
i2pre-exp-f	Preexponential factor
i2act-ener-f	Activation energy
i2coefd	1 if counter-ion is formed. 0 otherwise
tref	Reference temperature

CHAIN-INI-T	Reaction identifier for chain initiation of transfer active species of type i by monomer of type j.

site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
cid	Component ID of monomer
itpre-exp-f	Preexponential factor

itact-ener-f	Activation energy
itref-temp	Reference temperature

PROPAGATION Reaction identifier for polymer chain propagation on an active species of type i.

site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
cid1	Component ID of active segment
cid2	Component ID of monomer
prpre-exp-f	Preexponential factor
pract-ener-f	Activation energy
prref-temp	Reference temperature

ASSOCIATION Reaction identifier for polymer association with active species of type i.

site-id	Site type identifier for active species formed (i = 1, 2, ..., NSITE)
cid	Component ID of active segment
aspre-exp-f	Preexponential factor for forward reaction (forming aggregate polymer)
asact-ener-f	Activation energy for forward reaction
aspre-exp-r	Preexponential factor for reverse reaction
asact-ener-r	Activation energy for reverse reaction
asasso-no	Polymer association
asref-temp	Reference temperature

EXCH-GENERAL Reaction identifier for general exchange reaction between two growing polymer chains with unique active species and end segments attached to them.

rxn id	Reaction ID for a unique rate constant
site-id1	Site type identifier for first active species (i = 1, 2, ... , NSITE)
cid1	Component ID of active segment on siteid1

site-id2	Site type identifier for second active species (i = 1, 2, ... , NSITE)
cid2	Component ID of active segment on siteid2
egpre-exp-f	Preexponential factor
egact-ener-f	Activation energy
egref-temp	Reference temperature

EXCH-AGENT — Reaction identifier for exchange between growing i polymer species with k segment attached to it and an exchange-agent of type m.

rxn id	Reaction ID for a unique rate constant
site-id1	Site type identifier for first active species (i = 1, 2, ... , NSITE)
cid1	Component ID of active segment on siteid1
site-id2	Site type identifier for second active species (i = 1, 2, ... , NSITE) formed
cid2	Component ID of exchange agent
eapre-exp-f	Preexponential factor for forward reaction
eaact-ener-f	Activation energy for forward reaction
eapre-exp-r	Preexponential factor for reverse reaction
eaact-ener-r	Activation energy for reverse reaction
eacoefd	0 if Po is absent. 1 if Po is present
earef-temp	Reference temperature

EQUILIB-CION — Reaction identifier for equilibrium with counter-ion between i and j active species with kth segment attached to it.

site-id1	Site type identifier for first active species (i = 1, 2, ... , NSITE)
cid	Component ID of active segment
site-id2	Site type identifier for second active species (j = 1, 2, ... , NSITE)
eqpre-exp-f	Preexponential factor for forward reaction
eqact-ener-f	Activation energy for forward reaction

	eqpre-exp-r	Preexponential factor for reverse reaction
	eqact-ener-r	Activation energy for reverse reaction
	eqcoefd	0 if counter-ion is absent. 1 if counter-ion is present
	eqref-temp	Reference temperature
CHAT-SPON	Reaction identifier for spontaneous chain transfer on active species of type i.	
	site-id	Site type identifier for active species (i=1, 2, ... , NSITE)
	cid	Component ID of active segment
	cspre-exp-f	Preexponential factor
	csact-ener-f	Activation energy
	csref-temp	Reference temperature
CHAT-MONOMER	Reaction identifier for chain transfer to monomer of type j on active species of type i.	
	site-id	Site type identifier for active species (i=1, 2, ... , NSITE)
	cid1	Component ID of active segment
	cid2	Component ID of monomer
	cmpre-exp-f	Preexponential factor
	cmact-ener-f	Activation energy
	cmref-temp	Reference temperature
CHAT-DORM-P	Reaction identifier for chain transfer on active species of type i to form dormant polymer of type j.	
	rxn id	Reaction ID for a unique rate constant
	site-id1	Site type identifier for growing active species (i = 1, 2, ... , NSITE)
	cid1	Component ID of active segment on siteid1
	site-id2	Site type identifier for product active species (j = 1, 2, ... , NSITE) formed
	cid2	Component ID of monomer

	cdpre-exp-f	Preexponential factor
	cdact-ener-f	Activation energy
	cdref-temp	Reference temperature
CHAT-AGENT	Reaction identifier for chain transfer to chain transfer agent on active species of type i.	
	site-id	Site type identifier for active species (i=1, 2, ... , NSITE)
	cid1	Component ID of active segment
	cid2	Component ID of chain transfer agent
	capre-exp-f	Preexponential factor
	caact-ener-f	Activation energy
	caorder	Reaction order for chain transfer agent concentration
	caref-temp	Reference temperature
TERM-C-ION	Reaction identifier for chain termination with counter-ion.	
	site-id	Site type identifier for active species (i=1, 2, ... , NSITE)
	cid	Component ID of active segment
	tcpre-exp	Preexponential factor
	tcact-energy	Activation energy
	tcoefb	0 if counter-ion does not participate in the reaction. 1 if it does
	tcref-temp	Reference temperature
TERM-AGENT	Reaction identifier for termination with terminating agent.	
	site-id	Site type identifier (i = 1, 2, ... , NSITE)
	cid1	Component ID of active agent
	cid2	Component ID of terminating agent
	tapre-exp-f	Preexponential factor
	taact-ener-f	Activation energy

taorder	Reaction order for terminating agent concentration
taref-temp	Reference temperature

Input Language Example

```
REACTIONS R-ION IONIC
    DESCRIPTION "IONIC KINETIC SCHEME"
    SPECIES ASSOC-INIT=BULI-3 INITIATOR=BULI-1 POLYMER=PS
    MON-RSEG STYRENE STY-SEG
    INIT-DISSOC    BULI-3  BULI-1   1.0       0.  2.715E10  0. 3
    CHAIN-INI-2  1 BULI-1  STYRENE  0.2055E-2 0.  1
    CHAIN-INI-2  2 BULI-1  STYRENE  0.2055E0  0.  0
    PROPAGATION  1 STY-SEG STYRENE  0.2055    0.
    ASSOCIATION  2 STY-SEG          149.7     0   1.0       0  2
    EQUILIB-CION 2 STY-SEG 1        1E-3      0.  1.0       0. 1
```

SPECIFYING SEGMENT-BASED POLYMER MODIFICATION REACTIONS

The input language for the SEGMENT-BAS REACTIONS paragraph is described here.

Input Language

```
REACTIONS    rxnid    SEGMENT-BAS
  DESCRIPTION   '...'
  PARAM  T-REFERENCE=value   PHASE=V/L/L1/L2   CBASIS=basis
  SPECIES  POLYMER=polymerid
  STOIC    reactionno    compid    coef  /  ...
  RATE-CON  reactionno  pre-exp  act-energy  [t-exp]   /  ...
  POWLAW-EXP    reactionno    compid    exponent  /
```

The keywords for specifying rate constant parameters, reaction stoichiometry, reacting polymer, are described below.

Input Language Description

reacid	Unique paragraph ID.	
DESCRIPTION	Up to 64 characters between double quotes.	
PARAM	Used to enter reaction specifications.	
	T-REFERENCE=*value*	Reference temperature. If no reference temperature is given, the term 1/Tref is dropped from the rate expression shown in Equation 4.48.
	PHASE=*V/L/L1/L2*	Reacting phase
	CBASIS	Basis for power law rate expression. Choices are as follows MOLARITY MOLALITY MOLEFRAC MASSFRAC MASSCONC
SPECIES	Used to specify reacting polymer.	
	POLYMER=*polymerid*	Polymer component ID (for reacting polymer)

STOIC	Used to specify stoichiometry for user reactions.	
	reactionno	Reaction number
	compid	Component ID
	coef	Stoichiometric coefficient (negative for reactants and positive for products)
RATE-CON	Used to specify rate constant parameters.	
	reactionno	Reaction number
	pre-exp	Pre-exponential factor in SI units
	act-energy	Activation energy in mole enthalpy units
	t-exp	Temperature exponent
POWLAW-EXP	Used to specify power-law exponents.	
	reactionno	Reaction number
	compid	Component ID
	exponent	Power law exponent

Input Language Example

```
REACTIONS R-1 SEGMENT-BAS
   PARAM CBASIS=MOLEFRAC
   SPECIES POLYMER=PVAC
   STOIC 1 VAC-SEG -1 / H2O -1 / VOH-SEG 1 / ACETIC 1
   RATE-CON 1 .001
   POWLAW-EXP 1 VAC-SEG 1 / H2O 1 / VOH-SEG 0 / ACETIC 0
```

REFERENCES

Aspen Plus Reference Manual, "Physical Property Data," Aspen Technology, Inc. (1998).

Aspen Plus Reference Manual, "User Models," Aspen Technology, Inc. (1998).

GLOSSARY

Activity coefficient	Quantity related to partial molar excess Gibbs energy. Used to quantify liquid phase non-idealities in phase equilibrium calculations.
Addition polymerization	Classification for polymerization reactions which do not produce small molecule byproducts. The repeating units within the polymer have the same structure as the monomers from which they are originated.
Addition reaction	Type of reaction found in step-growth polymerization in which small molecules, including free monomers, dyadic salts, and cyclic monomers and dimers react with the end of a growing polymer molecule. These reactions are responsible for the conversion of the monomers and most of the conversion of functional end groups.
Aggregate polymer	Polymer formed by aggregation of growing anionic polymeric species during ionic polymerization.
Amorphous polymer	Polymer where the chains are not arranged in a particular pattern.
Bimodal distribution	Distribution which has two peaks in its graphical representation.
Bivariate distribution	Distribution which considers two independent parameters simultaneously, e.g. chain size and copolymer composition.
Bulk polymer chain	Term referring to the sum of the live and dead polymer chains.
Bulk polymerization	Chain-growth polymerization carried out in the bulk monomer phase without a solvent.
Cell models	Type of polymer solution thermodynamic models, in which polymer segments and monomer molecules are confined on lattice cells. Compressibility is accounted for through variable cell volume.
Chain growth polymerization	Chain growth polymers are formed through the addition of monomers to an active center (free-radical, ion, or polymer-catalyst bond) in a chain reaction at very fast rate. Several different types of reactions occur to initiate, propagate, and terminate polymer growth. Examples include various polyolefins, polyvinyls, and several copolymers.
Chain-transfer reaction	Reaction that results in the transfer of the active center from a live polymer chain to another molecule, i.e. monomer, solvent, polymer, chain-transfer agent.
Chemical equilibria	Type of equilibria that involves chemical reactions between the components present.

Combination reaction	Type of reaction found in step-growth polymerization which involves the end groups of two polymer molecules. In most systems, combination reactions play an important role in molecular weight growth.
Compressibility	Property of fluids that measures their ability to change their volume or density with changing pressure.
Condensation polymerization	Classification for polymerization reactions resulting in the elimination of a smaller molecule, water for example, through the reaction of bi- or polyfunctional monomers.
Condensation reaction	Type of reaction found in step-growth polymerization and defined as polymerization reaction which produces a small molecule as a by-product. Typically, the condensate is a volatile compound such as water, methanol, acetic acid, or phenol.
Constant pressure heat capacity	Measures the change in enthalpy of a fluid by changing the temperature by one degree, at constant pressure.
Constant volume heat capacity	Measures the change in enthalpy of a fluid by changing the temperature by one degree, at constant volume.
Copolymer	Polymer chain which has two or more types of repeating units. Copolymers can be in a random alternating, block or graft configuration.
Critical micelle concentration (CMC)	Concentration of emulsifier above which micelles are formed in an emulsion polymerization recipe.
Crystalline polymer	Polymer where the chains are arranged in a regular pattern.
Cyclodepolymerization reaction	Type of reaction found in step-growth polymerization which is an intramolecular reaction in which a polymer end group reacts with a segment in the same molecule, forming a ring. The ring-shaped molecule is lost from the linear parent molecule, reducing the molecular weight of the parent.
Dead polymer chain	Term referring to terminated polymer chains that do not have an attached active center. Dead polymer is represented using the symbol (D_n) where the n refers to the chain length in terms of the number of segments or monomer units incorporated in the polymer chain.
Density	Reciprocal of molar volume. It measures the mass of a fluid per unit volume.
Departure function	Enthalpy, entropy, and Gibbs free energy departure or the difference between the actual property (H, S, or G) and the property at ideal-gas conditions.
Devolatilization	Process of removing unreacted monomers and other undesired volatile substances from the polymer, downstream from the reactor.

Diffusion coefficient	Transport property that is related to the mass transfer (diffusion) of one component into a fluid.
Distribution moment	Average property of the full molecular weight distribution.
Dormant polymer	Polymer chain which is dormant during ionic polymerization. It can become active again and start growing.
Electrophile	Reactant with the strongest electrophilic group among the reacting species in step-growth polymerization.
Electrophilic group	Electron-weak group in step-growth polymerization, such as acids (~COOH), esters (~COO~), amides (~CONH~), and isocyanates (~NCO).
Emulsion polymerization	Chain-growth polymerization involving monomers, water-soluble initiators, and surfactants effected in a water-based medium. Resultant product is latex which is a dispersion of polymer particles in water.
End group reformation	Type of reaction found in step-growth polymerization in which one type of end group is converted into another without influencing chain length.
Enthalpy	Energetic quantity defined as the sum of the internal energy and the product of pressure by volume. Used in energy balance calculations of open systems.
Entropy	Thermodynamic quantity used to describe quantitatively the second law of thermodynamics.
Equation of State (EOS)	Thermodynamic relation between temperature, pressure, and volume of fluids and their mixtures. Used to evaluate thermodynamic properties and fluid phase equilibria.
External seed process	Process in emulsion polymerization where a well characterized seed is used as the starting material for emulsion production.
Free radical	Active center in a free-radical polymerization scheme to which monomers add-on to grow the live polymer chain.
Free-volume	See Compressibility.
Fugacity coefficient	Quantity related to chemical potential. Used in phase equilibrium calculations to describe quantitatively the non-idealities of liquid and vapor phases.
Gel effect	Phenomenon of reduced effective termination rates during free-radical polymerization due to increased viscosity of the reaction medium, as a result of the high molecular weight of the produced polymer.

Generalized Flory theory (GF)	Theory developed by Hall and coworkers (See Chapter 3) which extended the Flory estimate for the probability of inserting an r-mer molecule into a sea of r-mers on a lattice. Hall's extended insertion probability to the continuous-space fluid is related to the thermodynamic pressure.
Gibbs energy minimization	Technique used to calculate chemical and phase equilibria which consists of searching for the total minimum of the mixture Gibbs energy for the different phases involved.
Glass transition temperature	Lower-use limit of a rubber and upper-use limit of a thermoplastic material.
Growth stage	Stage II of the emulsion polymerization process in which the polymer particles grow through a steady diffusion of monomer from the monomer droplets to the particles. When the monomer droplets are totally depleted the stage is considered complete.
Hole model	Model for polymer solutions which represents a combination of two approaches for incorporating free-volume effects into the lattice model. In particular, the lattice has vacant sites and, at the same time, the cell volume is variable.
Homogeneous nucleation	Mechanism for particle formation in which radicals produced in the aqueous phase polymerize with dissolved monomer and precipitate out to form precursor particles. The precipitated precursor particles coagulate with each other until a stable particle is formed.
Homopolymer	Polymer chain containing one type of repeating unit which can be mapped into one segment.
Initiation reaction	First step of polymerization where the active centers are created.
In situ seed process	Process in emulsion polymerization where micelles are used for the generation of seeds.
Instantaneous property	Instantaneous rates of reaction for chain-growth polymerization used in the calculation of chain length distribution.
Interfacial	Phenomena that occur at the interface between two fluids in contact.
Intermolecular reactions	Type of reaction found in step-growth polymerization involving two or more molecules.
Internal energy	Energy of a system due to the sum of molecular interactions between its molecules.
Intramolecular reaction	Type of reaction found in step-growth polymerization involving two sites on the same molecule.
Ionic polymerization	Classification of addition polymerization where the propagating species can be an anion or a cation.

Latex	The final product obtained at the end of the emulsion polymerization after the seed stage, growth stage, and monomer finishing stage are complete.
Lattice model	Model of polymer solutions based on the incompressible lattice theory which assume that the lattice is occupied by both chains and vacant lattice sites or holes. Pure chain fluid is therefore treated as a binary mixture between molecules and holes.
Linear polymer	Polymer with repeat units placed one after the other, without forming branches.
Liquid-liquid phase equilibrium (LLE)	State of coexistence of two liquid phases at equilibrium.
Live polymer chain	Term referring to growing polymer chains containing n segments, with an active center attached to a segment of type i, i.e., segment formed from monomer i. Live polymer is represented using the symbol (P_n^i) where the n refers to the chain length in terms of the number of segments or monomer units incorporated in the polymer chain.
Low molecular weight polymer	Polymer formed from a high molecular weight polymer by random scission.
Lower critical solution temperature (LCST)	Temperature above which liquid-liquid phase split occurs in polymer mixtures due to free volume effects.
Melt	State of aggregation of a polymer at temperatures higher than its melting temperature.
Melting temperature	Highest temperature at which polymer crystallites can exist.
Method of moments	Method of calculating polymer properties through distribution moments or averages of the distribution rather than through the full distribution.
Micellar nucleation	Mechanism for particle formation in which micelles act as the site of nucleation.
Micelles	Aggregates of emulsifier molecules.
Monomers	Small molecules. Two or more bond together form a polymer.
Monomer finishing stage	Stage III of the emulsion polymerization process in which the reaction mixture consists of the monomer swollen polymer particles and the aqueous medium. During this stage further polymerization of the monomer in the particles takes place and results in a decrease of the particle size due to the higher density of the polymer compared to the monomer.
Multimodal distribution	Distribution which has several peaks in its graphical representation.
Nucleation period	Time required for particle nucleation to be completed in emulsion polymerization, also called nucleation time. Usually last 10-15 minutes in conventional polymerization systems.

Nucleophile	Reactant with the strongest nucleophilic group between the reacting species in step-growth polymerization.
Nucleophilic groups	Electron-strong groups in step-growth polymerization, typically alcohols (~OH), amines (~NH2), or water.
Nucleophilic addition	Bimolecular reaction mechanism in which electrophile and nucleophile combine to form a new functional group. These reactions are typically irreversible.
Nucleophilic substitution	Bimolecular reaction mechanism in which reactions from a nucleophilic group from one reactant, nucleophile, displace a nucleophilic group in the other reactant, electrophile, resulting in two new products. These reactions tend to be highly reversible.
Oligomer	Small polymer chain containing 20 or less repeating units. Structure and properties of oligomers are fixed.
Perturbation	Any deviation between the properties of the real and the reference fluid.
Perturbed-hard chain theory (PHCT)	Theory developed by Prausnitz and co-workers (See Chapter 3) which represents the first attempt to combine the vast body of knowledge in statistical mechanics into a practical equation of state for industrial applications.
Phase equilibria	Coexistence of two or more homogeneous phases at equilibrium.
Polydispersity	Polymer property that accounts for the fact that a polymer is a multicomponent mixture of species with differing chain lengths, chain compositions, and degrees of branching. The polydispersity represents the width of the full distribution of the polymer.
Polymer	Macromolecule made up of many smaller repeating units providing linear and branched chain structures.
Polymer fractionation	Process during which the low molecular weight fractions of a polydisperse polymer are separated from the high molecular weight fractions. This is often achieved by the addition of an anti-solvent in the polymer solution, which tends to precipitate the high ends of the polymer.
Precipitation polymerization	Polymerization process where the formed polymer precipitates out of the reaction medium.
Propagation reaction	Main step of polymerization where monomers add-on to a growing polymer chain.

Pseudocondensation reaction	Type of reaction found in step-growth polymerization which involves rearrangement of atoms in two different functional groups, resulting in a new functional group. No by-products are produced by pseudocondensation reactions. Pseudocondensation reactions can involve two monomers, a monomer and a polymer end group, or two polymer end groups, and are nucleophilic addition reactions.
Quasi-steady-state approximation (QSSA)	Assumption that live polymer moments attain their steady-state value instantaneously during free-radical polymerization. This assumption makes the system of differential equations much easier to integrate. QSSA is also referred to as stationary-state hypothesis.
Random scission reaction	Type of reaction that occurs during polymerization which involves the spontaneous cleavage of a polymer chain due to thermal degradation.
Rearrangement reaction	Type of reaction found in step-growth polymerization which occur between two polymer molecules, resulting in two new polymer molecules with different molecular weights. These reactions may involve the end group of one molecule and an internal site on another molecule, or they may involve internal sites on both molecules.
Reverse condensation reaction	Type of reaction found in step-growth polymerization where condensate molecules cleave an existing polymer chain, producing two smaller chains. Reverse condensation reactions near the end of a polymer molecule can generate free monomers
Ring-addition reaction	Type of reaction found in step-growth polymerization which involves intermolecular reactions between polymer end groups and cyclic monomers or oligomers. The end group of the polymer links to the cyclic compound, opening the ring and lengthening the chain of the linear molecule.
Ring-opening reaction	Type of reaction found in step-growth polymerization which involves intermolecular reactions between condensate or monomer molecules and cyclic monomers or oligomers. Condensate molecules or monomers react with cyclic compounds, opening the ring structure to produce linear oligomers or cyclic monomers.
Ring-closing reaction	Type of reaction found in step-growth polymerization which involves intramolecular reactions which occur between the two end groups of a linear molecule. Ring-closing reactions which occur between two end groups of a branched or network molecule are referred to here as intramolecular cyclization to differentiate them from reactions which form ring-shaped molecules.
Ring-scission reaction	See Ring opening reactions.
Seed stage	Stage I of the emulsion polymerization process in which particle nucleation is completed. Usually last 10-15 minutes in conventional polymerization systems.
Segments	Repeating structure within a chain that is produced when monomers bond together through polymerization and become polymers.

Semi-crystalline polymer	Polymer which contains regions that are crystalline and regions that are amorphous. The degree of crystallinity determines the fraction of the polymer that is in crystalline form.
Single-site catalyst	Catalyst with only one type of sites. These catalysts produce the same polymer on all their sites.
Solution polymerization	Polymerization carried out in the presence of an inert solvent in which monomers and polymers are dissolved.
Stationary state hypothesis	See Quasi-steady-state approximation.
Statistical associating fluid theory (SAFT)	Theory developed by Gubbins, Radosz and collaborators (See Chapter 3) which uses a reference fluid term to account for hard-sphere repulsions, chain formation, and chemical association or hydrogen-bonding.
Step-growth polymerization	Classification of polymerization reactions in which monomers react with their functional groups, resulting in a polymer chain that continues to grow from both ends as polymerization progresses.
Surface tension	Quantitative index of the tension of the surface layer of a liquid and its tendency to contract to the smallest possible area.
Surface area	Parameter that measures the surface area of chain molecules; used in some equations of state for chain molecules.
Suspension polymerization	Chain-growth polymerization involving monomers suspended as droplets in a continuous phase (usually water) to which an oil-soluble initiator is added.
Tangent sphere model	Model for polymer solutions which treats molecules as a chain of freely-jointed, tangent hard spheres, where segments on a chain are tangent with their neighboring segments, and can rotate freely without overlaps.
Terminal model	Model which uses the assumption the rate of propagation is independent of the polymer chain length, and depends only on the concentration of monomers and active segments on live polymer chains with active segments.
Terminal monomer loss	Type of reaction found in step-growth polymerization involving the loss of a monomer unit at the end of a polymer chain due to thermal degradation mechanisms.
Termination reaction	Last stage of polymerization where the active centers are destroyed and live polymer chains become dead polymer chains.
Thermal conductivity	Transport property related to heat transfer.
Thermophysical properties	Volumetric and calorimetric properties of pure components and their mixtures.

Transport phenomena	Transfer of mass, heat, and momentum at non-equilibrium conditions.
Unimodal distribution	Distribution where the graphical representation mimics the normal bell-shaped curve.
Upper critical solution temperature (UCST)	Temperature below which liquid-liquid phase split occurs in polymer mixtures.
Vapor-liquid equilibrium (VLE)	State of coexistence of a liquid and a vapor phase at equilibrium.
Viscosity	Transport property related to the rheological behavior of a fluid. It is a function of the polymer size, the mixture composition, the temperature, and the shear rate.
Zero-shear viscosity	Viscosity of a fluid at zero shear rate conditions.

INDEX

This index includes entries in the Examples and Applications Case Book for your convenience.

COMMENTS

Name ______________________________

Title ______________________________

Company ______________________________

Address ______________________________

Telephone ______________________________

What did you like most about this User Guide?

What did you like least about this User Guide?

What improvements or changes would you like in this User Guide?

____

____

____

____

____

____

____

What specific topics would you like included in this User Guide?

____

____

____

____

____

____

____

If you found any errors in this User Guide, list the error and page number:

____

____

____

____

____

____

____

Additional comments:

Forward this form to:

Aspen Technology, Inc.
Attn: Polymer Technology Coordinator
Ten Canal Park
Cambridge, Massachusetts 02141
USA
Telefax: +1-617/949-1030