Documentation of 37-Element, Nominal Pressure Tube, Coolant Mixing Model

Jaff Robertson
Engineering Physics Department
McMaster University
September 3, 2002

Abstract

In this report a new subchannel mixing model that is based on an initial model from references [1,2] is documented. The model is intended for use in the fuel channel behaviour code, FACTAR. The new model predicts the steady state, single-phase (vapour) coolant mixing in 37-element bundles for nominal geometry pressure tubes only. The operating conditions on which the model is based are consistent with those found in the early and late blowdown stages following a large break loss of coolant accident (LBLOCA). It is assumed that coolant mixing is a process consisting of three mixing mechanisms; turbulent diffusion mixing, mixing due to diversion flow around appendages, and buoyancy induced mixing. The model makes use of subchannel mass flow fraction correlations developed from the output of the subchannel mixing code ASSERT PV V3.0. The model predicts the coolant mixing from the 1^{st} bundle to the 12^{th} bundle of a channel. The model implicitly predicts the quantity of heat that is lost to the moderator. The model makes use of mean cross-flow mass transfer correlations developed from the output of ASSERT PV V3.0 in order to improve prediction of coolant mixing under buoyancy dominated flows. The new model gives reasonably accurate predictions of subchannel mixing for both convection and buoyancy dominated flows, and also gives reasonably accurate predictions of heat lost to the moderator.

The new model discussed in this report relies heavily on the use of the ASSERT PV V3.0 subchannel mixing code. There is some speculation as to the ability of ASSERT to properly predict coolant mixing under low flow conditions. For this reason, included in this report is a discussion of the applicability of ASSERT PV V3.0 to the prediction of subchannel mixing under low flow conditions. This discussion revolves around some informative input from the developers of ASSERT. Based on this input it has tentatively been concluded that within the boundary operating conditions and geometry specifications examined in this report, ASSERT results are reasonably trustworthy.

Table of Contents

Page

1. Introduction.....2
2. Synopsis of ASSERT PV V3.0 input file.....3
3. Discussion of Boundary Operating Conditions for the Model.....6
4. Control Volume Designation 7
5. Details of the Model.....8
   - 5.1 Control Volume Mass Flow Correlations.....8
   - 5.2 Modelling Heat Transfer and Mixing Mechanisms.....10
     - 5.2.1 Turbulent Mixing Mechanism.....11
       - 5.2.1.1 Rogers and Rosehart Turbulent Diffusion Model....12
     - 5.2.2 Diversion Induced Mixing Mechanism 14
     - 5.2.3 Buoyancy Induced Mixing Mechanism 15
     - 5.2.4 Moderator Heat Loss.....17
6. Numerical Implementation of Model.....19
7. Comparison of Results.....20
8. Validity of using ASSERT under Low Flow Conditions.....20
9. References.....23
10. Figures (1-59)..... 24

Appendices

A Sample ASSERT PV V3.0 Input Card

B New Model FORTRAN Code

C E-mail Correspondence Concerning ASSERT Limitations Under Low Flow

1. Introduction

This report documents a new subchannel mixing code that is intended for use in the Fuel Channel Behaviour Computer Code FACTAR. It is based on an earlier model documented in references [1,2]. The following paragraph is a synopsis of this earlier model, followed by a synopsis of the new model.

In the initial work in references [1,2], subchannel mixing models were developed that predicted the steady state, single-phase (vapour) coolant mixing in 37-element bundles for both nominal and ballooned geometry pressure tubes. The operating conditions on which the models were based are consistent with those found in the early and late blowdown stages following a large break loss of coolant accident (LBLOCA). The models made use of subchannel mass flow fraction correlations developed from the output of the subchannel mixing code ASSERT PV V2.8. The models assumed that total mixing between adjacent subchannels was dependant on three separate mixing mechanisms; turbulent induced mixing, mixing due to flow diversion around subchannel appendages, and buoyancy induced mixing. The turbulent diffusion mechanism used was the Petrunik single-phase turbulent diffusion correlation. The diversion mechanism was approximated as a diffusion process dependant on the total axial flow in adjacent control volumes. The buoyancy mechanism was approximated as a diffusion process dependant on an overall flow Richardson number. Given the inlet enthalpy distribution to the seventh bundle, the models predicted subchannel mixing from the 7^{th} to the 12^{th} bundles of a channel (the fully developed flow region). The models required that the quantity of heat lost to the moderator be explicitly given in the code.

The new model builds on this initial model, and contains the following main differences;

a) The new model allows for nominal pressure tube geometry only.
b) The new model uses the Rogers and Rosehart turbulent diffusion correlation, found in reference [3].
c) The new model assumes that buoyancy mixing is a mean cross-flow process as opposed to a diffusion process. The quantity of buoyancy cross-flow is given by correlations developed from ASSERT PV V3.0 output.
d) The new model predicts subchannel mixing along the entire bundle length (from the 1^{st} to the 12^{th} bundle)
e) The new model implicitly predicts the quantity of heat lost to the moderator.

The steps taken in order to develop and test the new mixing model are as follows:

a) Choose the appropriate options in an ASSERT PV V3.0 input file for the intended geometry and channel flow conditions.
b) Define the boundary operating conditions under which the new model must perform, and run numerous ASSERT simulations within these boundary operating conditions.
c) Choose appropriate bundle control volume designations based on the results of the ASSERT simulations.
d) Develop correlations to predict the distribution of coolant mass flows in the newly defined control volumes for the entire channel length.
e) Code in the turbulent mixing mechanism and the diversion flow mixing mechanism.
f) Code in a moderator heat loss model.
g) Code in the mean cross flow buoyancy mixing mechanism.
h) Compare the new model's output to the ASSERT PV V3.0 output.

Each of these steps is discussed in detail in the remainder of the report.

2. Synopsis of ASSERT PV V3.0 Input File

ASSERT PV V3.0 is a transient, quasi-two-fluid computer code developed to predict the heat transfer and fluid flow between CANDU fuel bundle subchannels. The ASSERT PV V3.0 theory manual is found in reference [5]. A subchannel is defined as the flow region bounded by fuel rods, and the imaginary lines that join the fuel rod centres. A gap is defined as the imaginary boundary dividing two subchannels. Figure 1 shows the subchannel designation for a 37-element bundle. The fluid within a subchannel is modelled as 1-dimensional. Mixing between adjacent subchannels accounts for a second and third dimension, creating a quasi 3-dimensional model. The ASSERT code solves the conservation equations for mass, momentum, and energy for the mixture, as well as separate energy equations for the vapour and liquid phases in each subchannel. For the present work, only single-phase mixing was allowed in the ASSERT simulations. The present work involved creating an ASSERT input card, and then varying the channel operating conditions. The ASSERT input card was created with the aid of the ASSERT User's Manuel, reference [4]. The four operating conditions that were varied were the channel outlet pressure (MPa), the channel inlet enthalpy (kJ/kg), the channel total mass flow rate (kg/s), and the channel power (MW). The ASSERT input card

written for this work is given in Appendix A. Input card groups of sufficient importance to mention in this section are groups 1, 2, 3, 4, 7, 10, 11 and 16, where;

Group 1 contains the fluid property options

Group 2 contains the friction factor options

Group 3 contains the axial heat flux distribution options

Group 4 contains the subchannel layout and dimension data

Group 7 contains the appendage form loss data

Group 10 contains the mixing model and parameter options

Group 11 contains the channel operating conditions

Group 16 contains the pressure tube heat transfer options

In card group 1, heavy water was chosen as the working fluid, and a heavy-light water property (HLWP) package was chosen for property calculations.

In card group 2, both laminar (f=64/Re) and turbulent (the Colebrook-White formula [5]) friction factors were specified. A single-phase, turbulent wall heat transfer coefficient was specified using the Dittus-Boelter correlation. It was learned that a single-phase, laminar wall heat transfer coefficient is implicitly defined in the ASSERT code. The actual Nu number for wall heat transfer is given by the maximum value between these two correlations.

$$Nu = MAX\begin{Bmatrix} 0.023\,\mathrm{Re}^{0.8}\,\mathrm{Pr}^{0.4} \\ 7.86 \end{Bmatrix} \quad \text{where,} \quad Nu = \frac{hD_{hyd}}{k}$$

h is the wall heat transfer coefficient (W/m^2K).

k is the conduction coefficient of the fluid (W/mK).

D_{hyd} is the hydraulic diameter of the subchannel (m).

Also, the inclusion of gravitational forces was specified in card group 2.

In card group 3, the axial heat flux distribution was defined as a cosine distribution with the axial heat flux-to-average heat flux ratio never exceeding 1.6.

In card group 4, the subchannel geometry and power distribution data was defined as given in Table 1.

Bundle Data	Dimension Length cm
CT Inside Diameter	11.38
PT Inside Diameter	10.338
Bearing Pad Height	0.12
Bundle Diameter	10.212
ΔY Eccentricity	0.063
ΔX Eccentricity	0.0
# Bundles	12

Ring Data	Number of Rods #	Ring Diameter cm	Offset Angle °	Rod Diameter cm	Radial Flux Dist'n
Centre Rod	1	0.	0.	1.312	0.784
Inner Ring	6	2.980	30.	1.312	0.815
Intermediate Ring	12	5.760	15.	1.312	0.917
Outer Ring	18	8.660	10.	1.312	1.129

Table 1: Subchannel Geometry and Power Distribution Data

An option to apply horizontal symmetry in the bundle was invoked in card group 4.

In card group 7, Idelchik's orifice formula was chosen, and the form loss k-factor data was given for the following five types of appendages; bundle endplate, junction gap, bearing pad plane, midplane spacer pads, and midplane bearing pads.

In card group 10, the Rogers and Rosehart turbulent diffusion mixing model was chosen.

In card group 11, the channel operating conditions were specified, and the inlet mass flow fractions of each subchannel were set as proportional to the coinciding subchannel cross-sectional areas.

In card group 16, the moderator temperature was set constant at 79.9°C, and the heat transfer coefficient to the moderator was set at 1000 W/m^2K. The pressure tube and calandria tube were modelled as Zirconium - 2.5% Niobium. The PT/CT gap was modelled as CO_2 gas.

At this point some brief discussion of the choice of the Rogers and Rosehart turbulent diffusion mixing model should be made. In card group 10, there are 6 options available to the user for the turbulent diffusion mixing model. The first option is to use no turbulent diffusion mixing at all. The next four options are variations of the Carlucci turbulent diffusion model [5] (which are more intended for predicting two-phase turbulent diffusion). The last option is the Rogers and Rosehart model. The Rogers and Rosehart model is the only available model intended for use in predicting single-phase turbulent diffusion. Turbulent diffusion and the Rogers and Rosehart model are discussed in detail in section 5.2.1.

3. Discussion of Boundary Operating Conditions for the Model

The model documented in this report is intended for use in predicting single-phase vapour mixing during the early and late blowdown stages following a LBLOCA. Channel conditions that typify these stages are given in Table 2.

Boundary Condition	Units	Initial (t=0) Condition	Early Blowdown Period	Late Blowdown Period
bundle geometry		nominal 37-element	same as initial	same as initial
pressure tube geometry		nominal Bruce (uncrept, unballooned)	same as initial	same as initial
bundle placement		eccentric in pressure tube	same as initial	same as initial
k-factors		non-uniform	same as initial	same as initial
Initial channel power	MW	3.2 to 7.2	NA	NA
Channel Thermal Power	%	100 of initial	25 to 8 of initial	8 to 3 of initial
Axial Flux Shape		cosine, max 1.6 times average	same as initial	same as initial
radial flux shape		(.784, .815, .917, 1.129, inside to outside)	same as initial	same as initial
Pressure at outlet	Mpa	9.5	5 to 4	4 to 0.25
Mass flow at inlet	kg/s	25	2 to 0	2 to 0
enthalpy at inlet	kJ/kg	1100	saturated dry steam up to 4000	saturated dry steam up to 4000

Table 2: Channel Conditions Typical of Early and Late Blowdown Stages Following a LBLOCA (taken from reference [8])

The channel thermal power could reach as high as 25% of the initial channel power which could have been as high as 7.2MW. This means that the channel power to be simulated could be as high as 1.8MW. The channel thermal power can reach as low as 3% of the initial channel power, which could have been as low as 3.2MW. This means that the channel power to be simulated could be as low as 0.1MW, or 100kW.

The channel mass flow could vary between 0 (stagnation) and 2 kg/s. There is a limit to the ability of ASSERT to converge as mass flow rates decrease. It was found that convergence was no longer possible at flow rates below approximately 0.01kg/s (using a nominal geometry pressure tube). There exist other limits to the abilities of ASSERT under low flow conditions. These are discussed in section 8.0.

In the choosing of boundary operating conditions for the model, one important parameter was found to be the ratio of channel power (MW) to channel mass flow (kg/s). It was generally found that ASSERT simulations using ratios of much more than 2.0 (MJ/kg) (with an inlet enthalpy of 2800kJ/kg) predicted subchannel enthalpies in excess of 6000kJ/kg. Generally, enthalpies of this magnitude are outside of heavy water property table ranges. Also, it is in this enthalpy range that heavy water is usually predicted to disassociate into Hydrogen and Oxygen. As a result, only ratios of 0.125,

0.5, 1.0 and 2.0 (MJ/kg) were used. Pressure values of 0.25, 1.0, 2.0, 3.0, 4.0 and 5.0 (MPa), and enthalpy values of 2800, 3400 and 4000 (kJ/kg) were used.
All of this is shown graphically (using a logarithmic scale) in Figure 1. There are 31 channel power/mass flow combinations shown in this figure, and each one is associated with the 6 different outlet pressures, resulting in 186 total simulations. These simulations were all run at inlet enthalpies of 2800kJ/kg. It was only necessary to test varying inlet enthalpies under high buoyancy flows. Consequently, 98 new simulations at two different enthalpies (49 at 3400kJ/kg, and 49 at 4000kJ/kg) were performed under buoyancy dominated flow conditions, resulting in 284 total simulations.
The ASSERT output was analyzed and plotted using Excel and Microsoft Visual Basic macros.
Typically, convection dominated flows resulted from channel mass flow rates of more than 0.1 or 0.2 kg/s, and buoyancy dominated flows resulted from channel mass flow rates of less than 0.1or 0.2 kg/s

4. Control Volume Designation

The model documented in this report is intended for use in the Fuel Channel Behaviour Computer Code FACTAR. Presently there are three options for predicting subchannel coolant mixing available to FACTAR. These options are the CHAN total mixing (CTM) model, the FACTAR total mixing model (FTM), and the CHAN partial mixing model (CPM). The CTM model assumes no mixing between the flow annuli along the bundle length and complete mixing at the bundle junctions, resulting in four coolant enthalpy calculations (one for each flow annulus). The FTM assumes complete coolant mixing between all annuli, resulting in one coolant enthalpy calculation. The CPM assumes no mixing between the outer annulus and the remaining (three) inner annuli, and complete mixing at the bundle junctions, resulting in two coolant enthalpy calculations. One of the reasons for employing such simple subchannel mixing schemes is that it is expensive for FACTAR to model many fuel elements and/or subchannels. Therefore, any new model that is intended for use in predicting subchannel mixing for FACTAR must limit the number of control volumes that it assumes.
For the model discussed in this report an attempt was made to limit the number of control volumes while still maintaining a reasonable degree of solution resolution. The model takes the 60 subchannels that exist in a 37-element bundle, and combines them

into 14 control volumes. Figure 2 shows the ASSERT subchannel designation for a 37-element, nominal pressure tube geometry fuel bundle. Figure 3 shows the control volume designation that has been implemented for the new model, for the same 37-element fuel bundle.

The method that was used to arrive at the control volume designation shown in Figure 3 was the following. First, all of the channel outlet enthalpy solutions predicted by ASSERT were examined. Then, a typical convection dominated flow outlet enthalpy distribution was compared with a typical buoyancy dominated flow outlet enthalpy distribution. Then subchannels that were predicted to have relatively similar enthalpies under both convection dominated and buoyancy dominated conditions were grouped together to form a new control volume. Figures 4, and 5 show a typical convection dominated enthalpy distribution. Figures 6, and 7 show a typical buoyancy dominated enthalpy distribution. The 3 vertical lines on figures 4 and 6 indicate the 3 divisions between the 4 flow annuli in a 37-element fuel bundle. Figures 5 and 7 are colour gradient scales of the subchannel enthalpies, blue being the coldest subchannels, and red being the hottest subchannels. As can be seen, convection dominated flows generally predict the bottom of the bundle to be the hottest, while buoyancy dominated flows generally predict the top of the bundle to be the hottest.

NOTE: the control volume designation in Figure 3 could very easily be altered to reduce the number of control volumes (and consequently decrease the cost of operating the code in FACTAR), but it would not be so easy to increase the number of control volumes.

5. Details of the Mixing Model

5.1 Control Volume Massflow Correlations

In ASSERT, the control volume mass flow rates are calculated by solving the mass, momentum and energy conservation equations. This detailed description of the heat transfer and fluid flow and the associated computational complexity is beyond the scope of FACTAR. The approach taken with the model documented in this report is the same as the approach taken with the initial models in references [1,2]. The approach is to use the output from the ASSERT simulations to develop correlations to predict the fraction of the mass flow that exists for each control volume. A typical prediction of control volume flow fractions as given by ASSERT is shown in figure 8. Upon examination of

the ASSERT simulation output, it was observed that the mass flow fractions in each control volume remained relatively constant over the entire channel length for all cases. This finding allowed mixing to be predicted along the entire channel length, instead of just the "fully developed region". It was observed that mass flow fractions were highly dependent on channel mass flow rate and pressure, and only mildly dependent on channel power and enthalpy. For this reason the mass flow fractions within the subchannels were correlated as only being dependent on mass flow and pressure. Figures 9 to 22 show the graphical form of the correlations used in the new model. As can be seen, mass flow fractions are independent of pressure for channel mass flows above approximately 0.15 kg/s, and increasingly more dependant on pressure as mass flow decreases below 0.15 kg/s. This change in behaviour at approximately 0.15 kg/s coincides with the transition from convection to buoyancy dominated flow. This reflects the following facts about mixing. In ASSERT the flow fractions are determined by the momentum equation. When buoyancy forces are high, the buoyancy term becomes a dominant factor in the lateral momentum equation. But buoyancy forces are highly dependent on channel pressure. When convection forces are high, the convection term becomes a dominant factor in the lateral momentum equation. But convection forces are independent of channel pressure. This explains why the mass flow fractions are pressure dependent in buoyancy dominant flows, and are pressure independent in convection dominant flows.

The mass flow fractions for channels mass flows above 0.15 kg/s were all fit by logarithmic equations of the following form:

$$\phi_i = D_i \ln(\dot{m}) + E_i$$

All other flow fractions were fit by 2nd order polynomial equations of the following form:

$$\phi_i = A_i \dot{m}^2 + B_i \dot{m} + C_i$$

The mass flow rate fractions for each of the 14 control volumes are calculated in this manner except for the 11th control volume. To ensure that the sum of all mass flow fractions is 1.0, the mass flow fraction of the 11th control volume is calculated as the difference between 1.0 and the sum of the other mass flow fractions. Linear interpolation was used to calculate control volume flow fractions for channel pressures not exactly equal to the nominal pressures tested (ie 0.25, 1.0, 2.0, 3.0, 4.0 and 5.0 (MPa)).

NOTE: It was also attempted to model only the "fully developed region" (7th to 12th bundles) in the new model (as was done in references [1,2]), and more accurate predictions of mixing were achieved. However, implementing this in the new model would require that the inlet enthalpy distribution to the 7th bundle be known. This would necessitate using ASSERT, which would defeat the entire purpose of the new model. As it stands, reasonably good results can be achieved by assuming constant control volume flow fractions along the entire channel.

5.2 Modelling of Heat Transfer and Mixing Mechanisms

The conservation of energy equation for any one control volume i, in the fuel channel can be expressed as follows:

$$0 = \dot{m}_i h_{i,in} - \dot{m}_i h_{i,out} + \sum_{j=1}^{N(i)} \dot{m}_{i,j}\left(h_j - h_{i,out}\right) + Q_{i,fuel} - Q_{i,\mathrm{mod}}$$

where:

$h_{i,in}$ Enthalpy of the axial coolant entering control volume i (kJ/kg).

$h_{i,out}$ Enthalpy of the axial coolant leaving control volume i (kJ/kg).

h_j Enthalpy of the coolant in the neighbouring control volume j (kJ/kg).

$\dot{m}_i$ Axial mass flow through control volume i (kg/s).

$\dot{m}_{i,j}$ Mass exchange between control volume j and control volume i (kg/s).

$N(i)$ Number of control volumes that surround control volume i.

$Q_{i,fuel}$ Net quantity of heat added to control volume i by the fuel (kW).

$Q_{i,\mathrm{mod}}$ The quantity of heat lost to the moderator from control volume i (kW).

Assume one is trying to solve for the outlet enthalpy of each control volume for the very first axial node. If the outlet pressure, inlet enthalpy, channel mass flow rate and channel power are all given, then the only unknowns in the equation are $h_{i,out}$ and $\dot{m}_{i,j}$ (the quantity of heat lost to the moderator will be discussed later). The variable we want to solve for is $h_{i,out}$. The challenge of modelling subchannel mixing is to properly predict $\dot{m}_{i,j}$ between adjacent control volumes. The model documented in this report specifies three separate mechanisms that contribute to $\dot{m}_{i,j}$. These mechanisms are turbulent

diffusion, diffusion due to diversion flow around appendages, and mean cross-flow due to buoyancy effects. Therefore:

$$\dot{m}_{i,j} = \dot{m}_{t,i,j} + \dot{m}_{d,i,j} + \alpha_{i,j}\dot{m}_{b,i,j}$$

where:

$\dot{m}_{t,i,j}$ The mass flow exchange between control volumes i and j due to turbulent diffusion.

$\dot{m}_{d,i,j}$ The mass flow exchange between control volumes i and j due to diversion flow around appendages.

$\dot{m}_{b,i,j}$ The mass flow transfer between control volumes i and j due to buoyancy.

$\alpha_{i,j}$ The direction of flow factor. If the buoyancy flow between i and j is into control volume i, then it's value is +1.0. If the buoyancy flow between i and j is out of control volume i, then it's value is 0.0.

The first two mechanisms assume that mixing between two adjacent control volumes involves an equal exchange (diffusion) of mass between those control volumes. These are reasonable assumptions because it is generally accepted that both turbulence and diversion flow result in diffusive mixing. The last mechanism assumes that the mixing between two adjacent control volumes involves a net transfer of mass between those control volumes. This is reflective of the actual physics because buoyancy mixing (which is caused by a density gradient) is more likely to result in a transfer of mass (than an exchange of mass) from a dense control volume to a less dense control volume. These three mixing mechanisms are fully documented in the following three sections.

5.2.1 Turbulent Mixing Mechanism

The following background knowledge of turbulent modelling is a summary of ideas taken from reference [6].

In a turbulent flow, the fluid velocities and enthalpies (for non-isothermal flows) become irregular and unpredictable. The usual approach taken to model a turbulent flow is to decompose the quantity of interest into a time-averaged component plus a fluctuating component. In this manner, the instantaneous velocity and enthalpy become the following:

$$v = \bar{v} + v'$$
$$h = \bar{h} + h'$$

where,

$\bar{v}$ and $\bar{h}$ Are the time-averaged or mean values of the velocity and enthalpy.

v' and h' Are the fluctuating components of the velocity and enthalpy.

By substituting these equations into the energy equation and time averaging each term in the equation, an additional term representing turbulent transport will arise. This term, which involves a correlation between v' and h', must be modelled. The approach used in ASSERT-PV to model the transport due to turbulence is the gradient diffusion method:

$$q_t'' = \rho\overline{v'h'} = -\rho\varepsilon_t \left.\frac{\partial \bar{h}}{\partial \psi}\right|_{i,j}$$

where

q_t'' Is the heat transfer per unit area due to turbulent motion.

ρ Is the fluid density.

ε_t Is the turbulent diffusivity.

ψ Represents the coordinate direction joining two adjacent subchannels i and j.

The enthalpy gradient can be written as:

$$\left.\frac{\partial \bar{h}}{\partial \psi}\right|_{i,j} = \frac{h_i - h_j}{z_{ij}}$$

where,

z_{ij} Is the centroidal distance between subchannels i and j.

h_i and h_j Are the time-averaged enthalpies.

The total energy transport due to turbulent fluctuations is obtained by multiplying by the heat transfer area between two adjacent subchannels,

$$A = Lc$$

where,

L is the axial length of a control volume.

c is the width of the gap between subchannels i and j.

$$Q_{ij} = -A\rho\varepsilon_t \frac{h_j - h_i}{z_{ij}} = (Lc)\rho\frac{\varepsilon_t}{z_{ij}}(h_i - h_j)$$

Since the turbulent diffusivity, ε_t, can be expressed as the product of a length scale l and a velocity scale V, the net turbulent heat transfer Q_{ij} can be rewritten as:

$$Q_{ij} = L\omega'_{ij}(h_i - h_j) = \dot{m}_{t,i,j}(h_i - h_j)$$

where:

$$\omega'_{ij} = \rho c \frac{\varepsilon_t}{z_{ij}} = \rho c V \frac{l}{z_{ij}}$$

The quantity ω'_{ij} (kg/ms) is defined as an effective mass flow (per unit control volume length) caused by turbulence. As mentioned in section 2.0, the Rogers and Rosehart [3] correlation option has been chosen for ω'_{ij} in the ASSERT input card. It has also been implemented for the new model documented in this report.

5.2.1.1 Rogers and Rosehart Turbulent Diffusion Model

Through extensive experimentation with ASSERT PV V3.0, it was observed that subchannel mixing solutions are quite sensitive to the turbulent diffusion mixing model implemented (i.e. the correlation for ω'_{ij}). This makes the choice of a turbulent diffusion mixing model quite a critical one. In reference [7], a comparison was made of three relatively well known single-phase, turbulent diffusion mixing models; the Rogers and Rosehart model, the Petrunik model, and the Rehme model. In this work, the author summarizes the assumptions and derivations used to arrive at each correlation, and assesses each model's ability to measure mixing under both air and water conditions. The author's conclusion was that the most appropriate overall model is Rehme's model, but that any of the three models may be used to predict air (or steam) mixing. Because Rehme's model is not available to ASSERT, and also to enable proper comparison to ASSERT results, it was decided that the new model should employ the Rogers and Rosehart model.

The Rogers and Rosehart model:

$$\varpi_{ij} = 0.0058\mu\left(\frac{c}{d}\right)^{-0.46} \mathrm{Re}_{i,j}^{0.9}$$

The Rogers and Rosehart turbulent diffusion model is considered valid for Reynolds numbers of greater than 2000. Therefore, the above correlation will over predict mixing for Reynolds numbers of less than 2000. It was discovered that the model implemented

in ASSERT uses a slightly altered version of the above correlation in order to minimize this over prediction.

ASSERT version of the Rogers and Rosehart model:

$$\varpi_{ij} = 0.0058\left(\frac{c}{d}\right)^{-1.46}(SM_i G_i + SM_j G_j)c$$

$$SM_i = \left(1+\left(\frac{D_{h,j}}{D_{h,i}}\right)^{1.5}\right)\left(\frac{D_{h,i}}{2d\,\mathrm{Re}_i^{0.1}}\right) \quad \text{(And vice verse)}$$

Where,

- c Gap width between the two control volumes (m).
- d Average diameter of fuel pins around the gap (m).
- SM_i ASSERT implemented laminar mixing reduction factor.
- G_i Mass flux (kg/m^{2}s).
- $D_{h,i}$ Hydraulic diameter of control volume i (m).
- Re_i Reynolds number in control volume i.

Because the new model involves control volumes that could contain more than one gap between neighbouring control volumes (in contrast to ASSERT's subchannels), another factor ($N_{i,j}$) has to be applied in order to evaluate $\dot{m}_{t,i,j}$.

$$\dot{m}_{t,i,j} = N_{i,j} L \varpi_{i,j}$$

Where,

- $N_{i,j}$ The number of gaps that separate control volume i and j.
- L The length of an axial node (the length of one fuel bundle, in this case).

5.2.2 Diversion Induced Mixing

Diversion induced mixing is the transfer of energy between adjacent control volumes caused by obstructions (fuel bundle appendages) in the axial flow path. It produces local flows across the gaps between the control volumes. The diversion induced mixing in ASSERT is modelled by applying K-factors (which simulate pressure drops near the appendages), subchannel area changes and subchannel wetted perimeter changes, for each appendage. The effect of the appendages is to produce small, local fluctuations in flow enthalpy, and to increase overall mixing slightly. The local fluctuations can be

considered as unimportant in terms of the solution resolution required for FACTAR. Therefore, only the slight effect of the overall mixing needs to be modelled. The approach taken for the new model is to simply say that the mixing effect of the appendages can be approximated by a diffusion process, and that the degree of mixing that occurs is proportional to the total axial mass flow rate through the adjacent control volumes. Therefore the following correlation was implemented in the new model:

$$\dot{m}_{d,i,j} = k_{i,j}\left(\dot{m}_i + \dot{m}_j\right)$$

Where,

$\dot{m}_i$ Axial mass flow through control volume i.

$\dot{m}_j$ Axial mass flow through control volume j.

$k_{i,j}$ Constant applied to the gap between control volumes i and j.

The method of calculating $k_{i,j}$ was quite simple. A convection dominant simulation was run with ASSERT, the ASSERT coolant enthalpy solution was compared to the new model enthalpy solution, and the value of $k_{i,j}$ was varied until the new model predicted the same degree of mixing as ASSERT. A good value for $k_{i,j}$ turned out to be 0.02.

NOTE: it is believed that the diversion flow mixing model could be improved as follows. Using ASSERT, simulations over a wide range of operating conditions could be performed with both the buoyancy option and the turbulent diffusion option turned off (only appendage induced mixing allowed). Then correlations could be developed from the ASSERT output that predict the value of $\dot{m}_{d,i,j}$ given the operating conditions. As this would be a somewhat time consuming task, time constraints did not allow for this to be performed. However, the new model would benefit from future work in this area.

5.2.3 Buoyancy Induced Mixing

As the axial Reynolds number of the coolant decreases, the degree of buoyancy induced mixing that takes place increases. Ideally, the Richardson number of two control volumes would predict the degree of buoyancy mixing between those two control volumes (where the Richardson number is a measure of the vertical force compared to the horizontal force, on a volume of fluid). The Richardson number shared by two adjacent control volumes can be defined using the combined Grashof and Reynolds numbers of the two control volumes, as follows:

$$Ri = \frac{Gr}{\mathrm{Re}^2}$$

$$Gr = \frac{\overline{\rho}_{i,j} g c^3 \Delta\rho_{i,j}}{\mu^2} \text{ And } \mathrm{Re} = \frac{G_{i,j} D_h}{\mu}$$

Where,

$\overline{\rho}_{i,j}$ The mass weighted average density of control volumes i and j (kg/m^3).

g Acceleration due to gravity (m/s^2).

c Gap width between control volumes i and j (m).

$\Delta\rho_{i,j}$ Difference in density between control volumes i and j (kg/m^3).

μ Average viscosity of control volumes i and j (kg/ms).

$G_{i,j}$ Mass flux through control volumes i and j (kg/m^{2}s)

D_h Combined hydraulic diameter of control volumes i and j (m).

An attempt was made to predict the degree of buoyancy mixing between control volumes by using this Richardson number and by assuming that the mixing was a diffusion process. Unfortunately, this attempt proved unsuccessful. The reason for this eventually became evident, as follows.

Upon further examination of ASSERT results under buoyancy dominated flow conditions, it was observed that there existed a consistent pattern of flow circulation in the fuel bundle, along the entire channel. This result implied that ASSERT models buoyancy mixing as a net mass transfer between control volumes, and not as a mass exchange between control volumes. The quantities of mass transfer through each gap were consistently the same, **relative to one another**. This implied that it was possible to specify the quantity of mass transfer across a gap by using a symbolic mass flow (proportional to the actual mass flow) in combination with some proportionality constant, as follows:

$$\dot{m}_{b,i,j} = \kappa \dot{m}_{symb,i,j}$$

Where,

κ Proportionality constant.

$\dot{m}_{symb,i,j}$ Symbolic mass flow from control volume i to control volume j (kg/s).

This symbolic mass flow is based on the pattern of mass flow circulation that was observed under buoyancy dominated conditions.

The proportionality constant was observed to be dependent on the channel pressure, channel mass flow rate, and channel inlet enthalpy. Numerous simulations were performed with the intent of developing a correlation between the proportionality constant, and the three operating conditions upon which this mass transfer was dependant (channel pressure, channel mass flow rate, and channel inlet enthalpy). Correlations of the following form for a proportionality constant were developed for each of the six tested channel pressures (0.25, 1.0, 2.0, 3.0, 4.0, 5.0 MPa):

$$\kappa = Cx^2 + Dx + E$$

With,

$$x = \left(\frac{Q_{chan}}{\dot{m}_{chan}}\right)^3 + \left(\frac{h_{chan}}{1000}\right)^3$$

Where,

Q_{chan}	Channel power (MW)
$\dot{m}_{chan}$	Channel mass flow rate (kg/s)
h_{chan}	Channel inlet enthalpy (kJ/kg)
C, D, E	Proportionality constant coefficients

The proportionality constant ranges in value from 0 and 1. The prediction of mass transfer due to buoyancy between control volumes is performed for both buoyancy and convection dominated flows. However, for convection dominated flows, the quantity of mass transfer due to buoyancy is insignificant in comparison to the quantity of mass exchange due to turbulent diffusion, and therefore it has no effect on the overall mixing. Figure 23 shows the curves that were used to arrive at the correlations for the proportionality constant. Linear interpolation is used to predict the proportionality constant for channel pressures that are not exactly equal to the nominal pressures (0.25, 1.0, 2.0, 3.0, 4.0, 5.0 MPa).

NOTE it is believed that the buoyancy mixing mechanism could be improved in a manner similar to the potential improvement outlined for the diversion flow mixing mechanism. Multiple simulations could be performed with the turbulent diffusion, and appendage form loss data options turned off (only buoyancy induced mixing allowed). Better correlations for the mass transfer due to buoyancy between adjacent control volumes could be developed. Again this would be a time consuming process, but the new model would benefit from such work.

5.2.4 Moderator Heat Loss

It is generally known that heat loss to the moderator is essentially insignificant for nominal pressure tube geometries in CANDU reactors. However, in section 3.0 it was mentioned that during the late blowdown stage of a LBLOCA, channel powers can reach as low as 0.1MW (100kW). Sufficiently low channel mass flow rates combined with such low channel powers can produce enthalpy solutions that are quite sensitive to the degree of heat lost to the moderator. It was for this reason that a moderator heat loss scheme was implemented in the model. The scheme assumes only radial heat transfer in the walls. This assumption is valid considering the small thickness of the walls (pressure tube, CO^2 gap, and calandria tube) in comparison to the radius of the pressure tube. Using a resistance method, the heat lost to the moderator takes the following form (taken from reference [9]):

$$Q_{i,\mathrm{mod}} = \frac{T_{i,cool} - T_{\mathrm{mod}}}{R_{tot}}$$

$$R_{tot} = \frac{1}{2\pi r_1 L h_{i,cool}} + \sum_{m=1}^{NW} \frac{\ln\left(\frac{r_{m+1}}{r_m}\right)}{2\pi k_m L} + \frac{1}{2\pi r_{NW} L h_{\mathrm{mod}}}$$

Where,

$T_{i,cool}$ Temperature of coolant for control volume i (K).

T_{mod} Temperature of moderator (K).

R_{tot} Total resistance to heat transfer from coolant to moderator (K/W).

$h_{i,cool}$ Convection coefficient for coolant in control volume i (W/m^2K).

h_{mod} Convection coefficient of moderator (W/m^2K).

k_m Conduction coefficient of wall segment m (W/mK).

NW Number of wall segments.

r_m Radius of wall at wall segment m (m).

r_1 Inner radius of Pressure tube (m).

r_{NW} Outer radius of Calandria tube (m).

L Length of one axial node (m).

Of course, only control volumes adjacent to the pressure tube wall can lose heat to the moderator.

Solving for the heat lost to the moderator in each control volume, in each axial node is an iterative process because both $h_{i,cool}$ and k_m are property dependant, as follows:

$$h_{i,cool} = \left(\frac{k_{i,cool}}{D_{h,i}} \right) 0.023 \mathrm{Re}_i^{0.8} \mathrm{Pr}_i^{0.4}$$

(This is simply the Dittus-Boelter correlation for turbulent wall heat transfer)

$k_{i,cool}$ Conduction coefficient of the coolant in control volume i (W/mK).

$D_{h,i}$ Hydraulic Diameter of control volume i (m).

k_m is solved for by using a correlation that is dependant on the wall segment material, and the material temperature. The same correlations for k_m used by ASSERT were implemented for the new model.

When solving for $h_{i,cool}$, there was some confusion as to whether to use unsaturated vapour properties, or saturated liquid properties. This confusion was due to the fact that a liquid film could exist on the inside surface of the pressure tube, depending on the channel conditions. ASSERT assumes a liquid film in all cases, but in many circumstances, it may be more reasonable to assume no liquid film. For this reason, an option exists in the new model code to let the user choose between using unsaturated vapour properties or saturated liquid properties. Convergence of the enthalpy solution in the new model is two to three times better when saturated liquid properties are used, instead of unsaturated vapour properties.

6. Numerical Implementation of Model

The computer code written for the new model can be found in Appendix B. This code simply builds upon the code documented in references [1,2].

The numerical implementation of the model is described in this section. As discussed in section 5.2, the conservation of energy equation for any one control volume can be expressed as follows:

$$0 = \dot{m}_i h_{i,in} - \dot{m}_i h_{i,out} + \sum_{j=1}^{N(i)} \dot{m}_{i,j} \left(h_j - h_{i,out} \right) + Q_{i,fuel} - Q_{i,\mathrm{mod}}$$

Where,

$$\dot{m}_{i,j} = \dot{m}_{t,i,j} + \dot{m}_{d,i,j} + \alpha_{i,j} \dot{m}_{b,i,j}$$

This energy equation can be rearranged to isolate for $h_{i,out}$, as follows:

$$\left(\dot{m}_i + \sum_{j=1}^{N(i)} \dot{m}_{i,j}\right) h_{i,out} = \dot{m}_i h_{i,in} + \sum_{j=1}^{N(i)} \dot{m}_{i,j} h_j + Q_{i,fuel} - Q_{i,\mathrm{mod}}$$

This equation was then solved for each control volume using a simple Point Gauss Seidel (PGS) solver.

The code written for the model consists of three nested loops.

The outer loop is the axial node advancement loop. Since one axial node is equivalent to one bundle, the loop solves the bundle enthalpy for each control volume from bundle #1 to bundle #12.

The middle loop is the property and wall temperature update loop. The coolant properties are solved using the same heavy water property package as is used by ASSERT. The middle loop solves the bundle enthalpy until either a maximum number of iterations is reached, or a minimum residual error is reached.

The inner loop is the PGS solver. It sweeps through the control volumes (from 1 to 14, and then backwards to 1), until either a maximum number of iterations is reached, or a minimum residual error is reached.

The code also solves the enthalpies that would be predicted by the FACTAR Partial Mixing model FPM, which is currently available in FACTAR.

7. Comparison of Results

Figures 24 to 59 show comparisons between ASSERT generated enthalpy solutions, and the new model generated enthalpy solutions. Six simulations were chosen from each of the six different tested channel outlet pressures. Simulations for channel powers lower than 25kW were not included in these results (100kW is the very lower bound of the channel power expected to exist during the late blowdown stage following a LBLOCA).

Better agreement between ASSERT results and the new model results was observed for convection dominant conditions in comparison to buoyancy dominant conditions. However, reasonably good agreement was found for all simulations run within the boundary operating conditions specified in section 3.0.

It is believed that the largest source of difference between ASSERT results and the new model results is the constant flow fraction assumption (discussed in section 5.1). It is

more reasonable to assume constant flow fractions for only the fully developed region, rather than for both the developing and fully developed regions. However this would require that the inlet enthalpy distribution to the fully developed region be known. Since the only way to predict this enthalpy distribution is with ASSERT, constant flow fractions must be assumed over the entire channel length.

8. Validity of using ASSERT under Low Flow Conditions

Some components of the new model rely heavily on the use of the ASSERT PV V3.0 subchannel mixing model. Specifically, the control volume flow fraction calculation and the buoyancy mass transfer calculation are both based on correlations derived solely from ASSERT output. For this reason it is important to be aware of any limitations that may be associated with ASSERT. For instance, there exists some speculation as to the ability of ASSERT to properly predict mixing under low flow conditions. The following discussion is meant only as a source of information about this speculation, rather than a challenge of this speculation. Ultimately, a tentative conclusion is made in this discussion regarding the applicability of ASSERT to predicting low flow mixing for the channel operating conditions discussed in this report.

The source of the above mentioned speculation stems from the assumptions that are made in the ASSERT axial momentum equation. If the axial velocity in subchannel i is 'u', and the lateral velocity from subchannel i to j is 'v', then a comprehensive axial momentum model would contain u^2 terms, uv terms, and v^2 terms. The assumption that is made in the ASSERT model is that the v^2 terms are negligible. This is a perfectly valid assumption for convection dominated flows, but it becomes increasingly less valid as the Richardson number (discussed in section 5.2.3) increases. The developers of ASSERT were contacted by email in an attempt to establish under exactly which conditions ASSERT results could no longer be trusted. The reply was that **'reasonable' results can still be obtained from ASSERT when the lateral velocity (the value of v) is no more than an order of magnitude greater than the axial velocity (the value of u).** Some may argue that when v is an order of magnitude greater than u, the assumption that the v^2 terms are negligible is no longer valid. The developers argue that v may be one order of magnitude larger than u in only a fraction of the subchannels, and therefore the code could still, on the whole, predict reasonably

accurate mixing. The entire email correspondence with the developers of ASSERT is included in Appendix C (read from the bottom of the correspondence to the top).
For each simulation performed with ASSERT, the channel averaged axial velocity in each subchannel was compared with the channel averaged lateral velocity in each gap associated with that subchannel, and ratios of lateral velocity to axial velocity (v/u) were calculated. It seems logical that this ratio (v/u) should increase with the following trends: decreasing mass flow rates, increasing pressure, and increasing channel power to channel mass flow rate ratios. By this logic, the maximum value for this ratio should occur at the highest pressure, the lowest mass flow, and a channel power to channel mass flow rate of 2.0 (which is the highest such ratio tested). This turned out to be true. The highest lateral velocity to axial velocity ratio (v/u) observed for any subchannel in any of the 284 ASSERT simulations was 0.267. The average lateral velocity to axial velocity ratio for this case was 0.142. This value was observed under the following channel operating conditions:

$$P_{chan} = 5.0MPa$$
$$\dot{m}_{chan} = 0.015kg/s$$
$$Q_{chan} = 0.03MW$$
$$h_{i,chan} = 2800kJ/kg$$

In no tested case did the lateral velocity ever come close to one order of magnitude larger than the axial velocity. This implies the tentative conclusion that ASSERT can be trusted to properly predict low flow rate mixing within the boundary operating conditions examined in this report.
Unsurprisingly, the maximum ratio of lateral velocity to axial velocity under convection dominated flows remained quite constant at a value of about 0.014. The average ratio of lateral velocity to axial velocity for these cases was about 0.009.

9. References

1. M.F. Lightstone to N. Wahba, "Draft Documentation of 37-Element Nominal Pressure Tube Coolant Mixing Model", February 1997.
2. M.F. Lightstone to N. Wahba, "Draft Documentation of 37-Element Ballooned Pressure Tube Coolant Mixing Model", September 1997.
3. J.T. Rogers, R.G. Rosehart, "Mixing by turbulent Interchange in Fuel Bundles. Correlations and Inferences", ASME, 72-HT-53, 1973
4. V.C.Frisina, E.K.Zariffeh, G.M.Waddington, N.Hammouda, L.N.Carlucci, P.Pfeiffer, "ASSERT-PV V3R0 User's Manual", FFC-FCT-327, COG-00-270, RC-2558, November 2000.
5. N.Hammouda, L.N.Carlucci, G.M.Waddington, "ASSERT-PV IST V3R0: Software Theory Manual", RC-2545, FFC-FCT-322 (Rev.0), August 2001.
6. M.F.Lightstone, "Intro to Turbulence", Computational Fluid Dynamics Course Notes, McMaster University, April 2002.
7. O.Eiff, "Single-phase turbulent Mixing Models for use in ASSERT-PV", Department of Mechanical Engineering, U of Toronto, September 1996.
8. R.Rock to H.Sills and N.Wahba, "Verification of Boundary Conditions for Subchannel Mixing Study using ASSERT-PV", N-03310 T10 RSOAD, August 1996.
9. F.P.Incropera, D.P.DeWitt, "Introduction to Heat Transfer", 3rd Edition, John Wiley & Sons, 1996

Figures 1-59
Are included in the following 59 pages

Figure #1
Boundary Operating Conditions

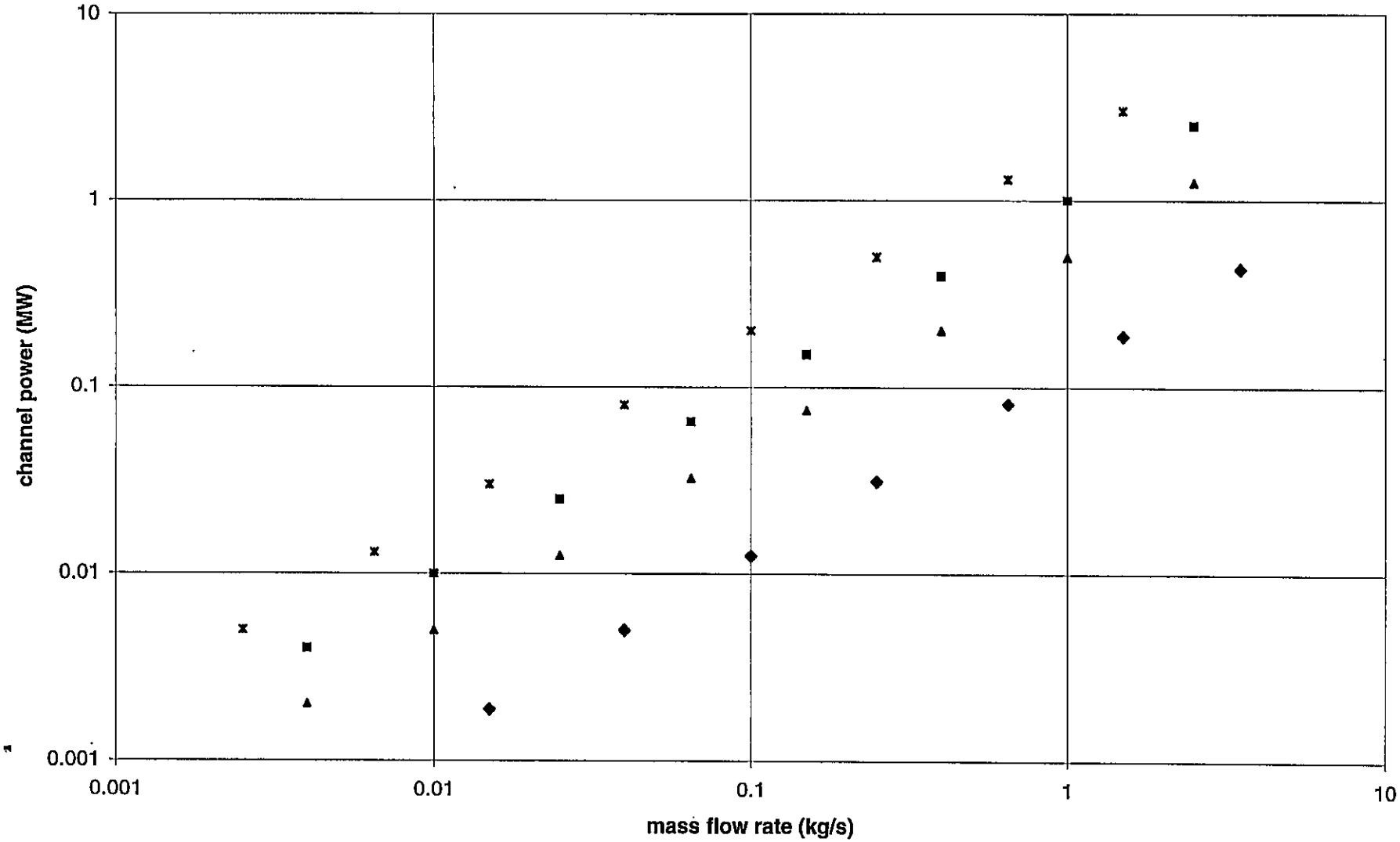


Figure #2
ASSERT Subchannel Designations

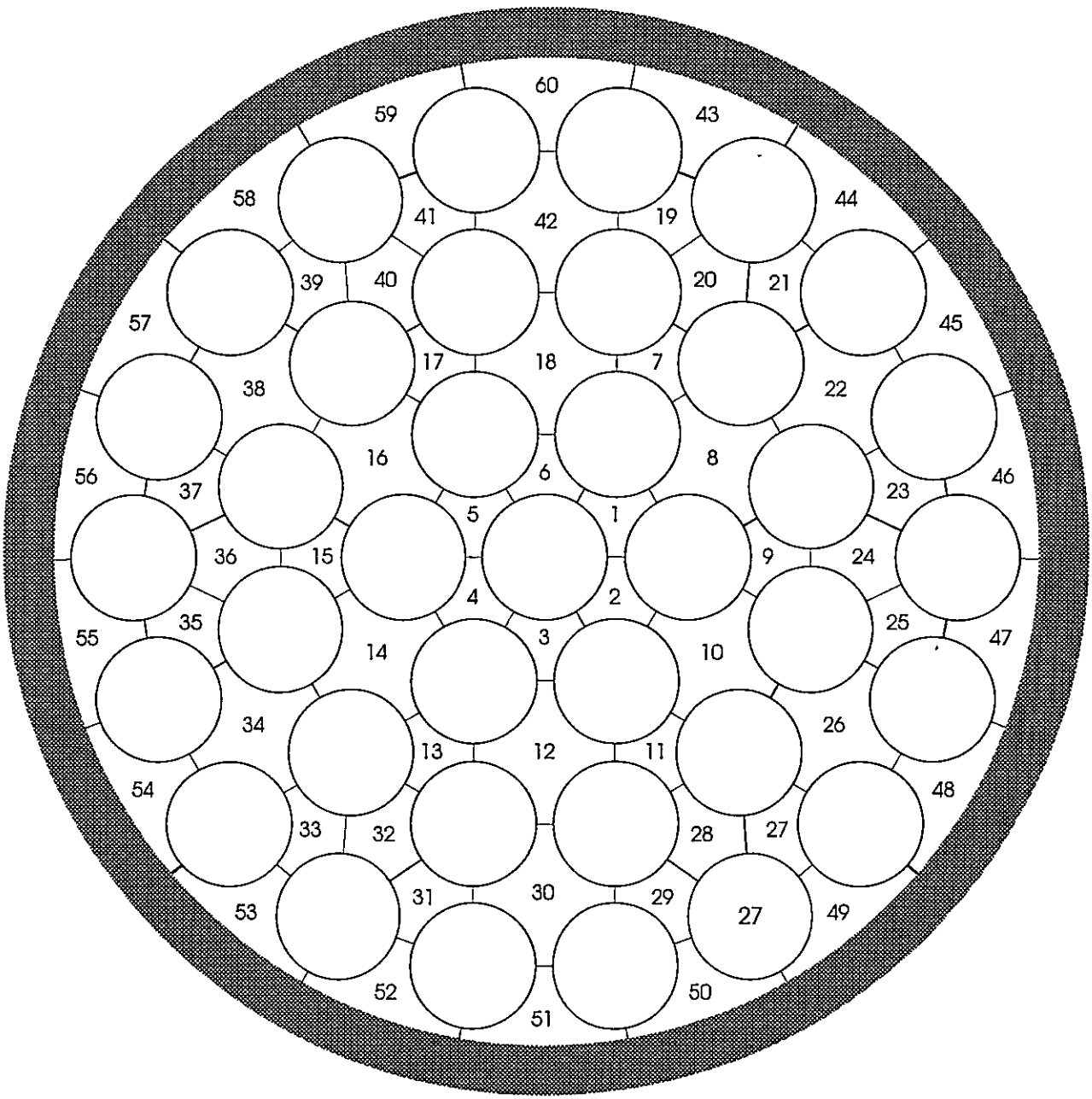


Figure #3
New Model Subchannel Designation

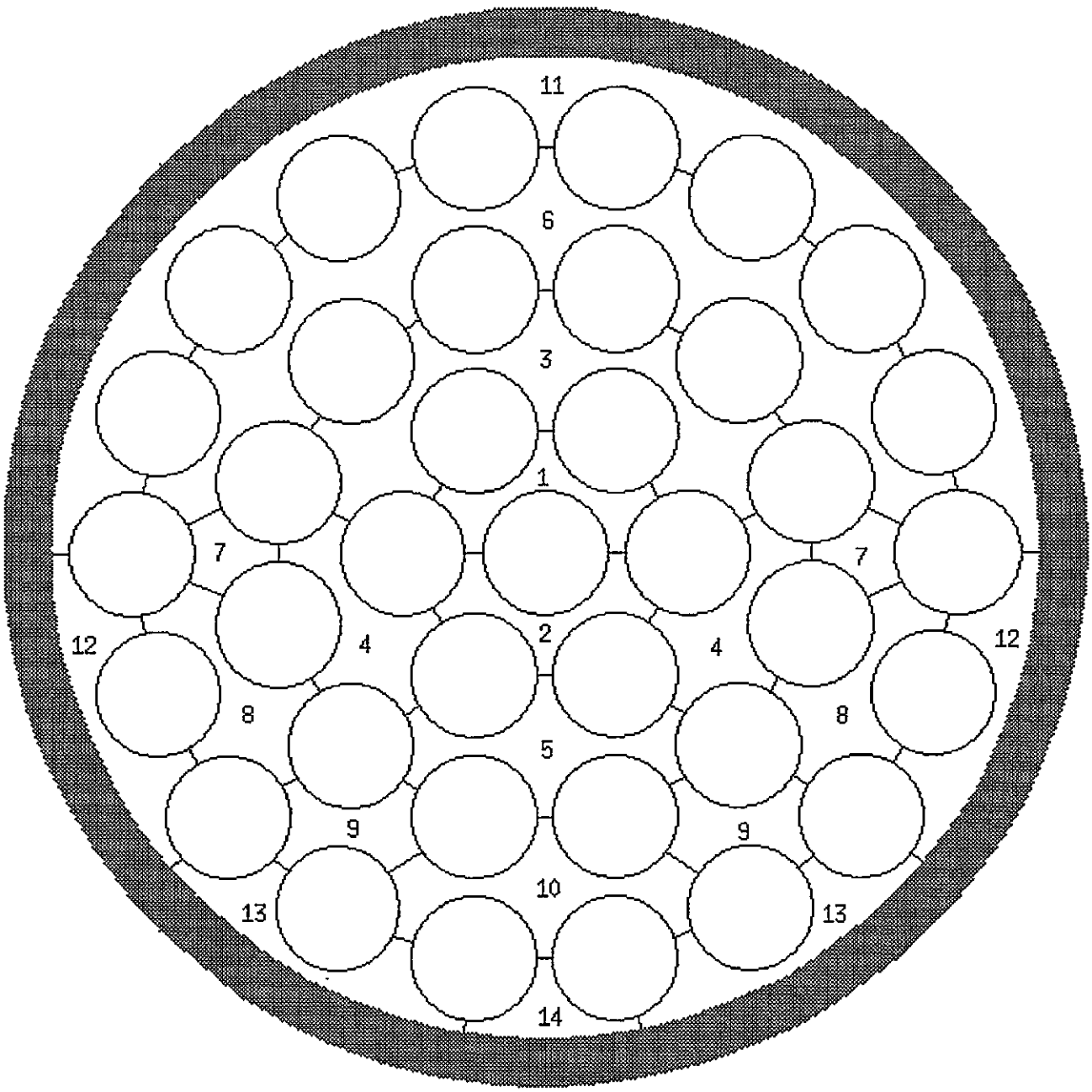


Figure #4
Outlet Enthalpy for
Typical Convection Dominant Flow

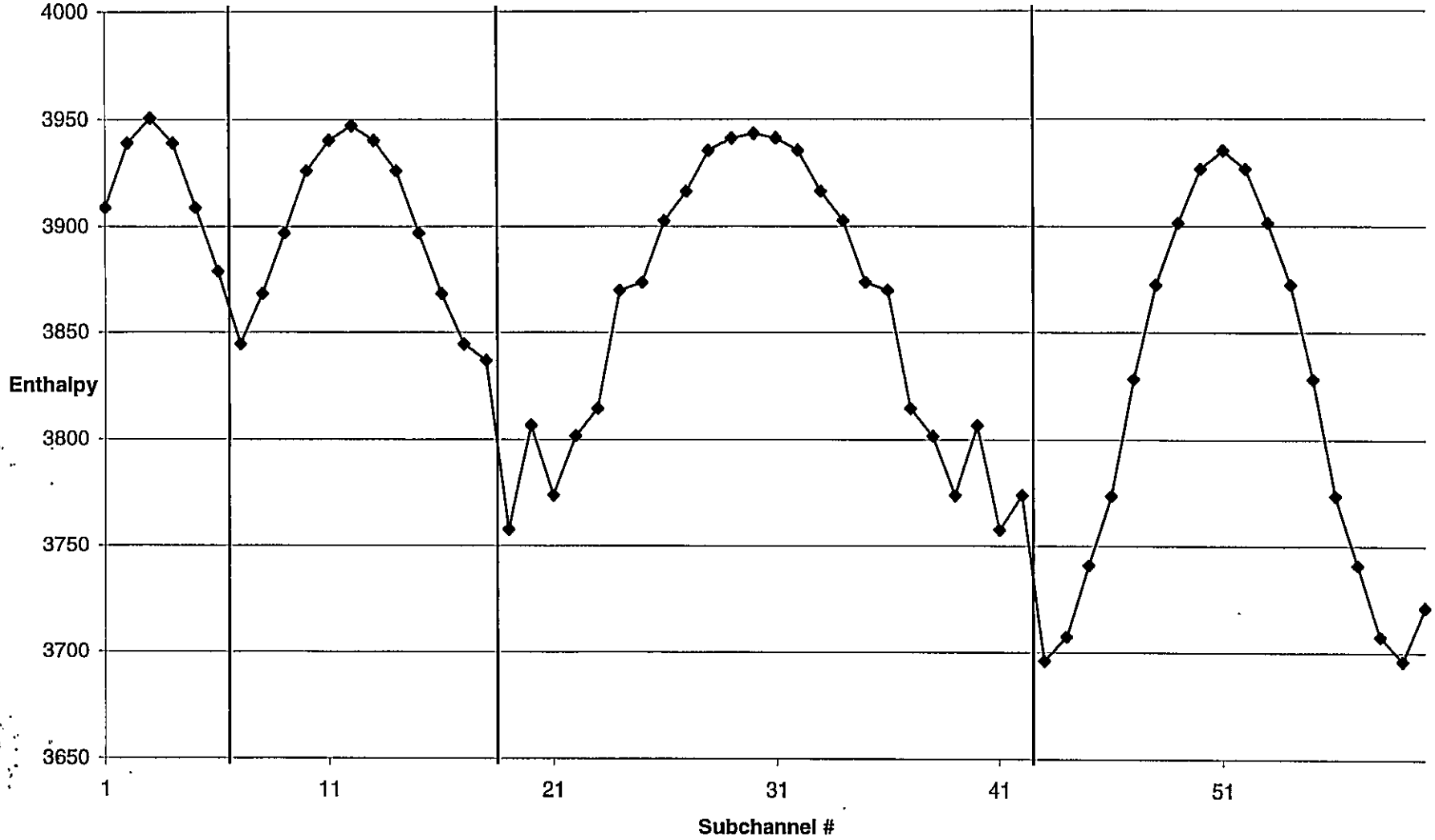


Figure #5 Outlet Enthalpy for Typical Convection Dominant Flow

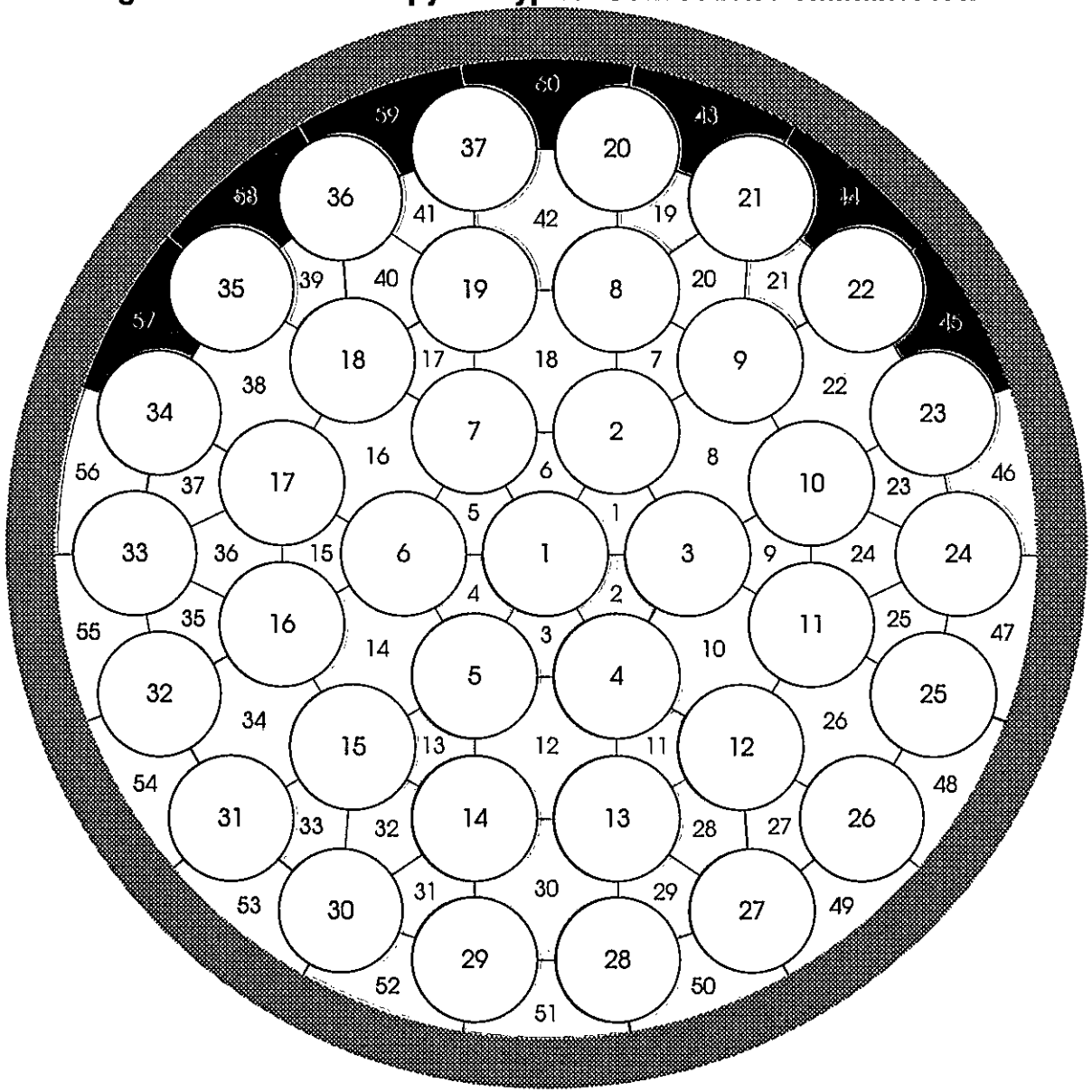


Figure #6
Outlet Enthalpy for
Typical Buoyancy Dominated Flow

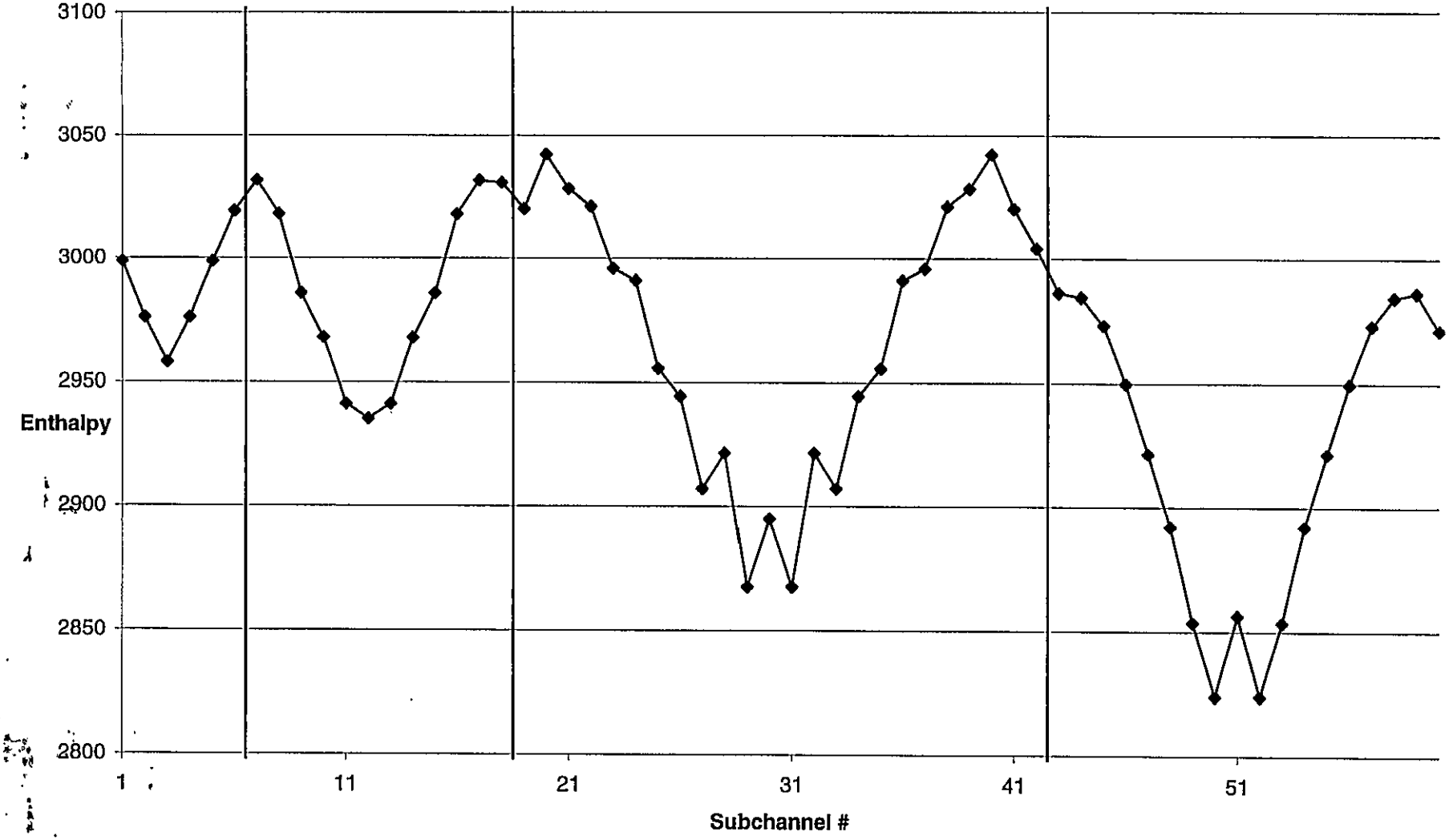


Figure #7: Outlet Enthalpy for Typical Buoyancy Dominated Flow

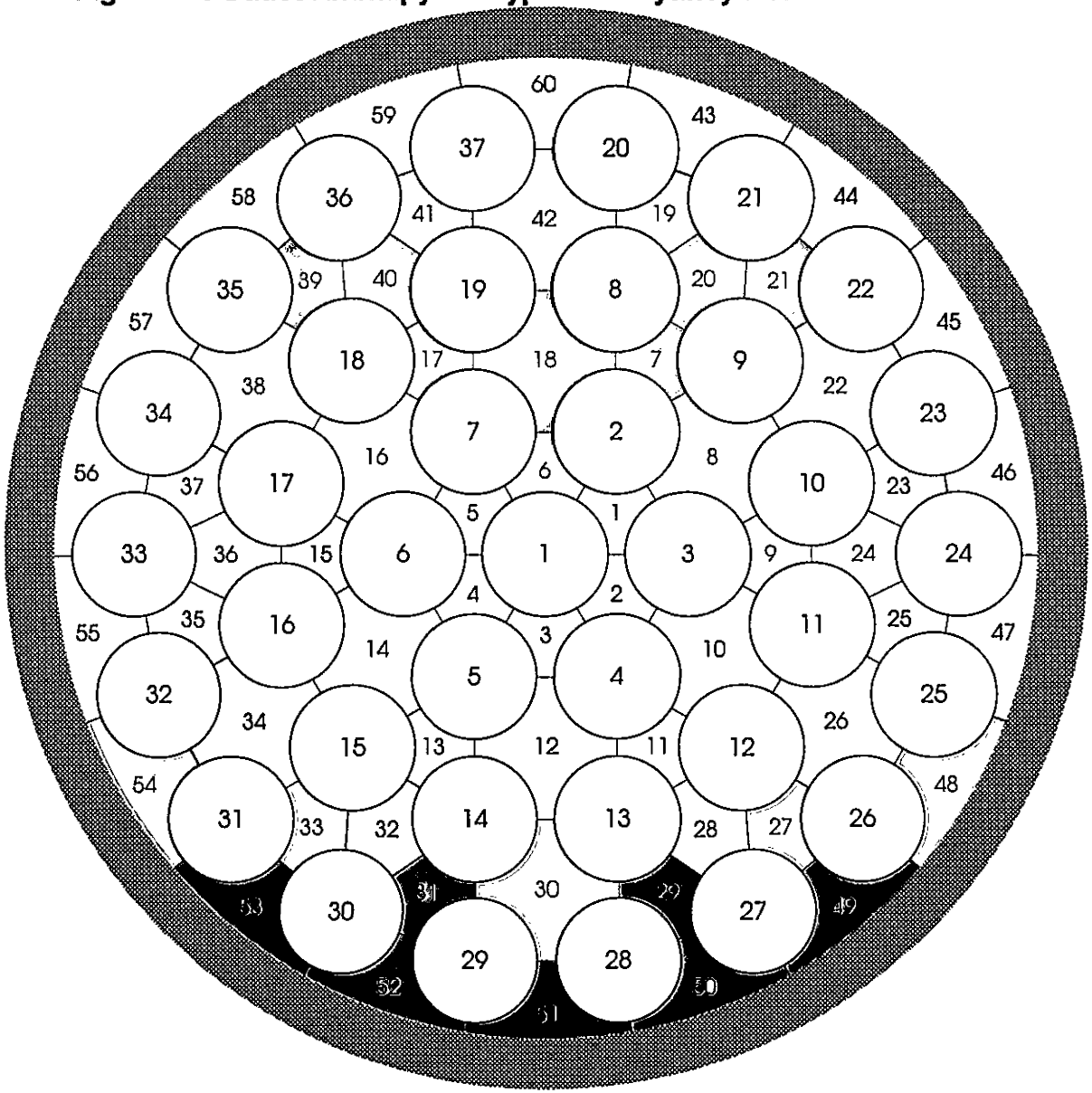


Figure #8
Typical CV Flow Fractions

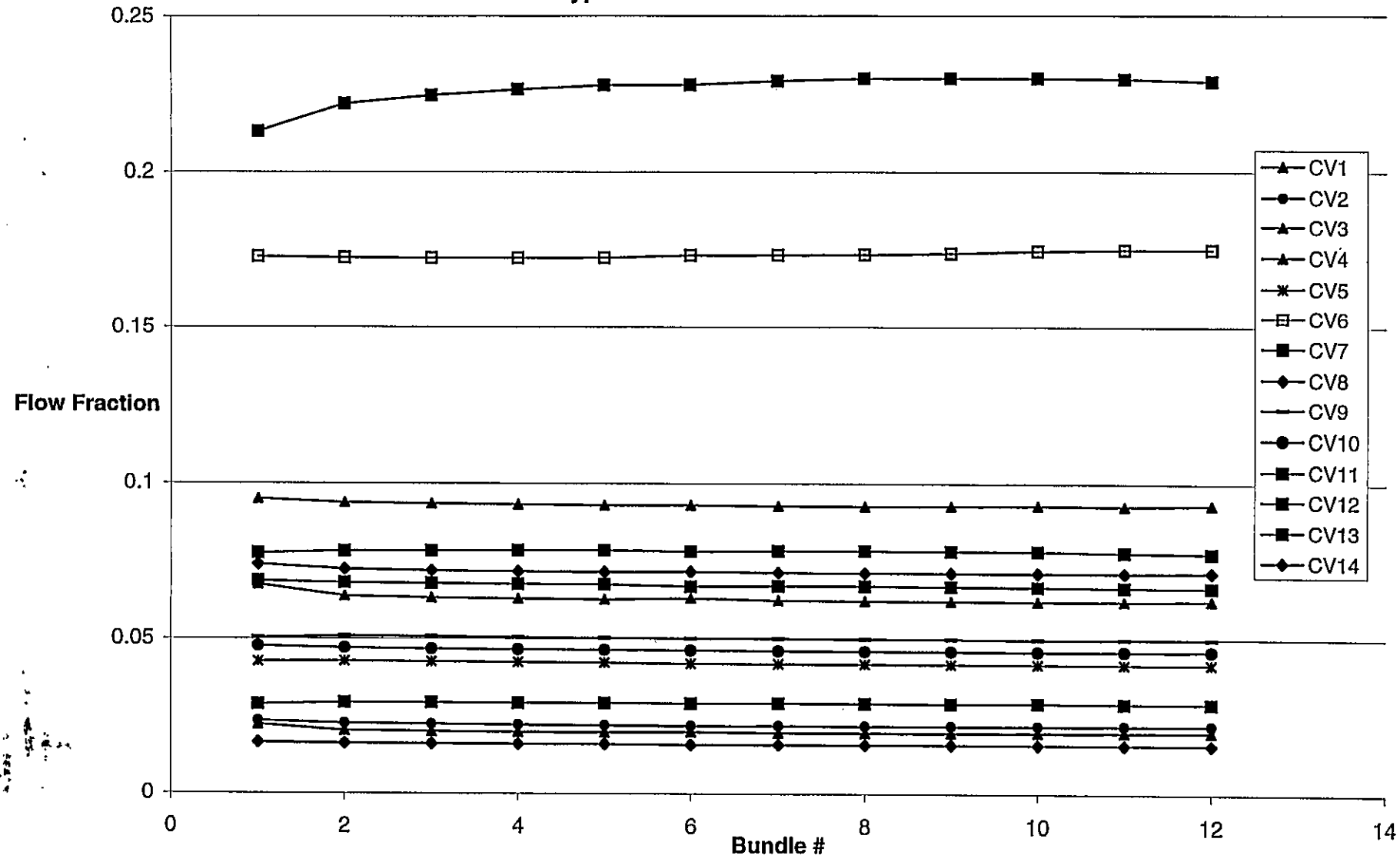


Figure #9
Flow Fractions for CV #1

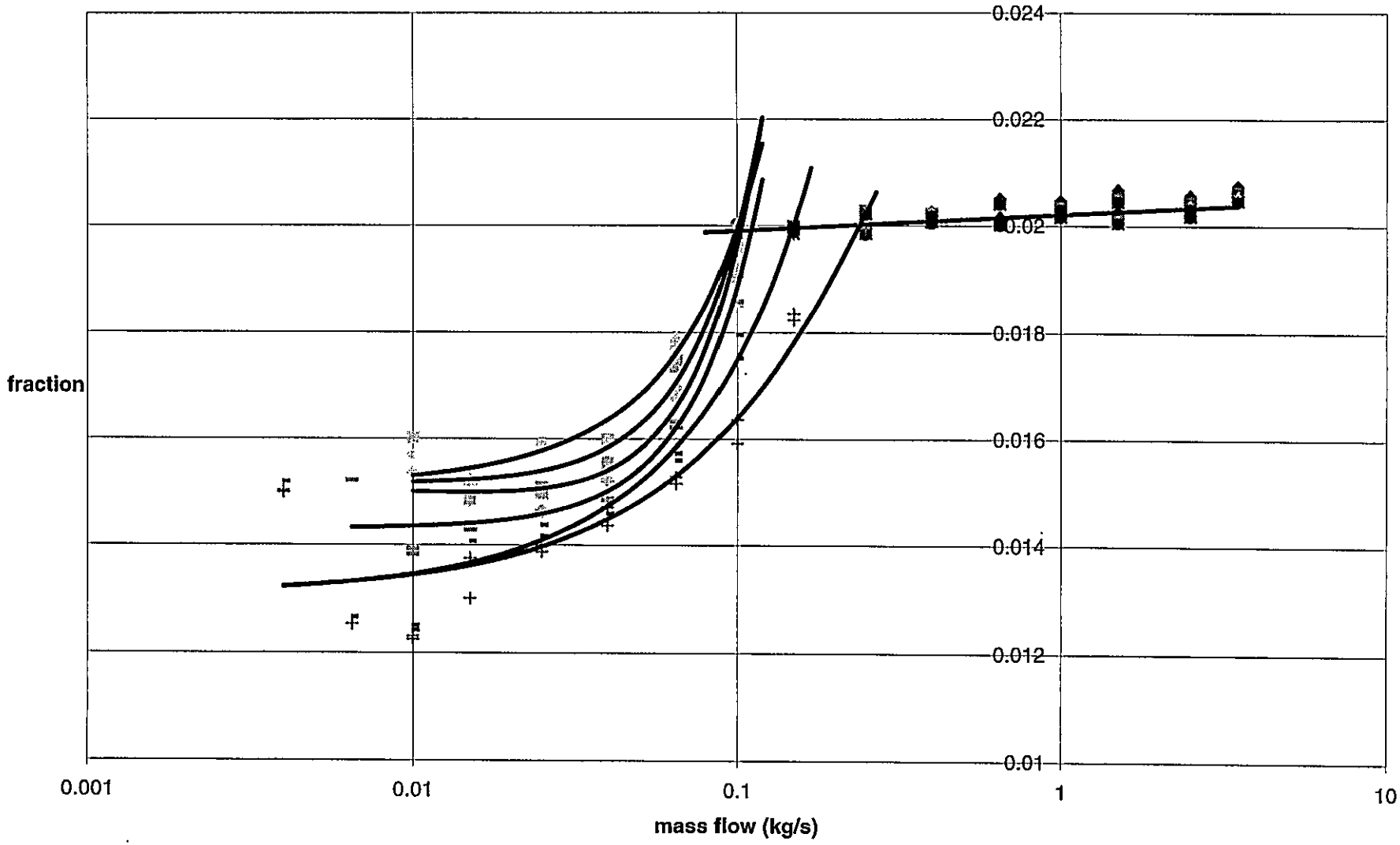


Figure #10
Flow Fractions for CV #2

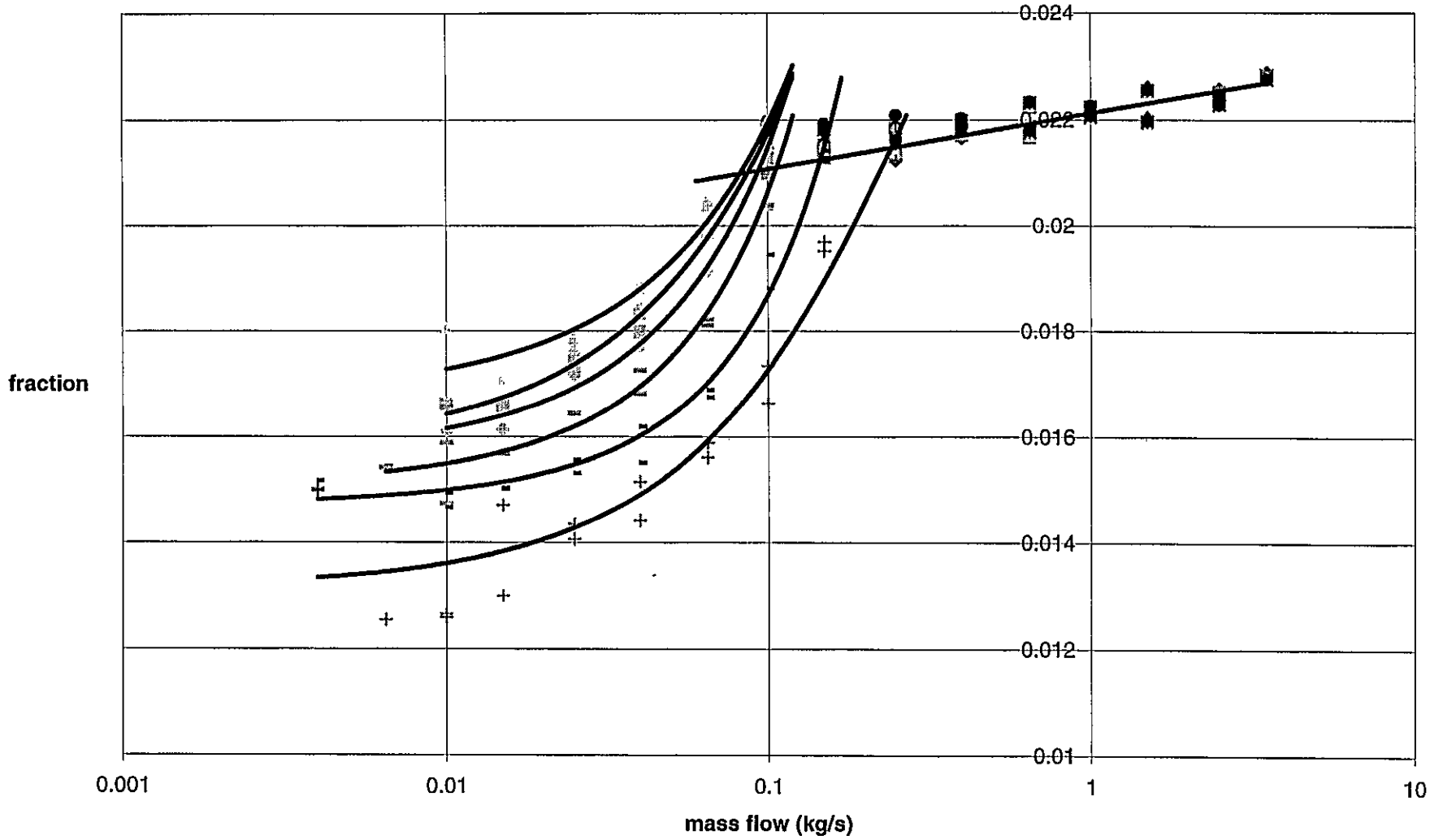


Figure #11
Flow Fractions for CV #3

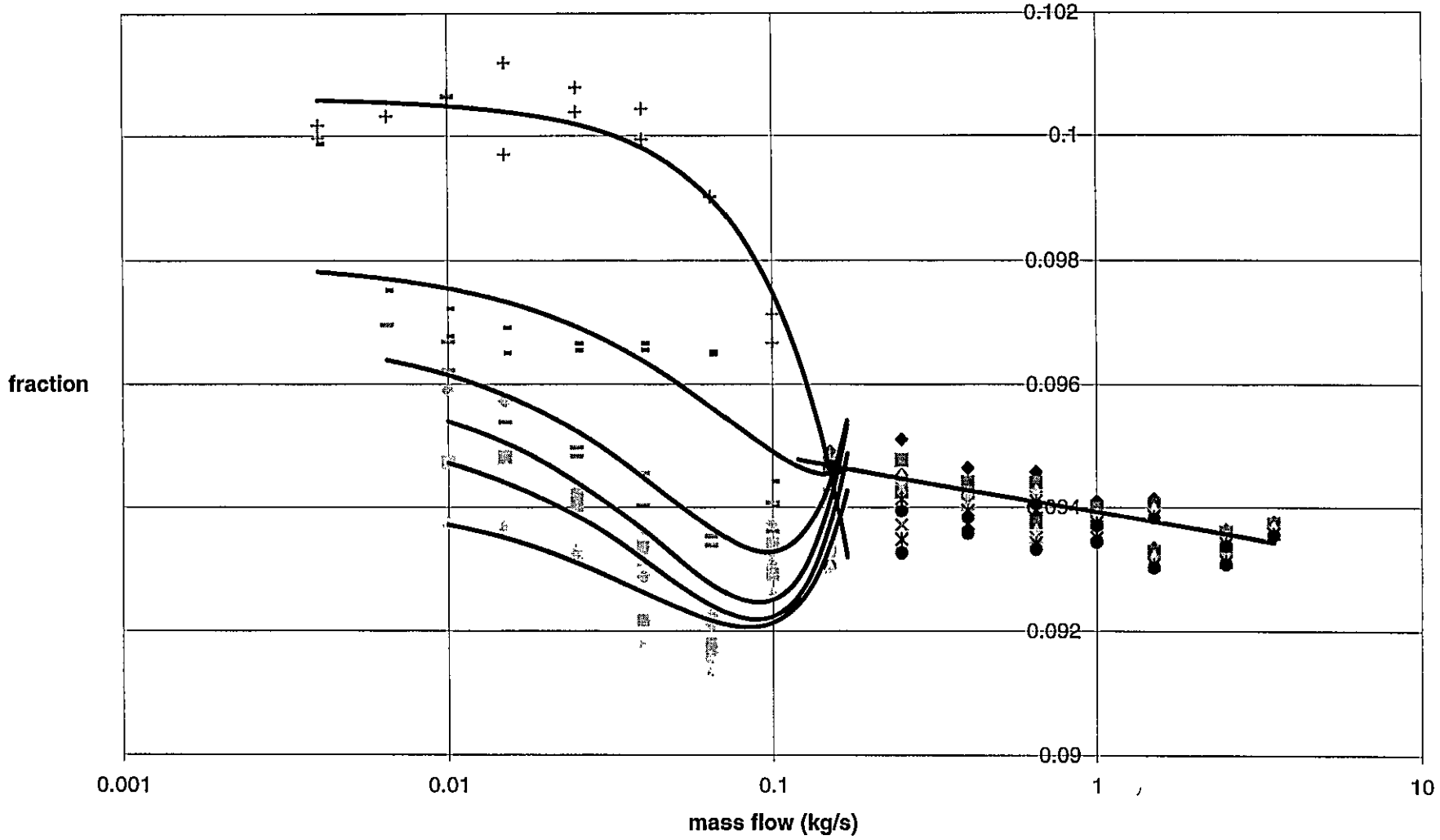


Figure #12
Flow Fractions for CV #4

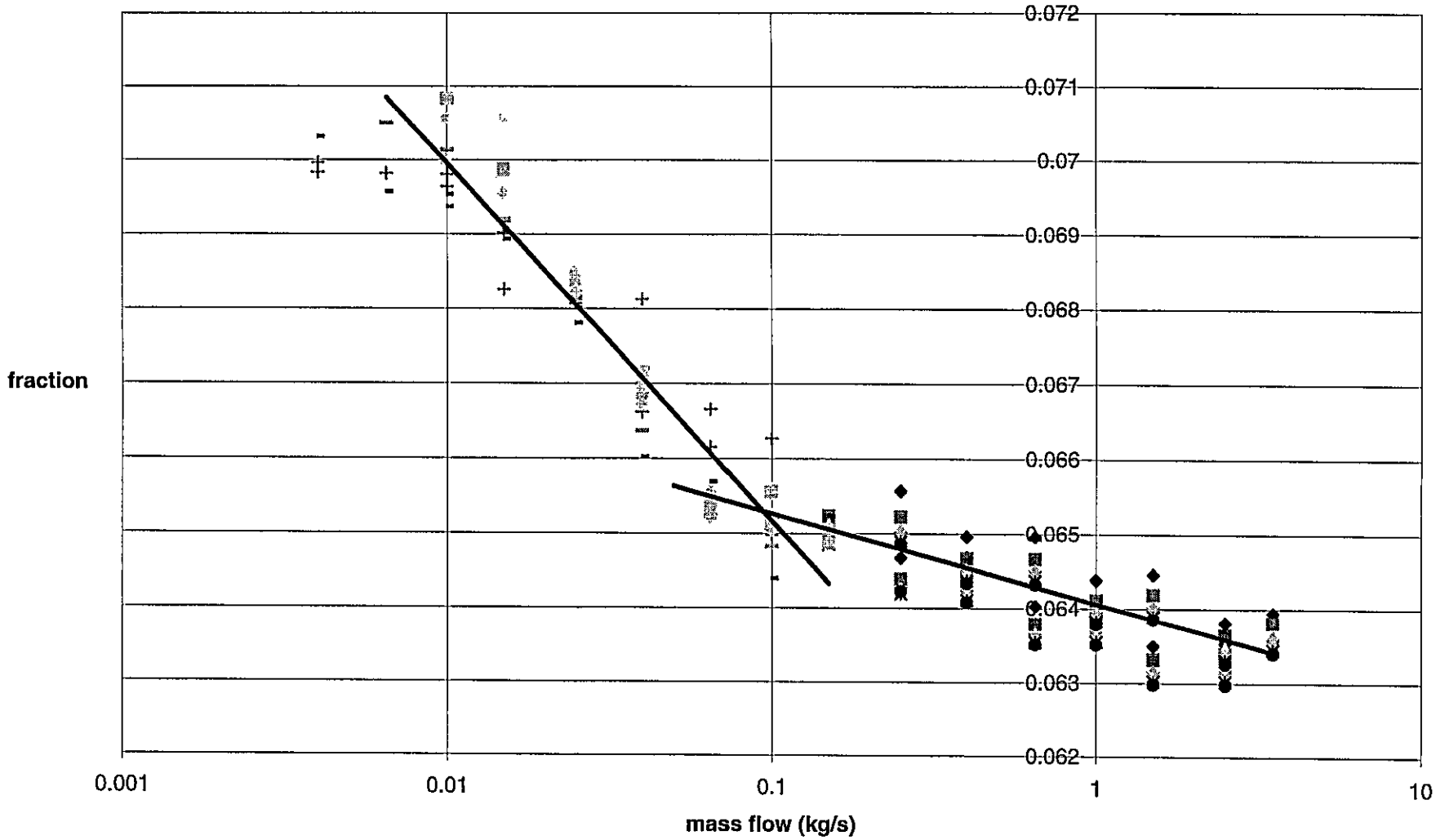


Figure #13
Flow Fractions for CV #5

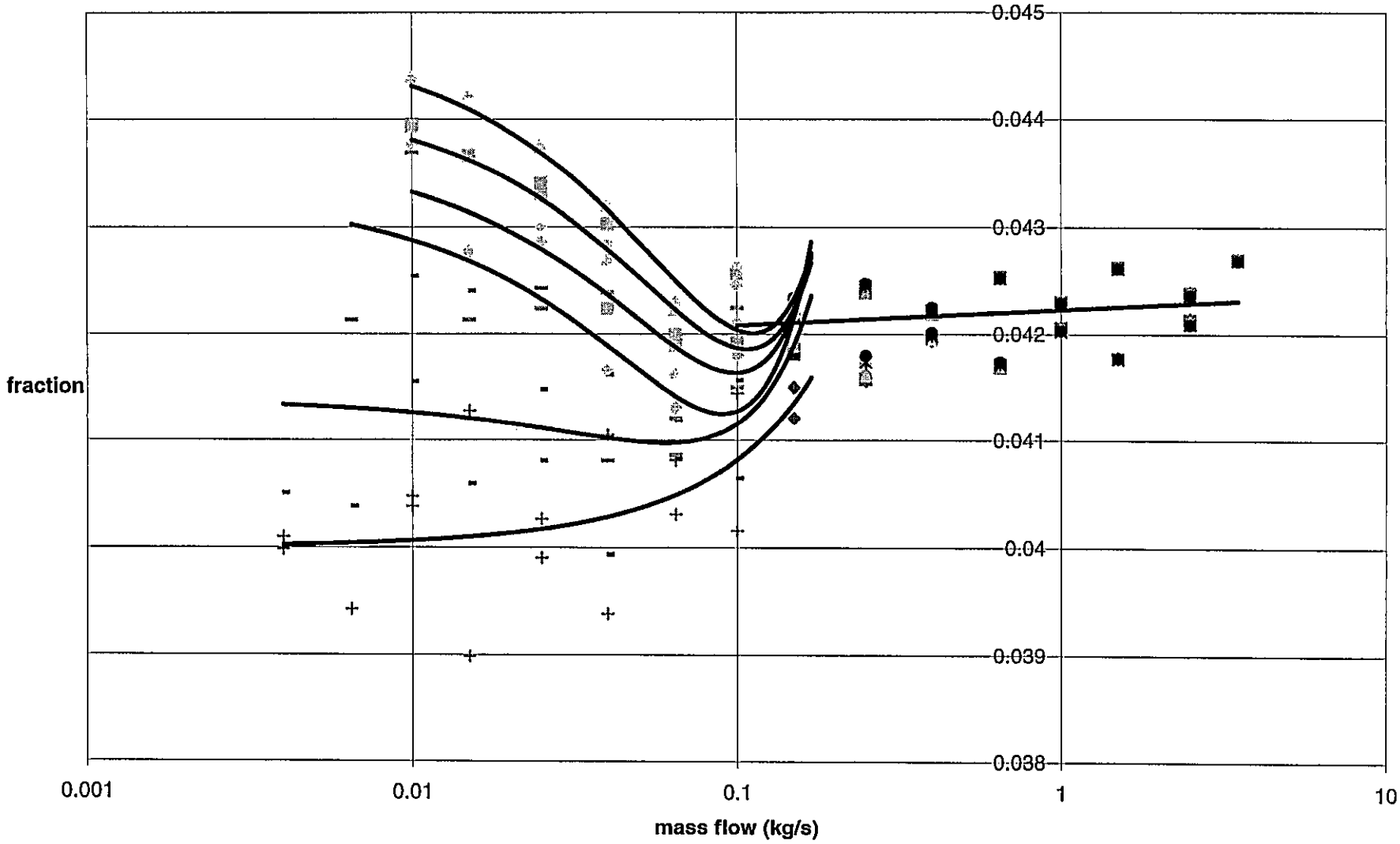


Figure #14
Flow Fractions for CV #6

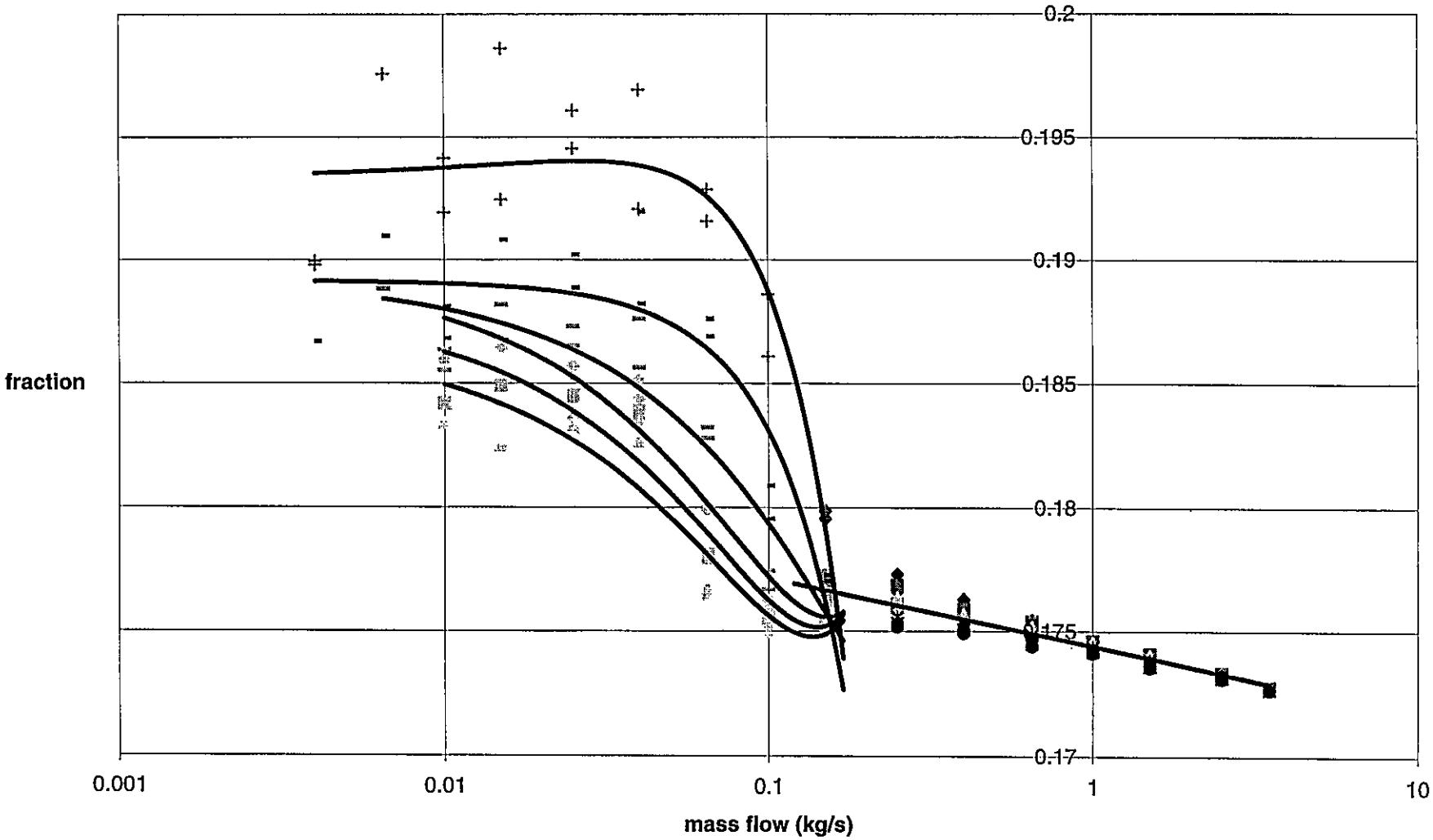


Figure #15
Flow Fractions for CV #7

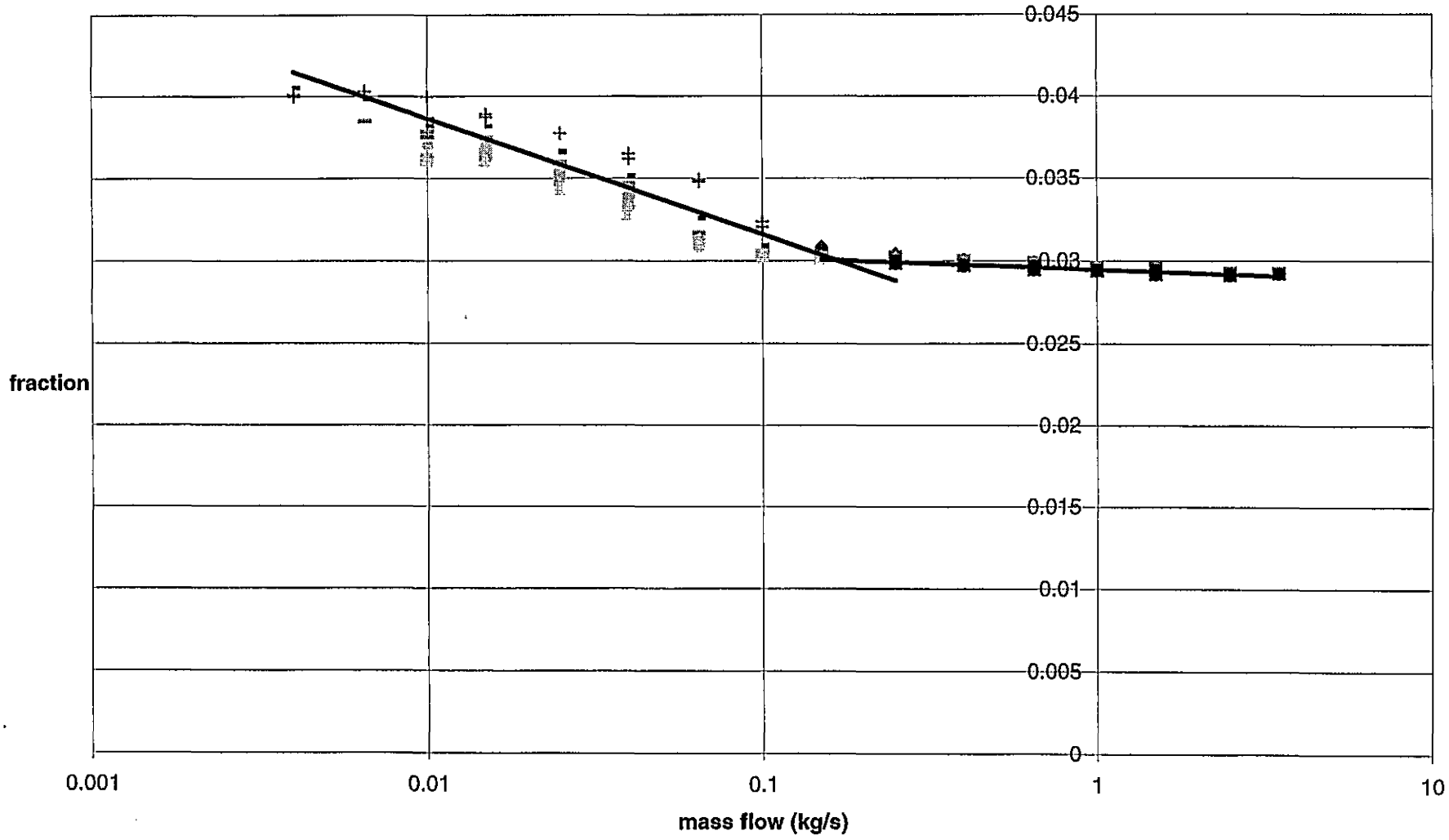


Figure #16
Flow Fractions for CV #8

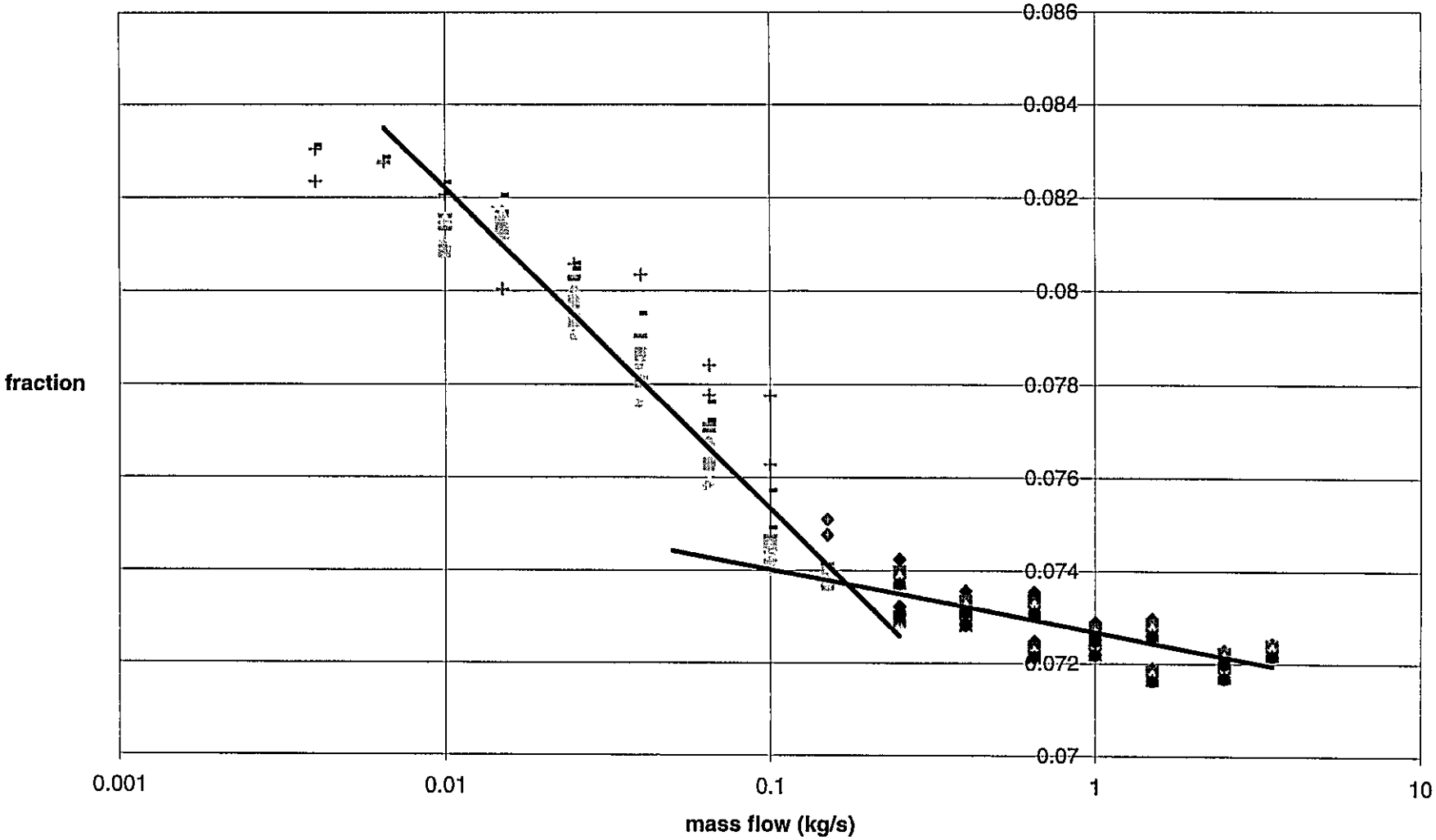


Figure #17
Flow Fractions for CV #9

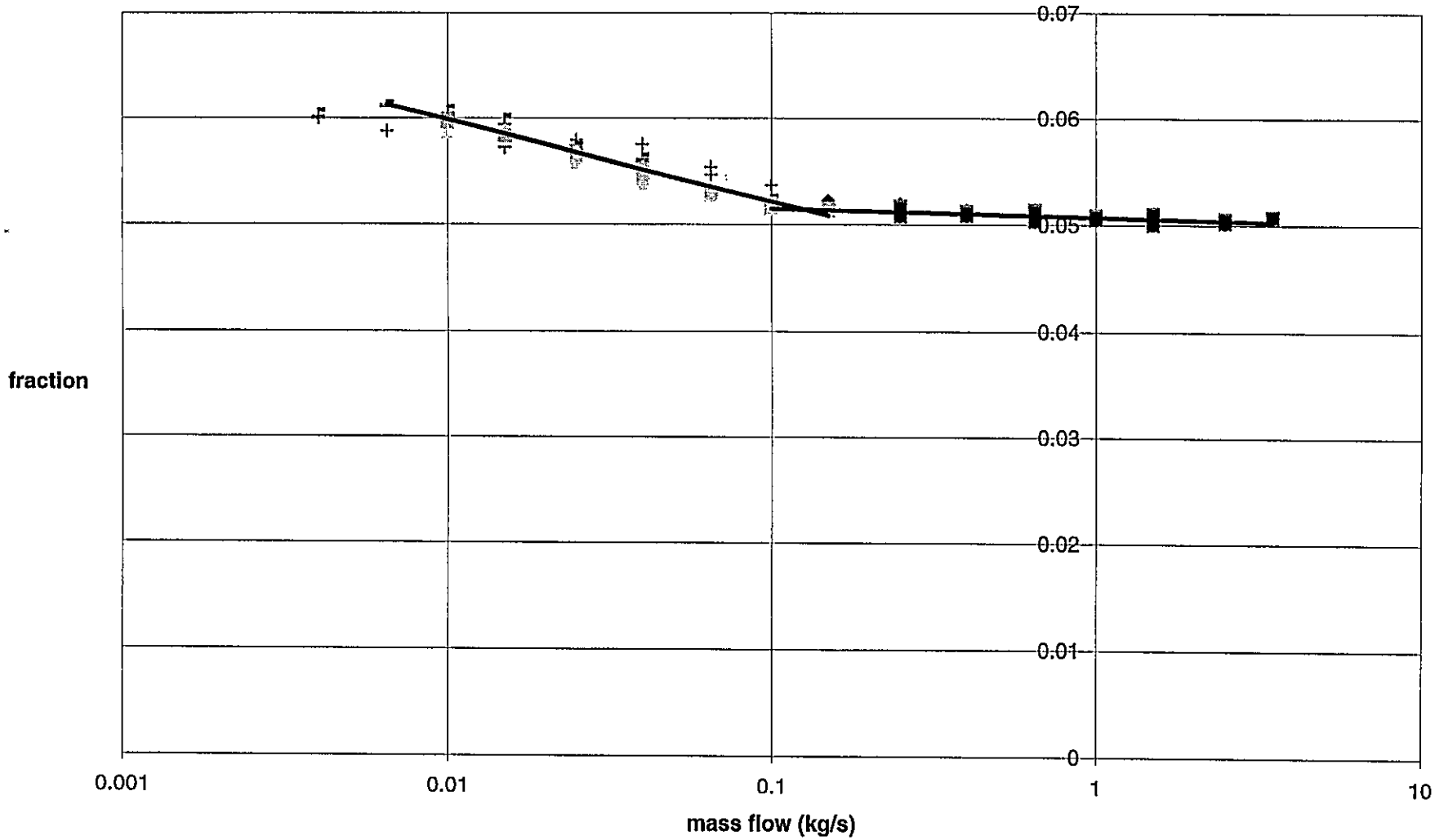


Figure #18
Flow Fractions for CV #10

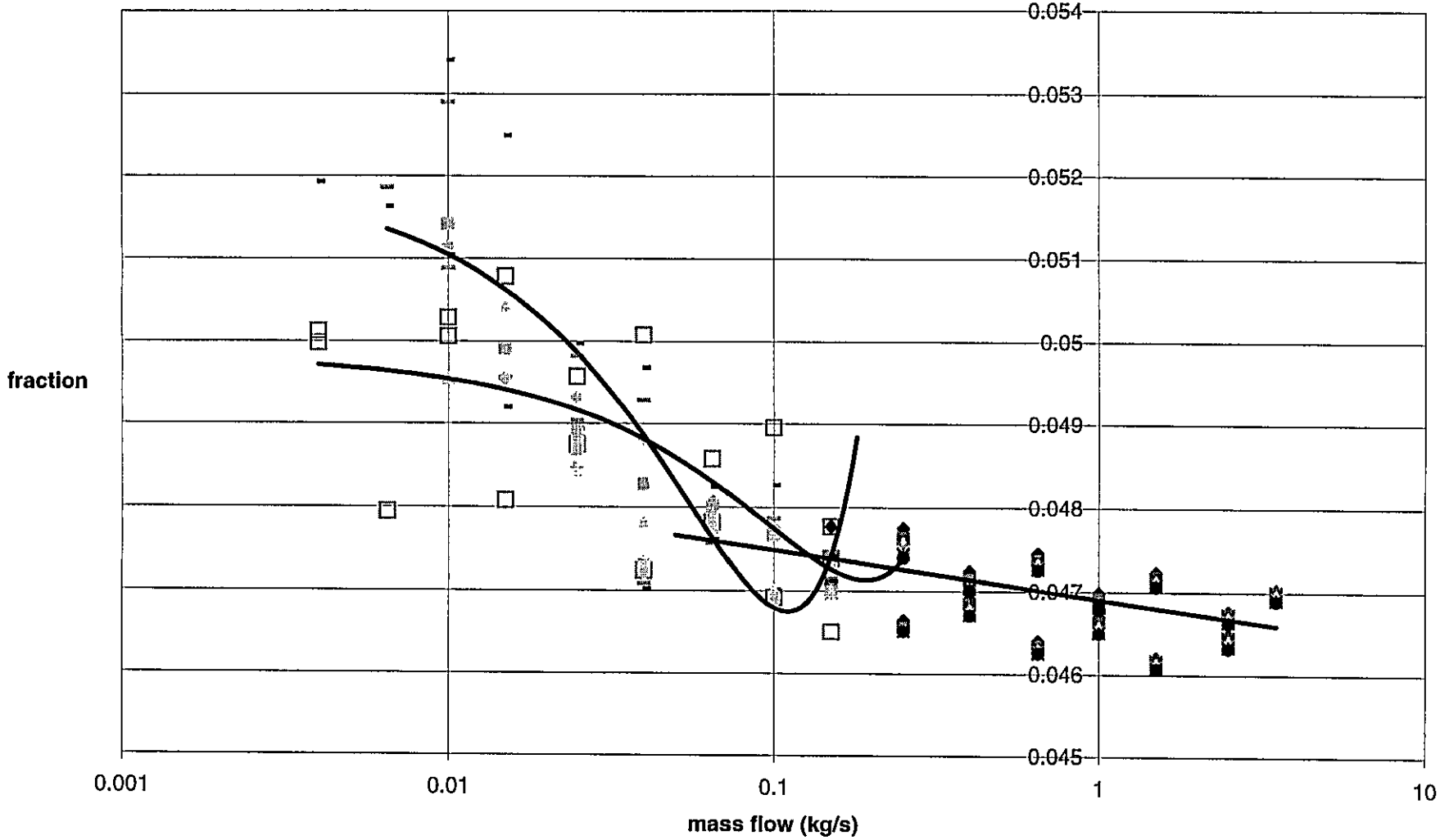


Figure #19
Flow Fractions for CV #11

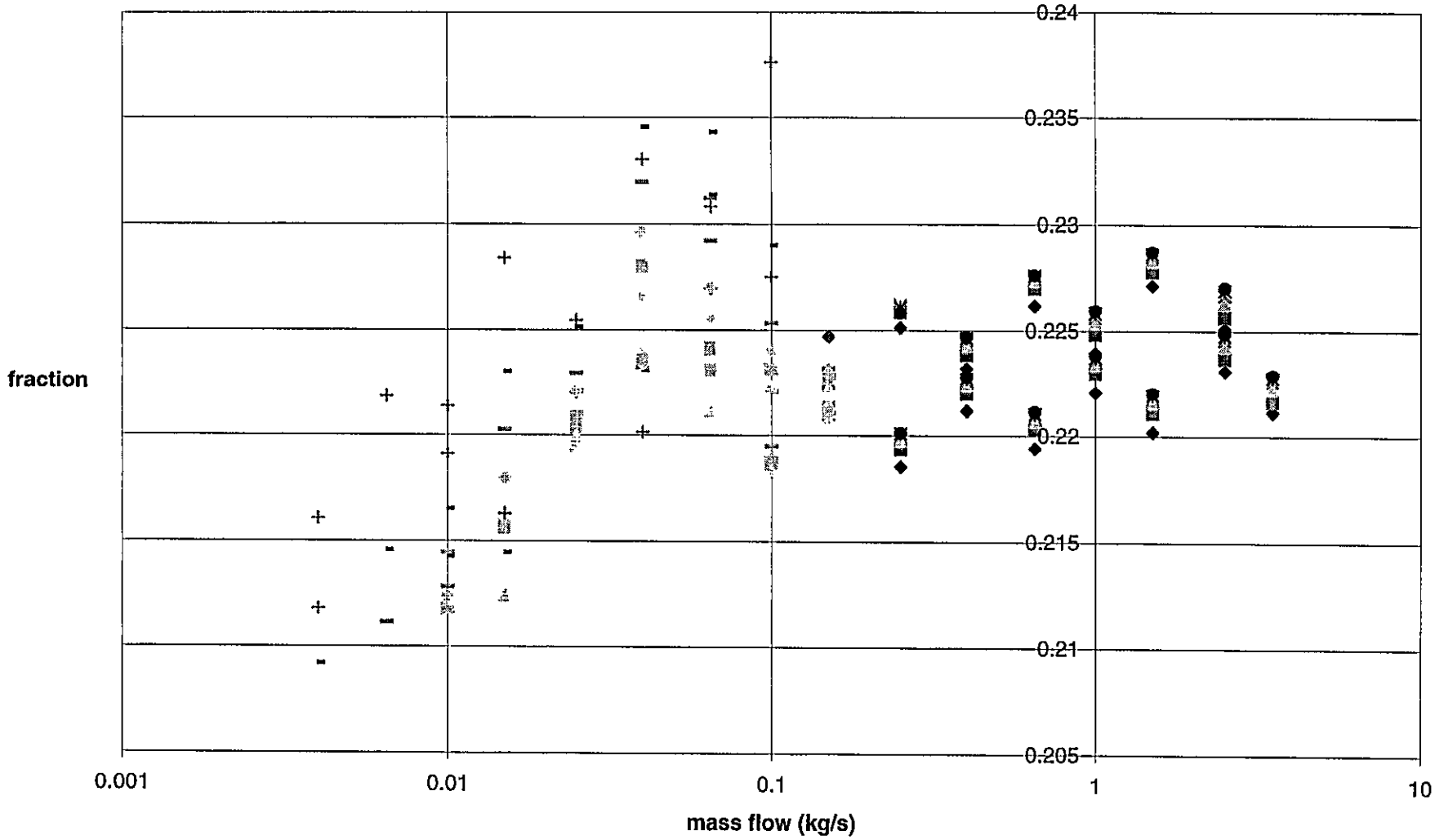


Figure #20
Flow Fractions for CV #12

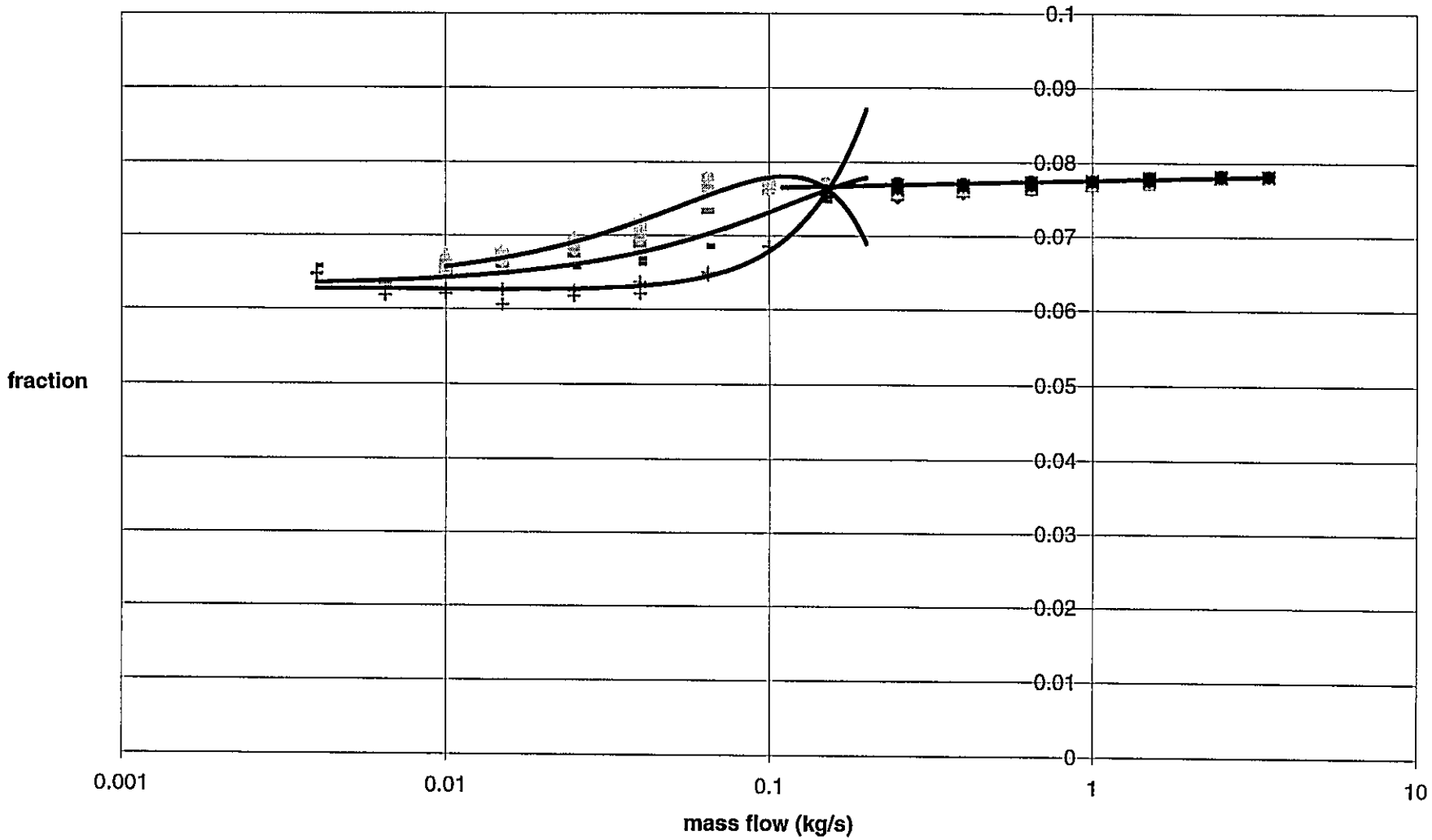


Figure #21
Flow Fractions for CV #13

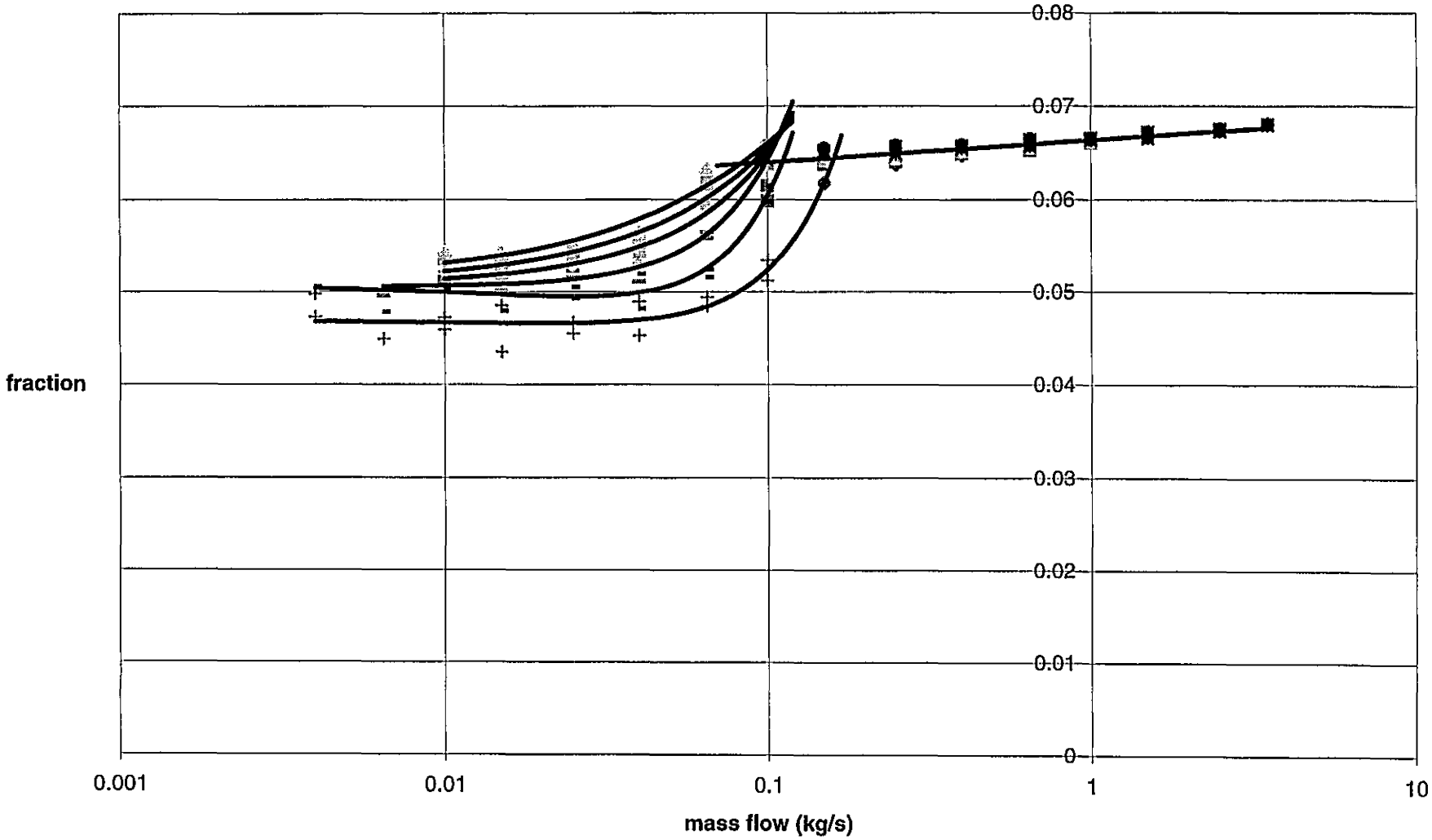


Figure #22
Flow Fractions for CV #14

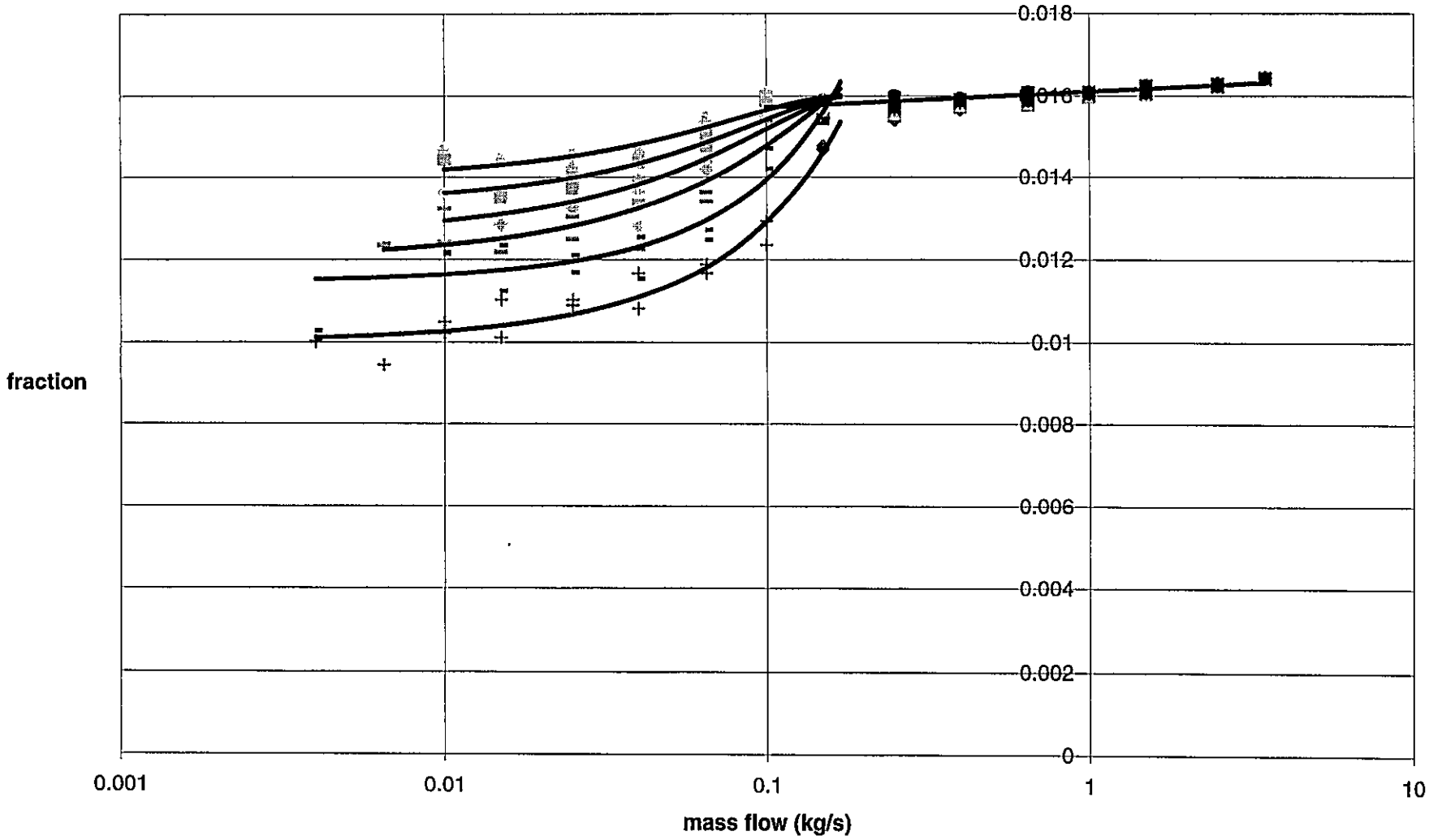
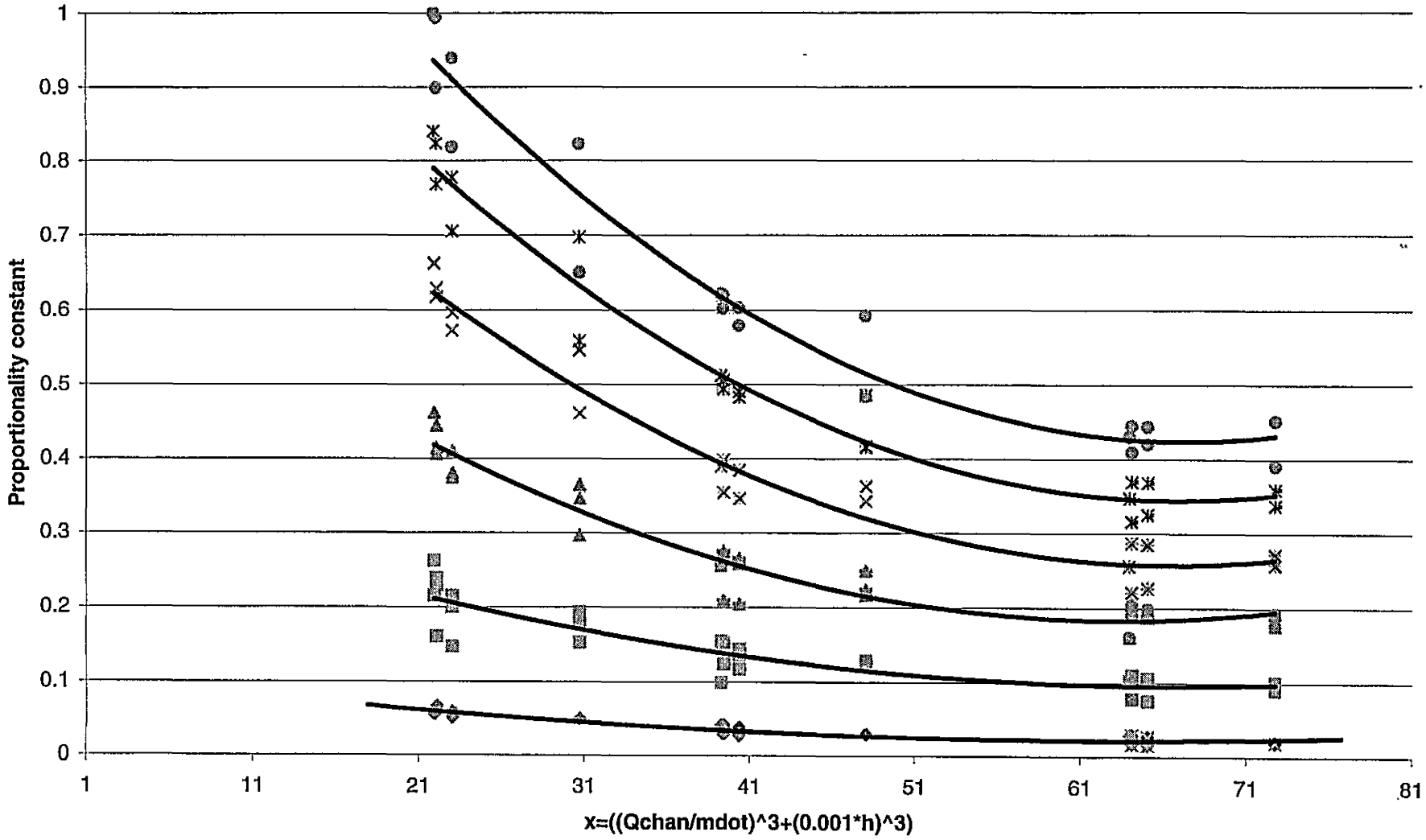
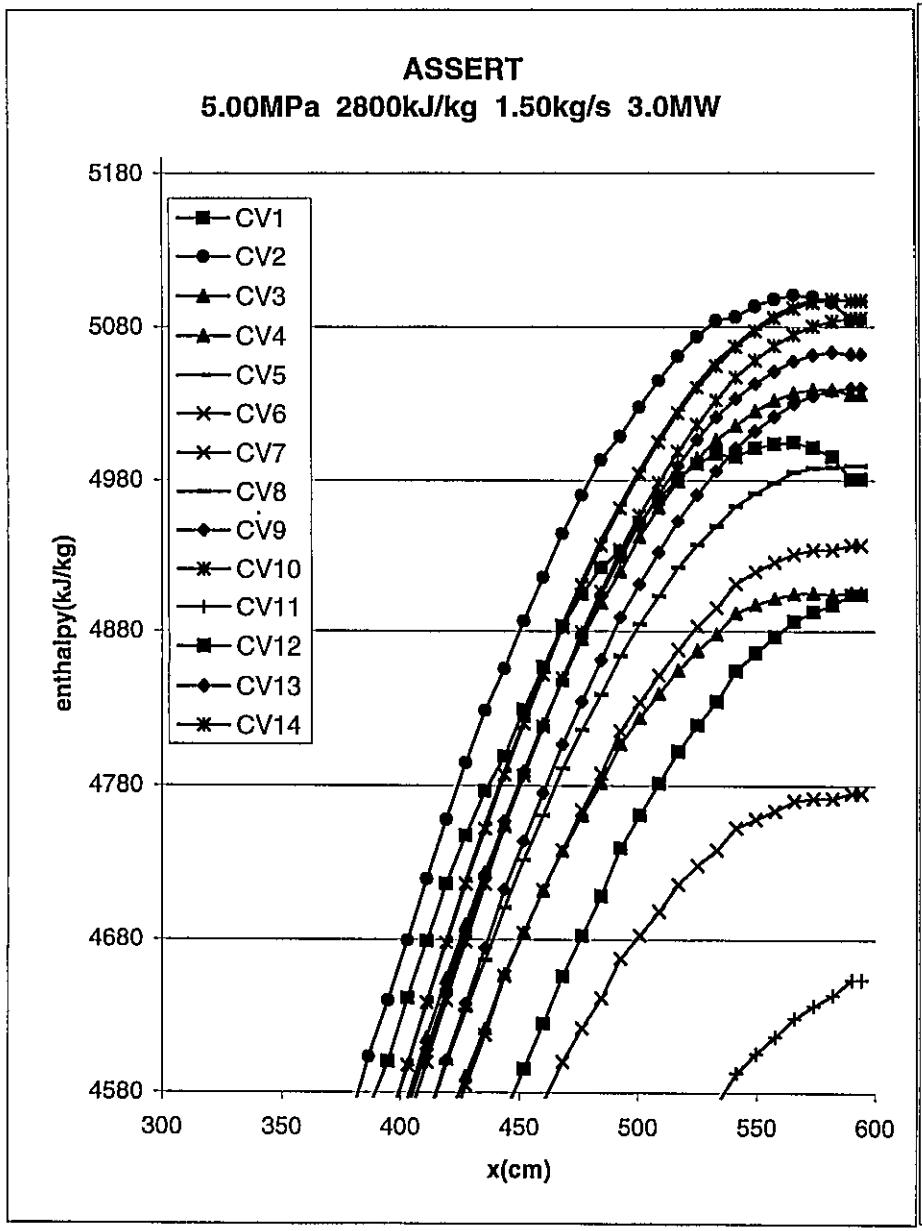
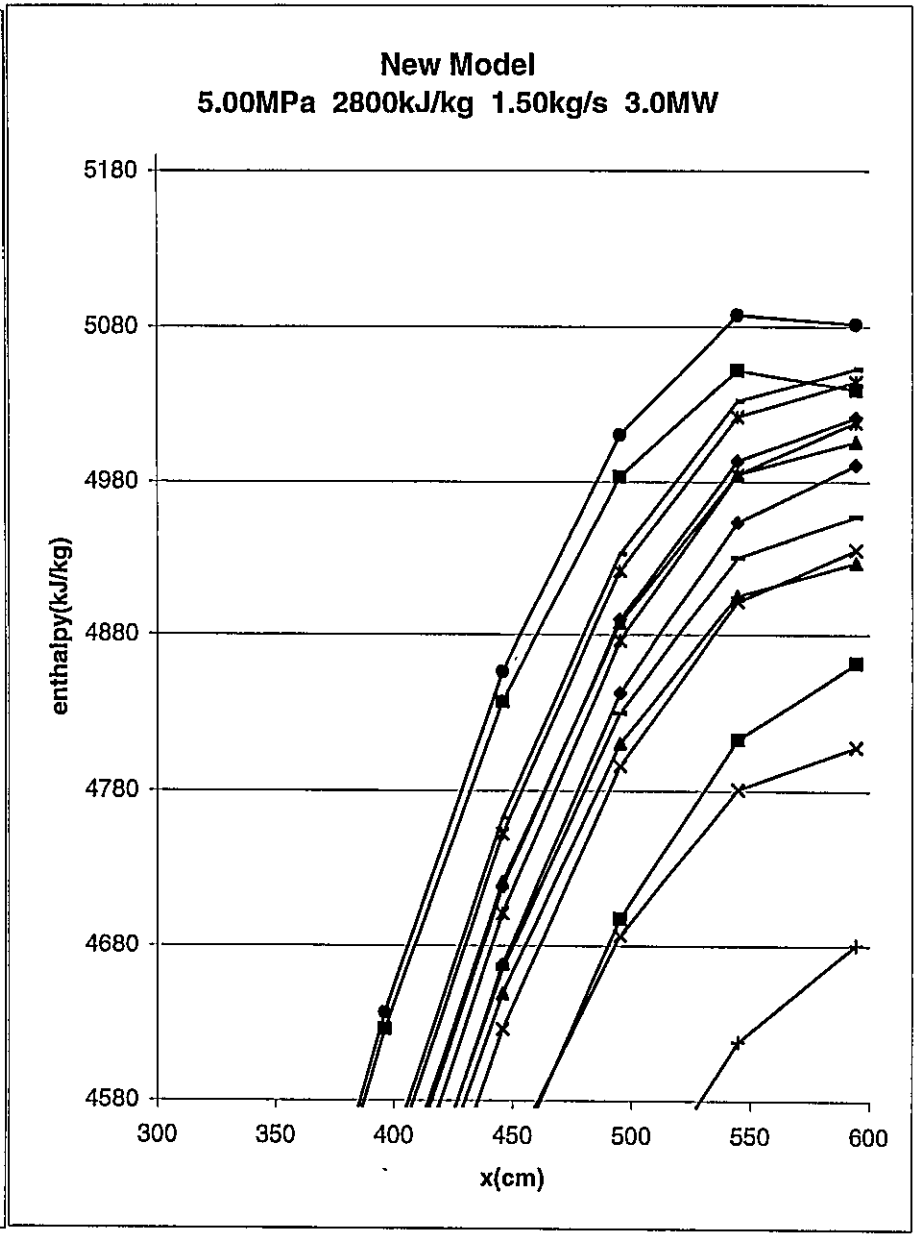
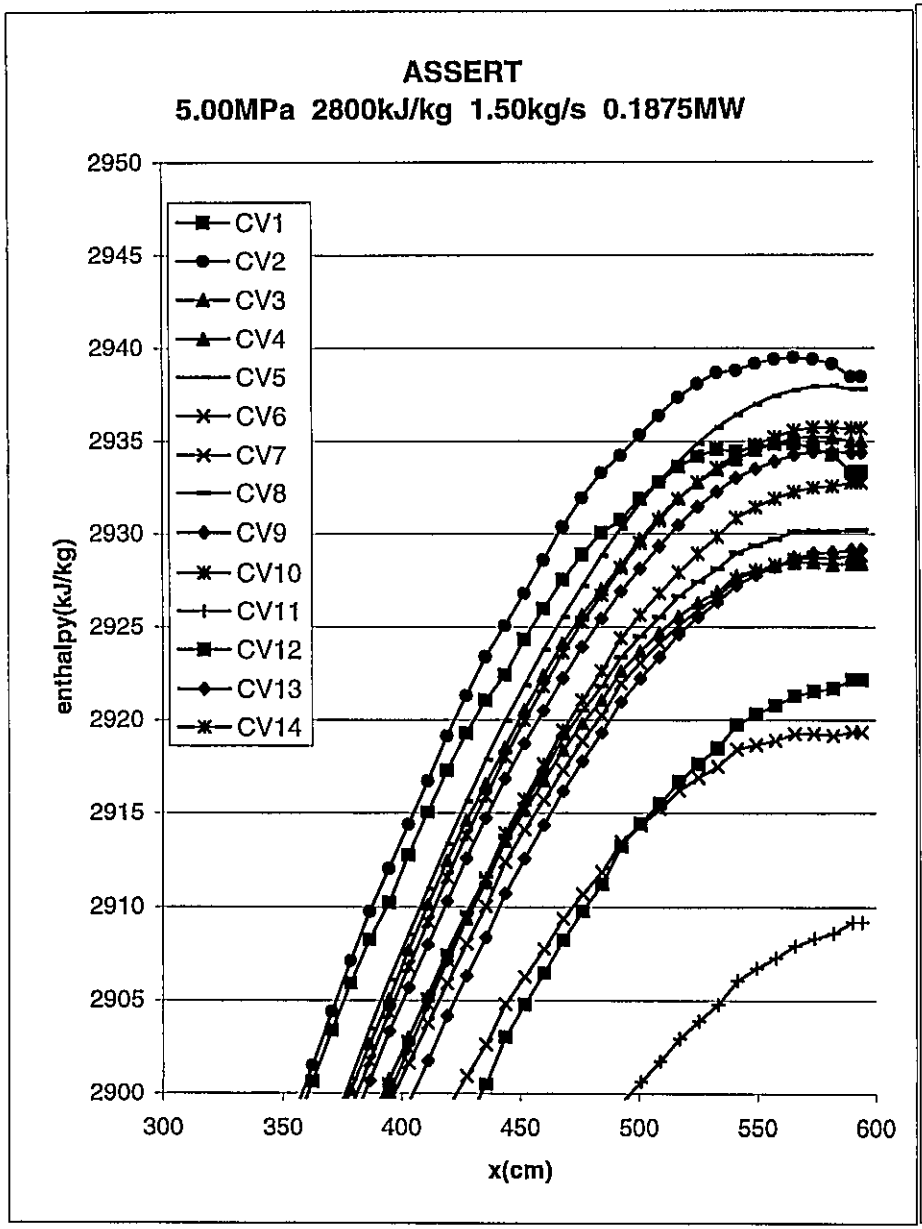


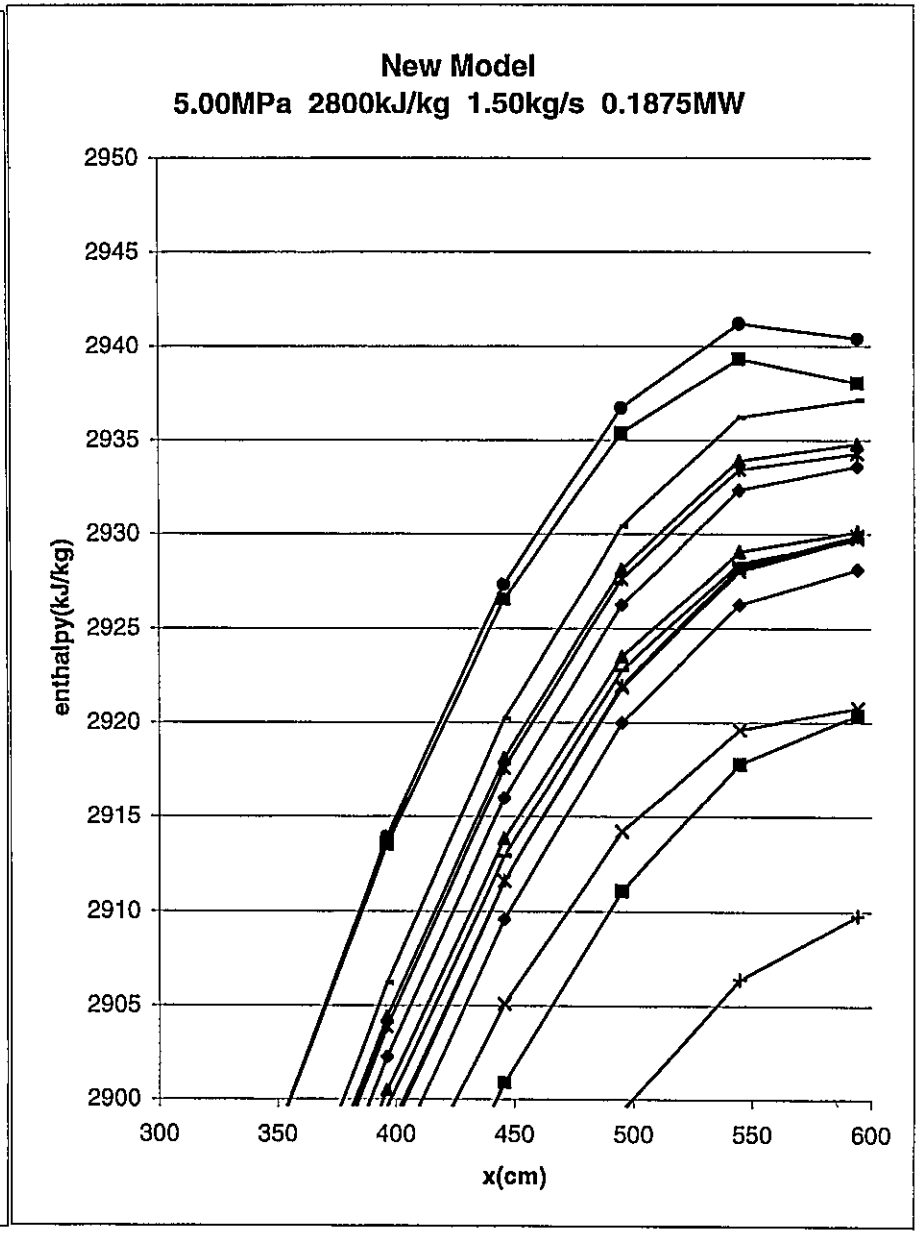
Figure #23
Proportionality Constant for Buoyancy Flow
(each curve represents a different channel outlet pressure)

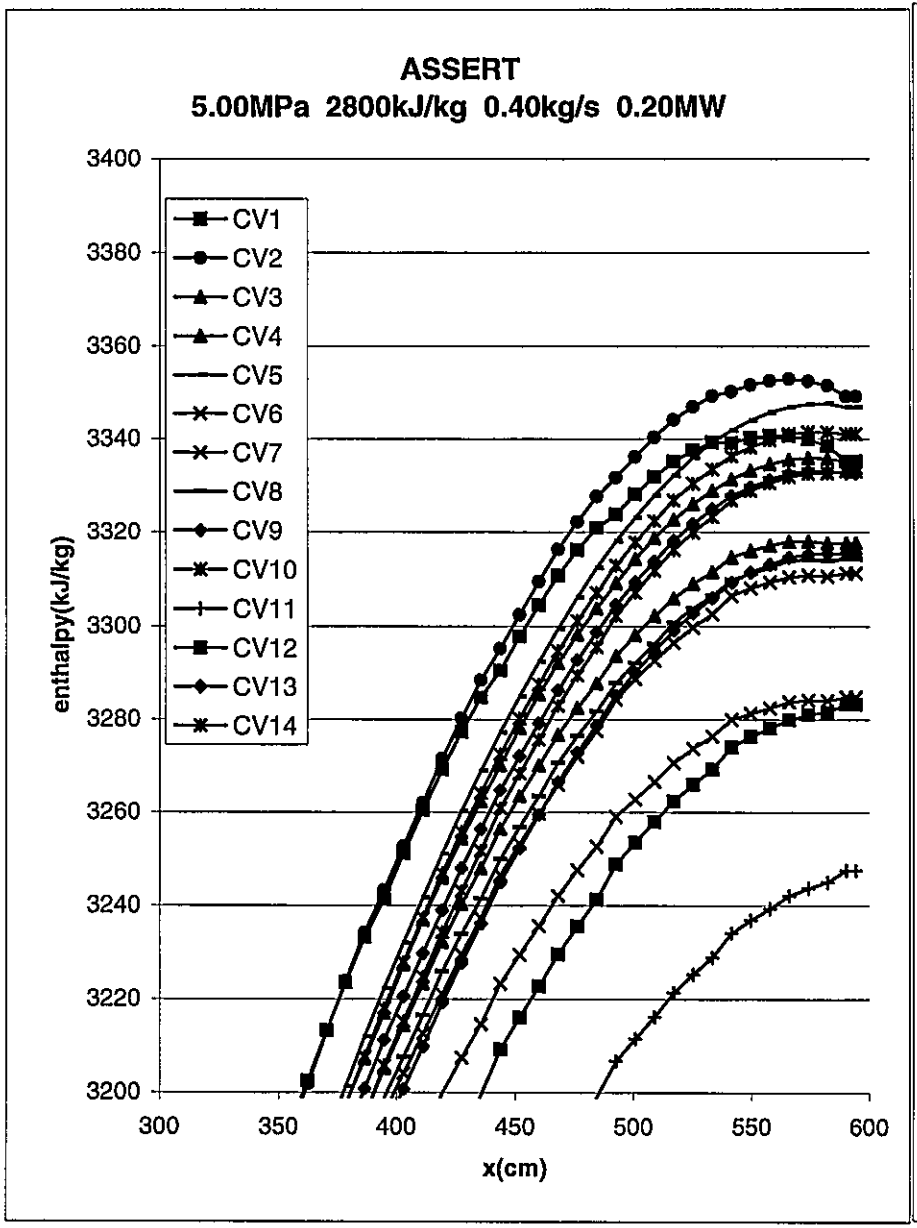


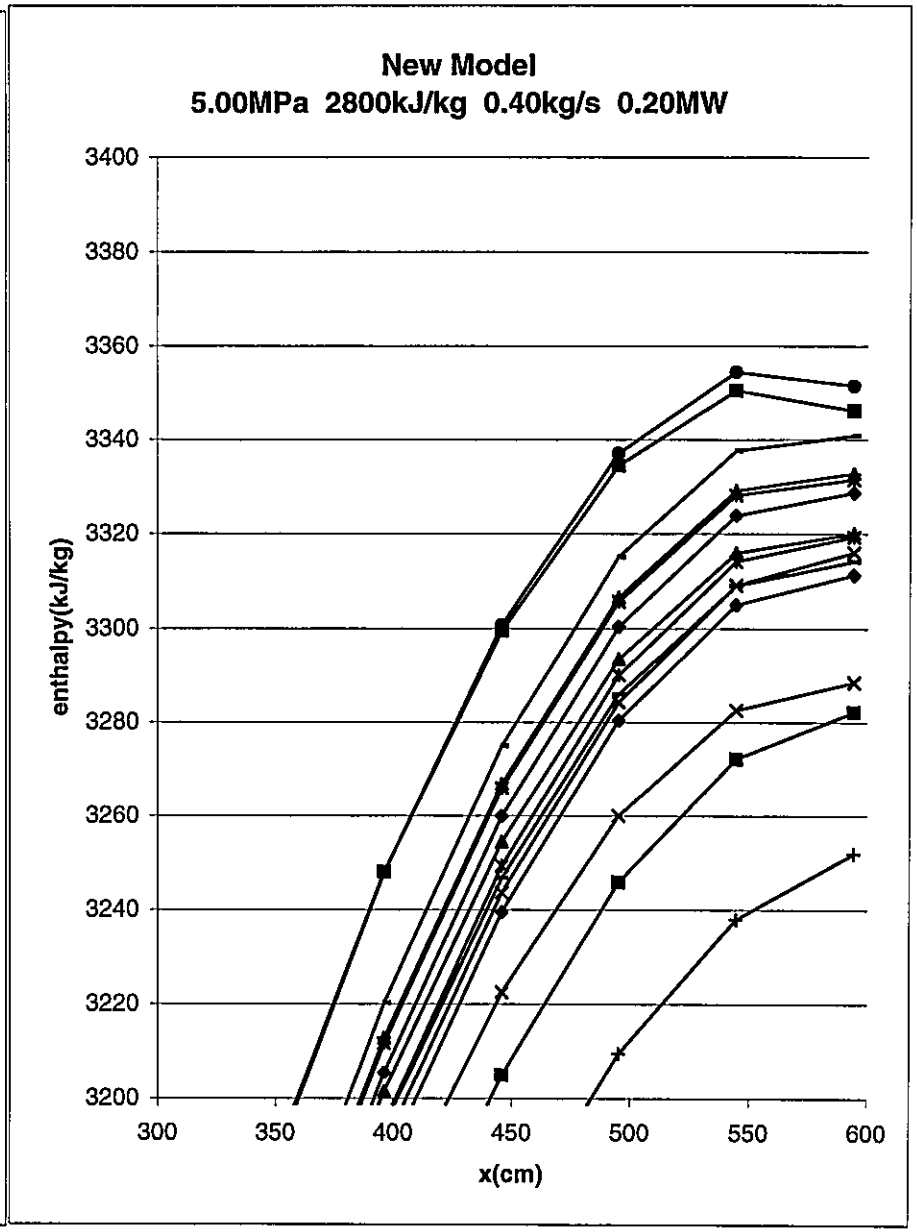

ASSERT
5.00MPa 2800kJ/kg 1.50kg/s 3.0MW
5180
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

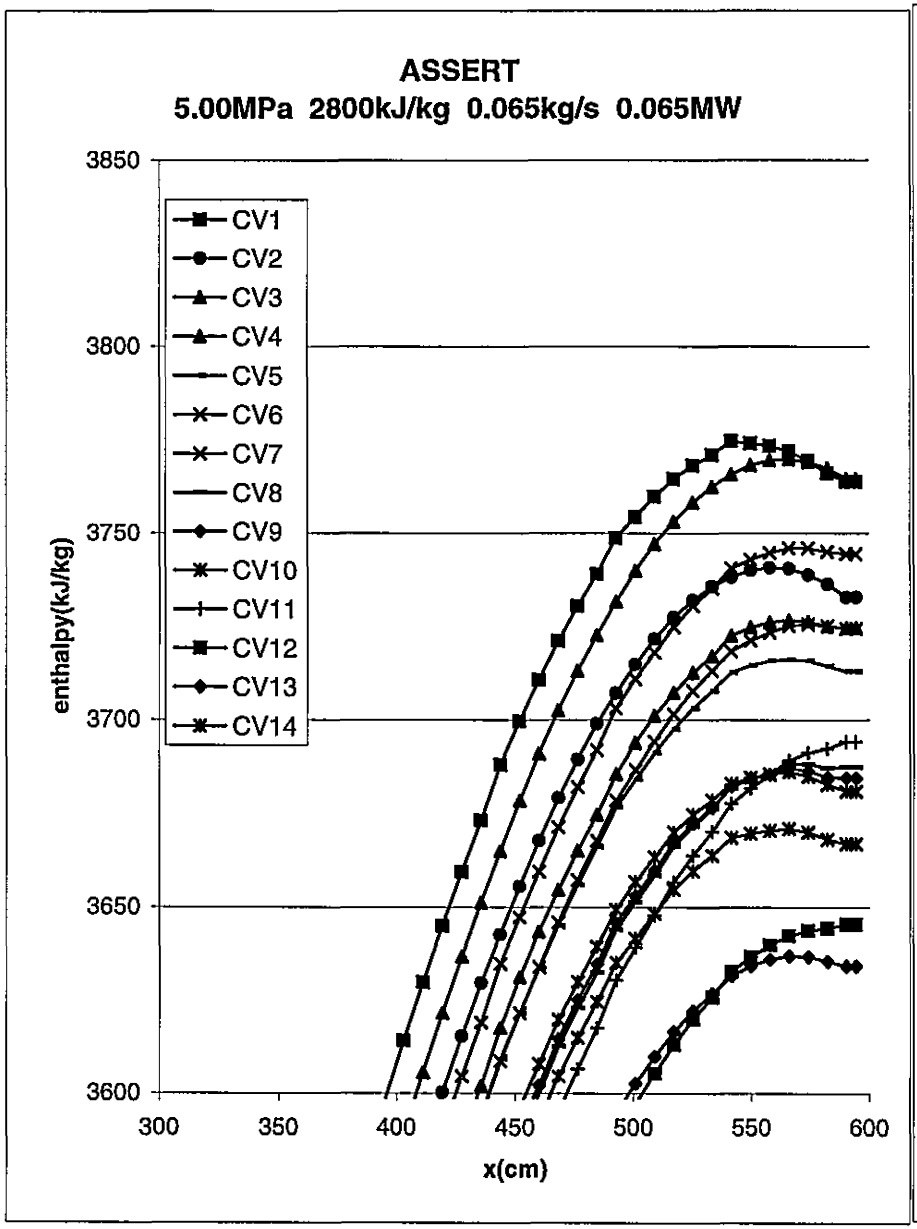

New Model
5.00MPa 2800kJ/kg 1.50kg/s 3.0MW
5180
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

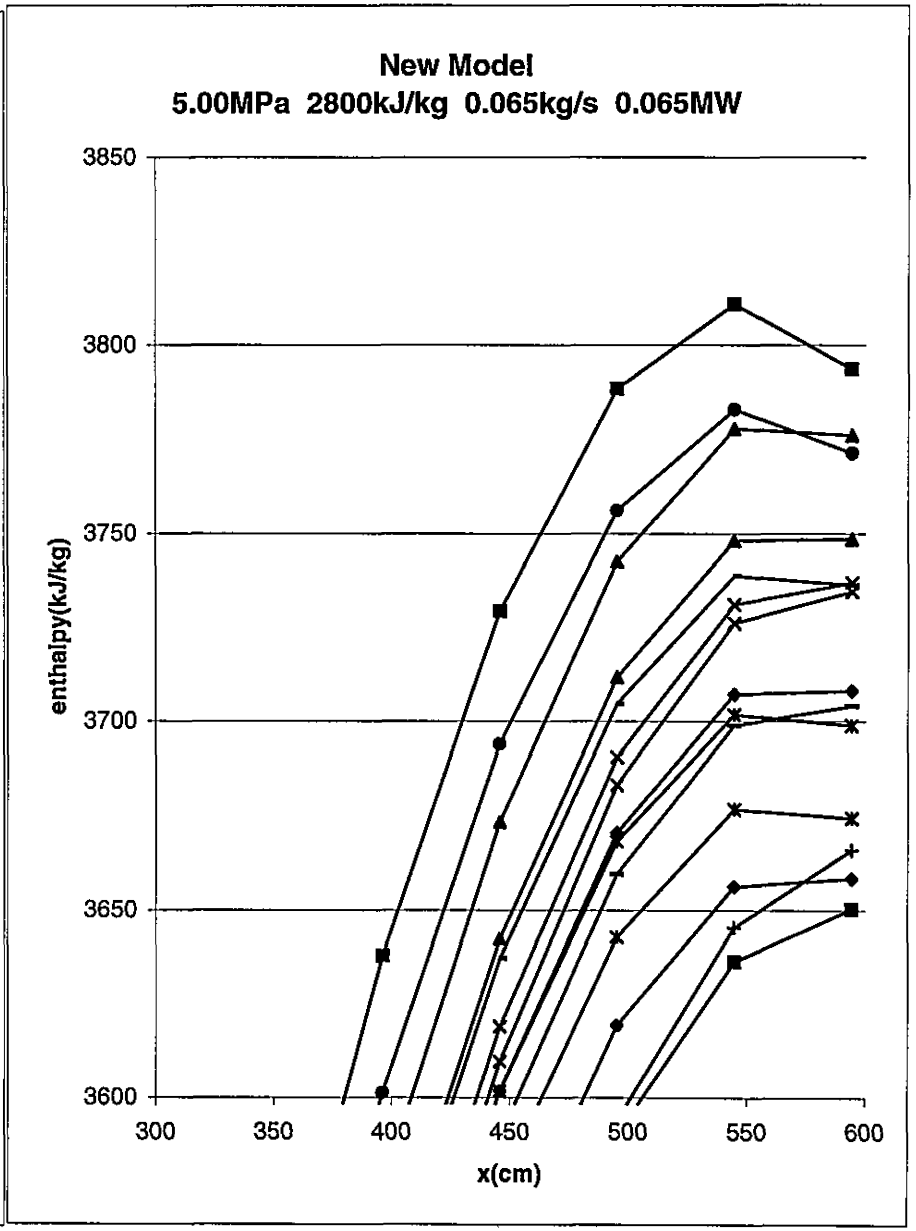

ASSERT
5.00MPa 2800kJ/kg 1.50kg/s 0.1875MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

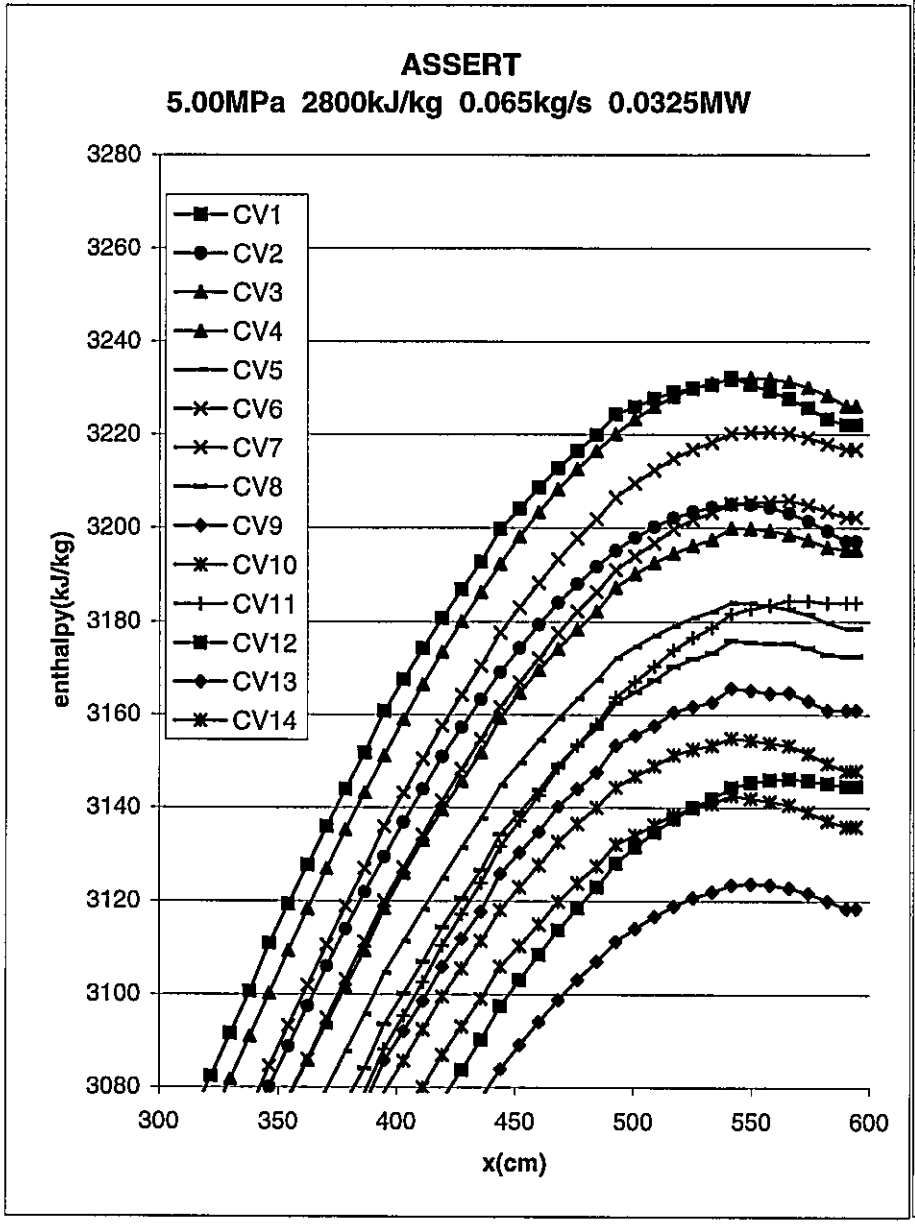

New Model
5.00MPa 2800kJ/kg 1.50kg/s 0.1875MW
enthalpy(kJ/kg)
x(cm)
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

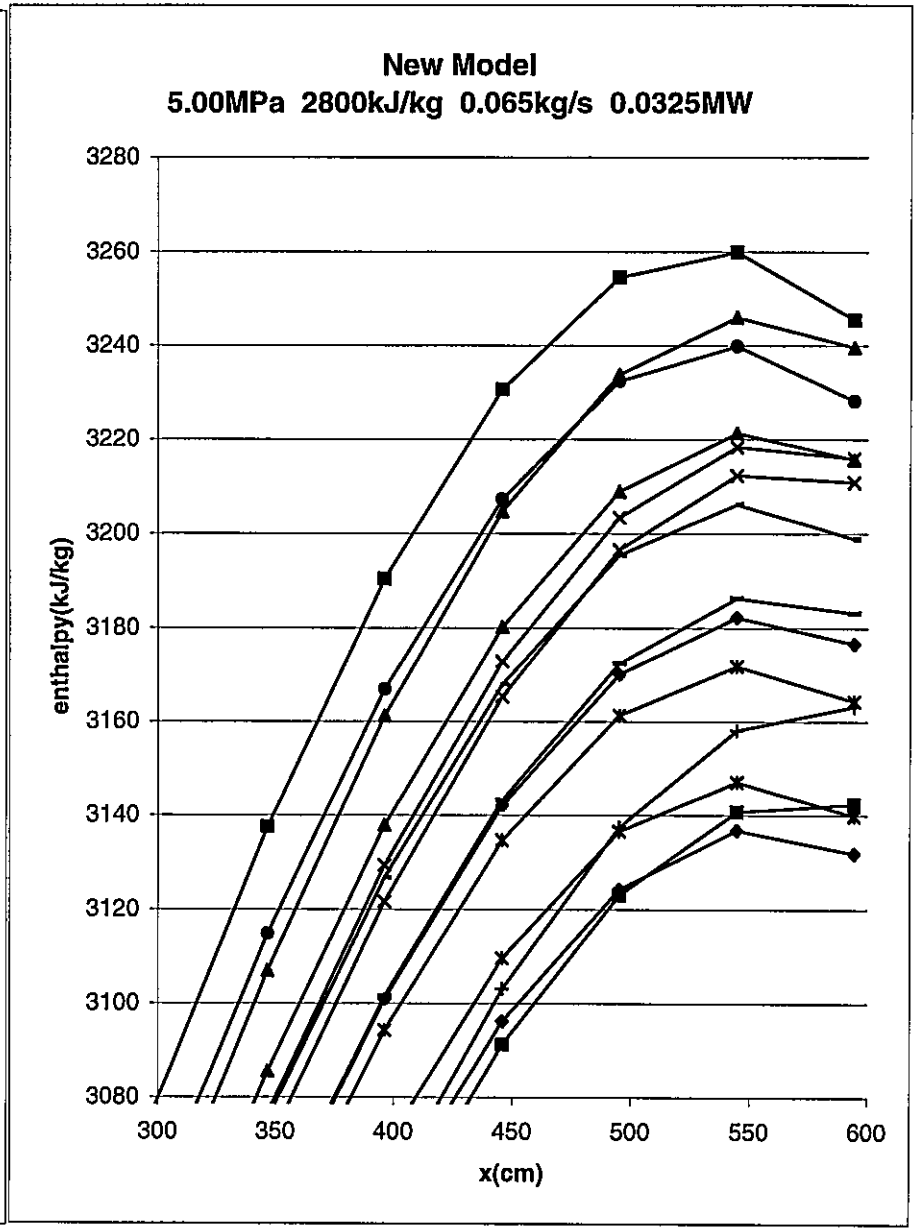

ASSERT
5.00MPa 2800kJ/kg 0.40kg/s 0.20MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3400
3380
3360
3340
3320
3300
3280
3260
3240
3220
3200
300
350
400
450
500
550
600

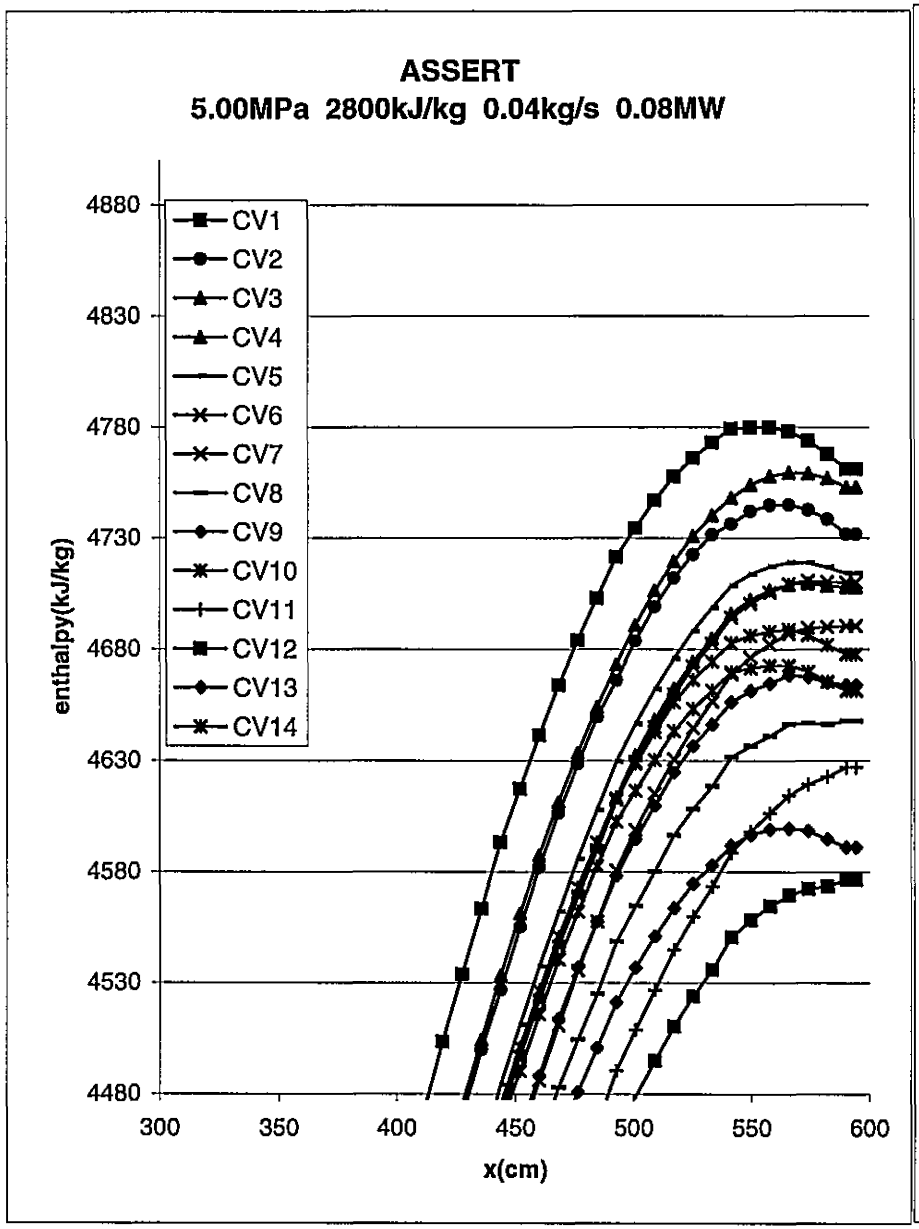

New Model
5.00MPa 2800kJ/kg 0.40kg/s 0.20MW
enthalpy(kJ/kg)
x(cm)
3400
3380
3360
3340
3320
3300
3280
3260
3240
3220
3200
300
350
400
450
500
550
600

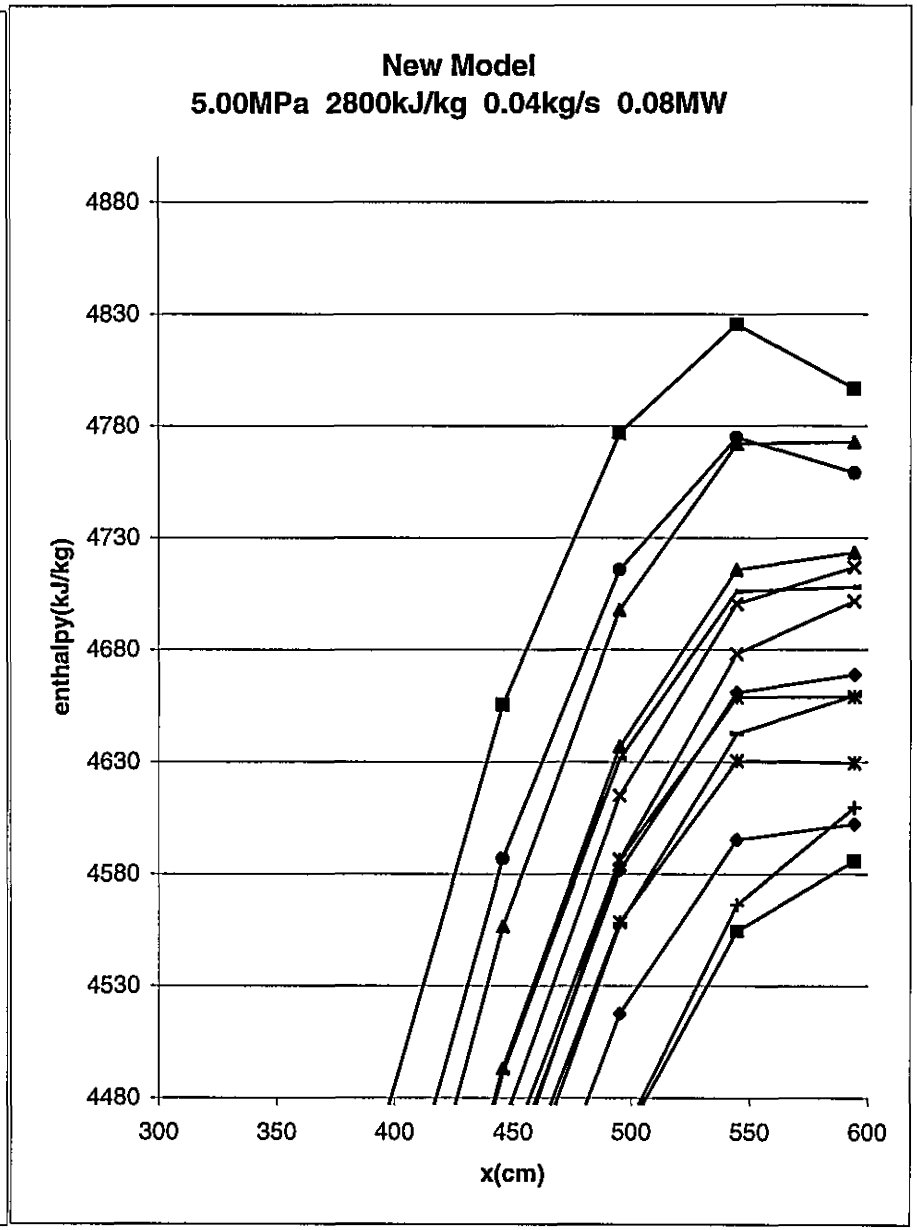

ASSERT
5.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3850
3800
3750
3700
3650
3600
300
350
400
450
500
550
600

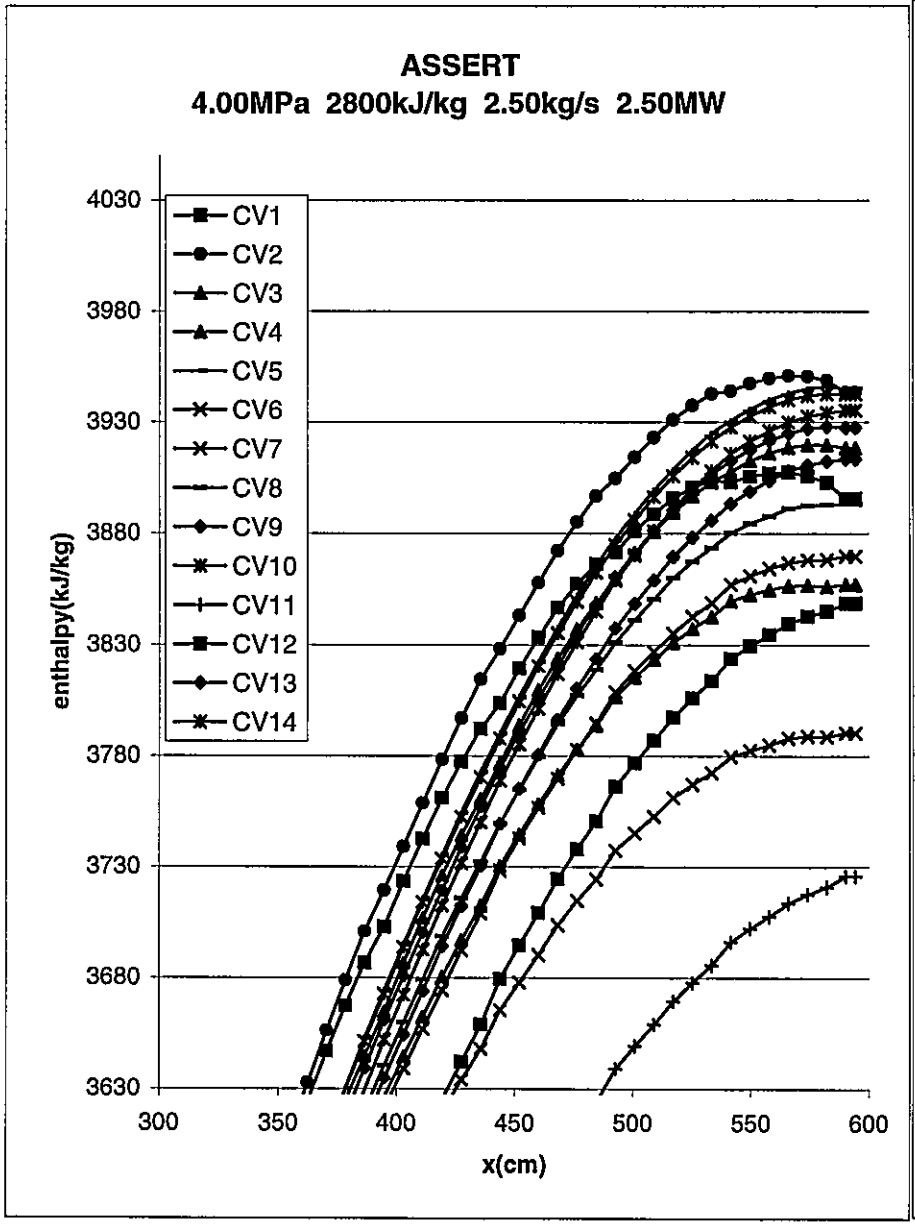

New Model
5.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
x(cm)
3850
3800
3750
3700
3650
3600
300
350
400
450
500
550
600

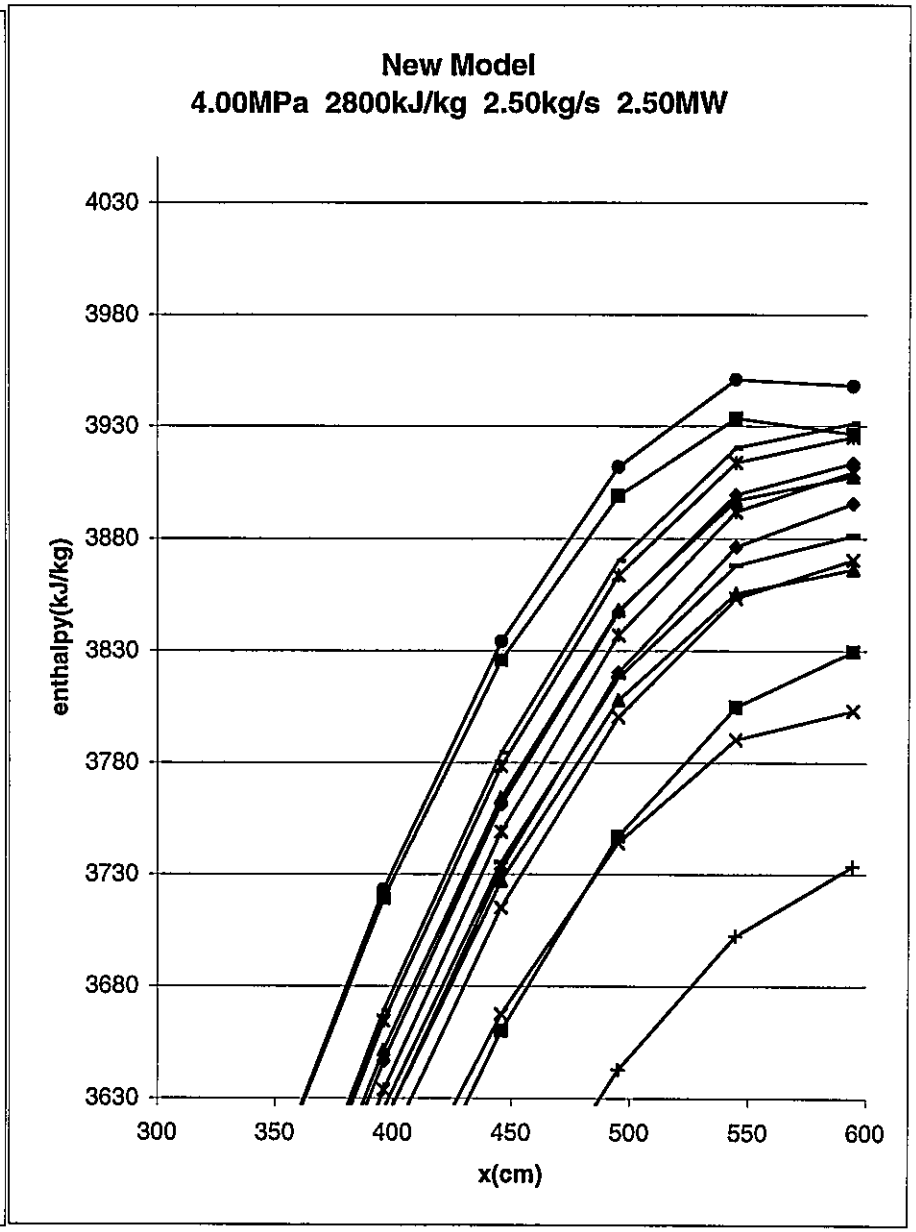

ASSERT
5.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3280
3260
3240
3220
3200
3180
3160
3140
3120
3100
3080
300
350
400
450
500
550
600

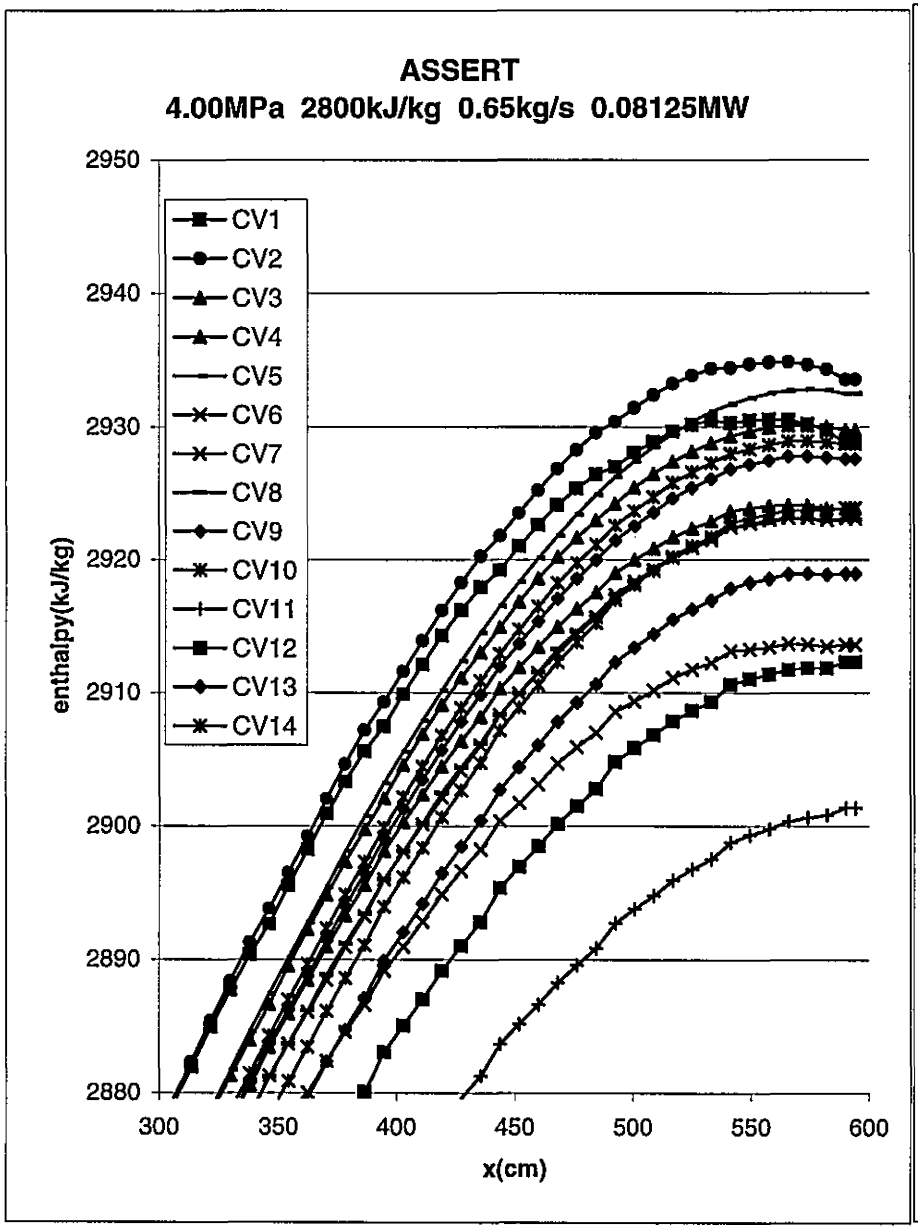

New Model
5.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
enthalpy(kJ/kg)
x(cm)
3280
3260
3240
3220
3200
3180
3160
3140
3120
3100
3080
300
350
400
450
500
550
600

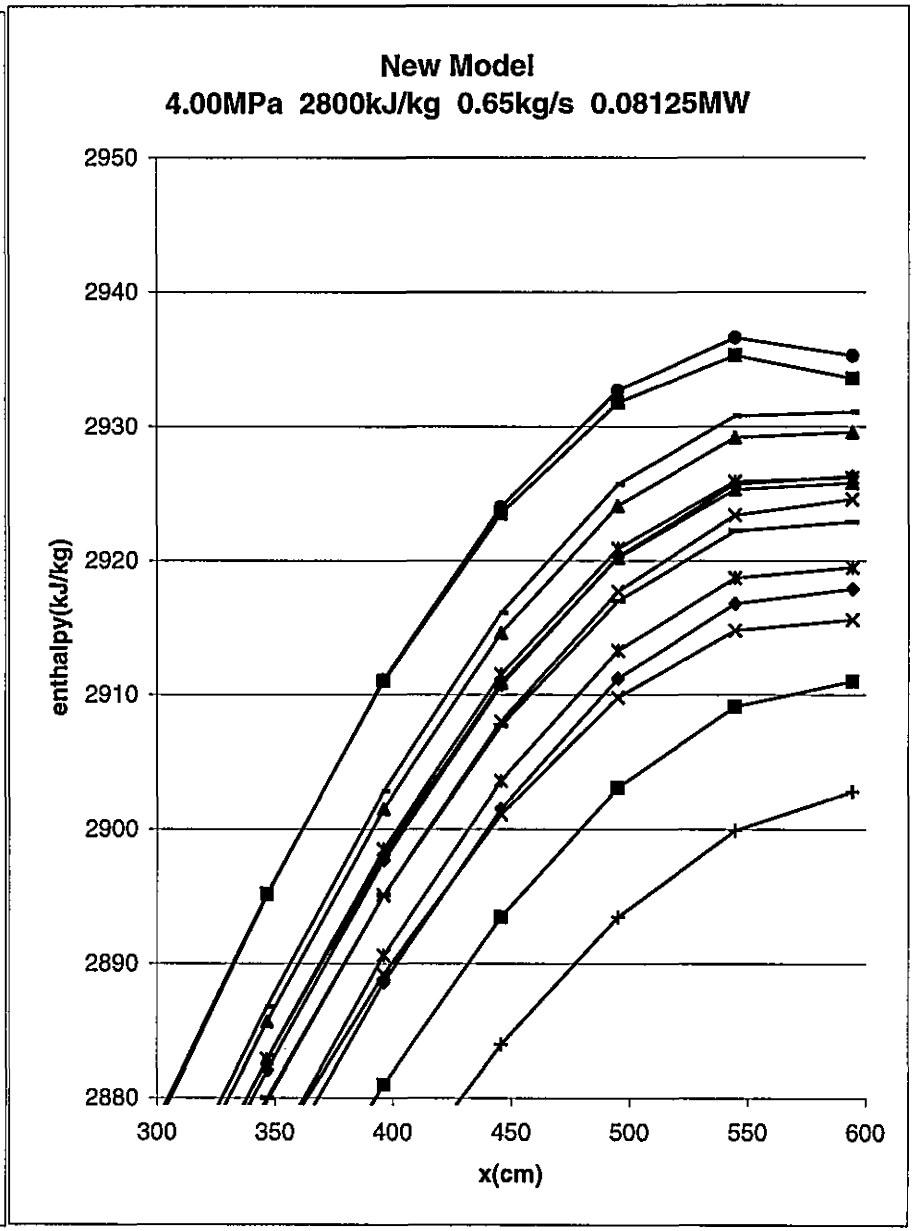

ASSERT
5.00MPa 2800kJ/kg 0.04kg/s 0.08MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
4880
4830
4780
4730
4680
4630
4580
4530
4480
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

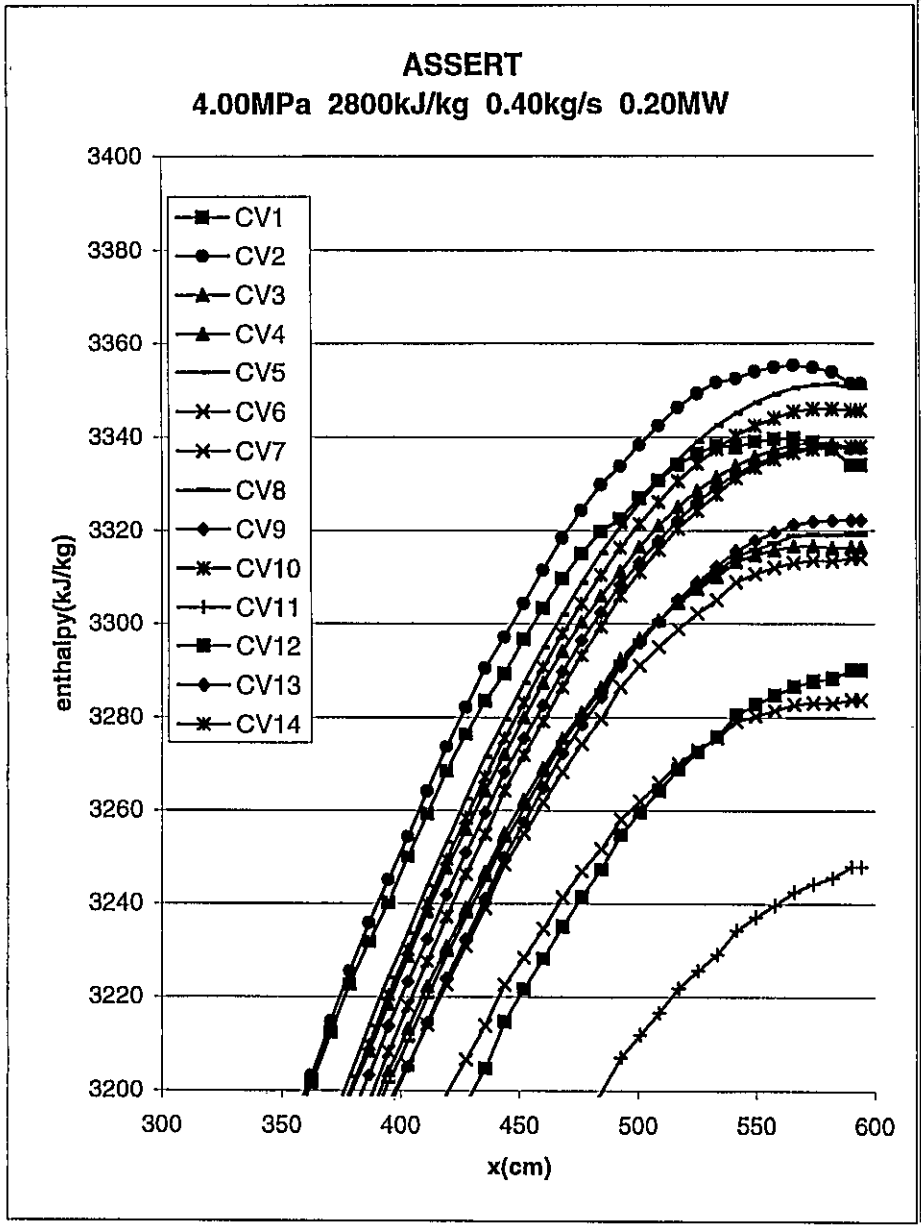

New Model
5.00MPa 2800kJ/kg 0.04kg/s 0.08MW
4880
4830
4780
4730
4680
4630
4580
4530
4480
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

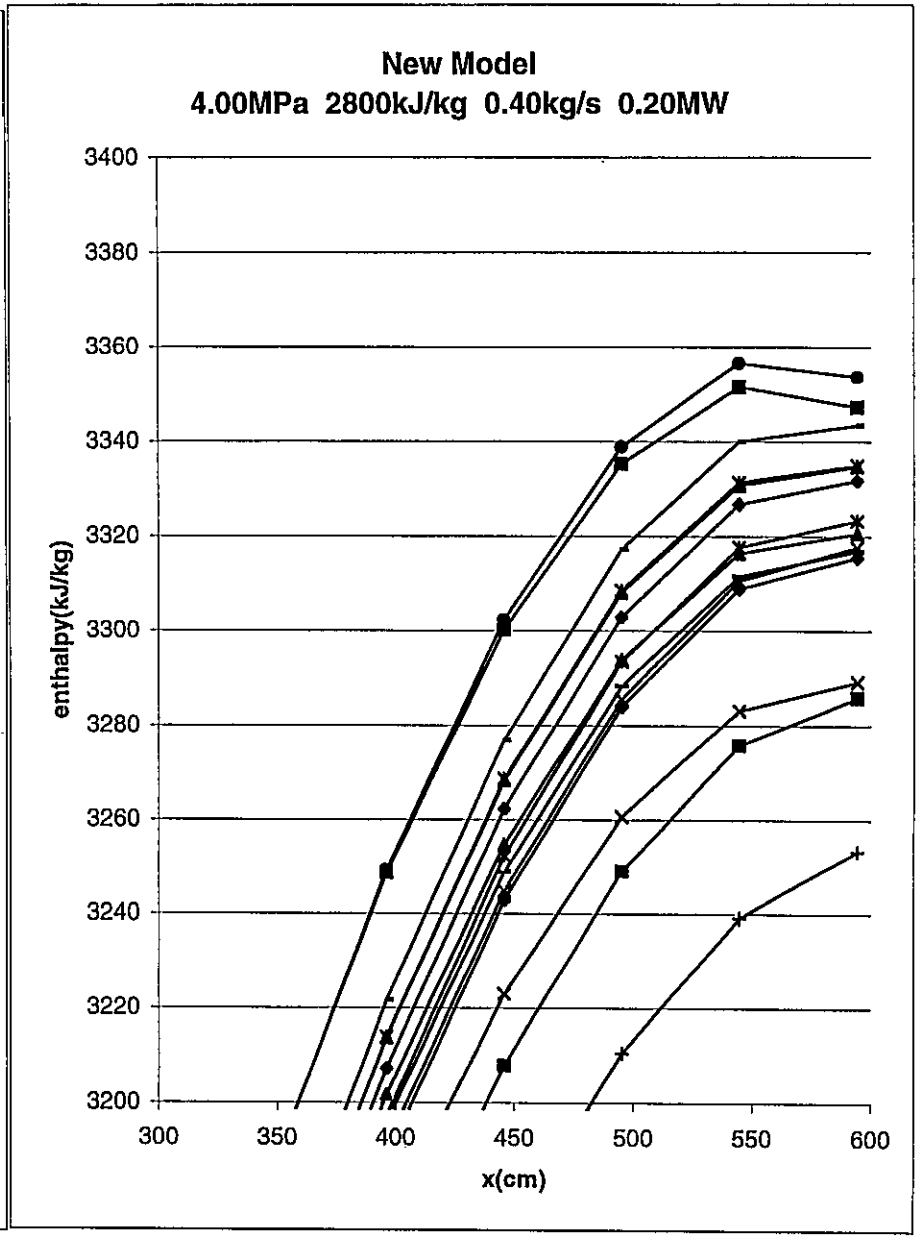

ASSERT
4.00MPa 2800kJ/kg 2.50kg/s 2.50MW
enthalpy(kJ/kg)
4030
3980
3930
3880
3830
3780
3730
3680
3630
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

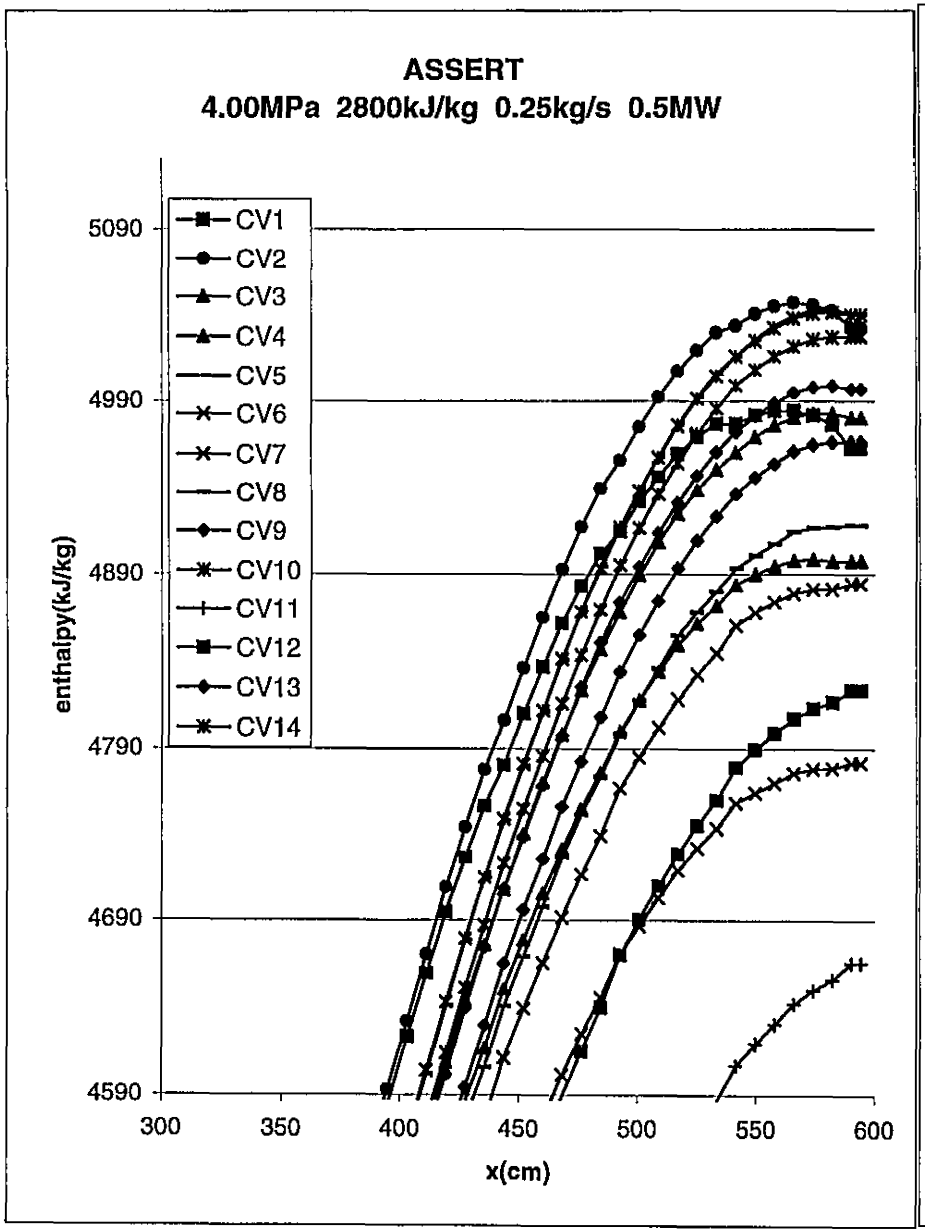

New Model
4.00MPa 2800kJ/kg 2.50kg/s 2.50MW
enthalpy(kJ/kg)
4030
3980
3930
3880
3830
3780
3730
3680
3630
300
350
400
450
500
550
600
x(cm)

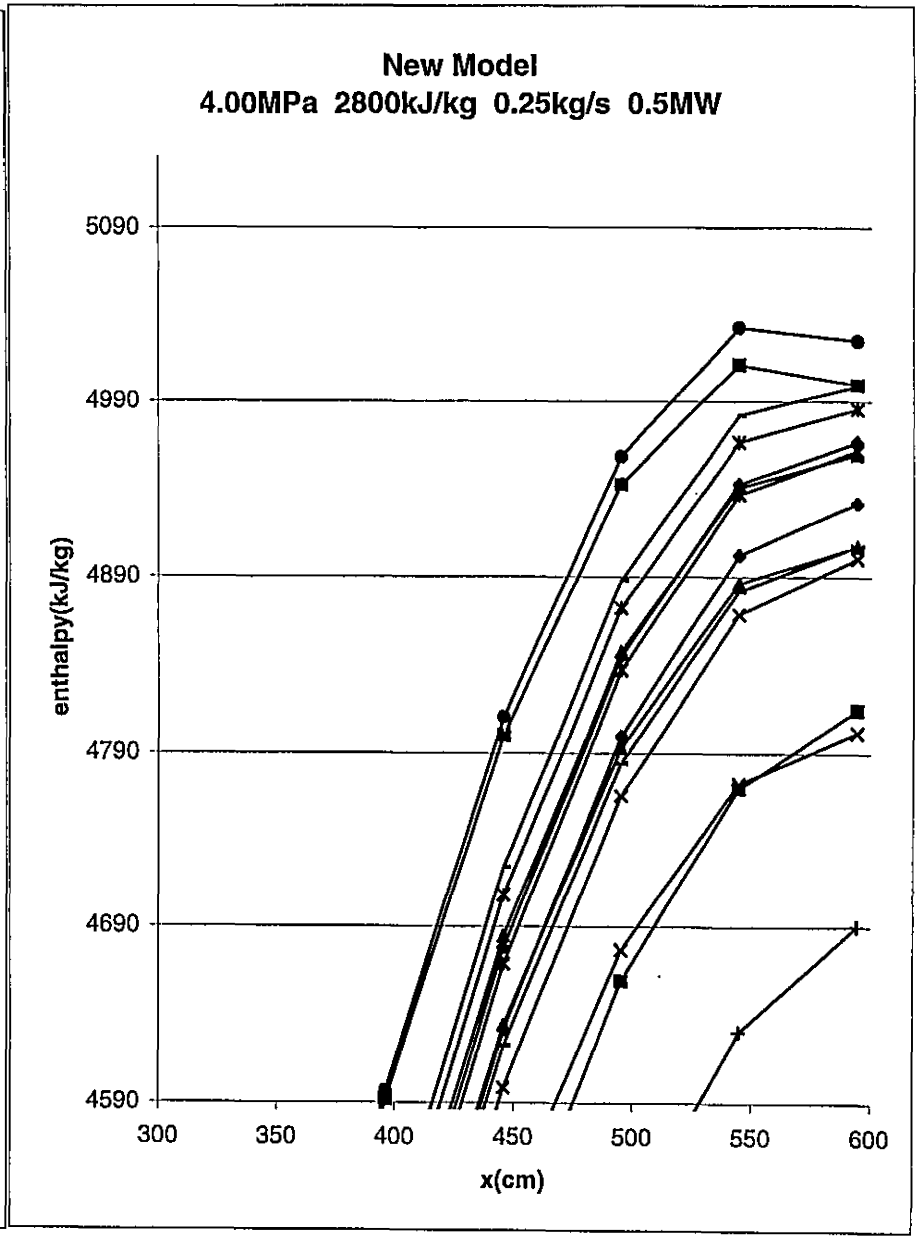

ASSERT
4.00MPa 2800kJ/kg 0.65kg/s 0.08125MW
2950
2940
2930
2920
2910
2900
2890
2880
enthalpy(kJ/kg)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
300
350
400
450
500
550
600
x(cm)

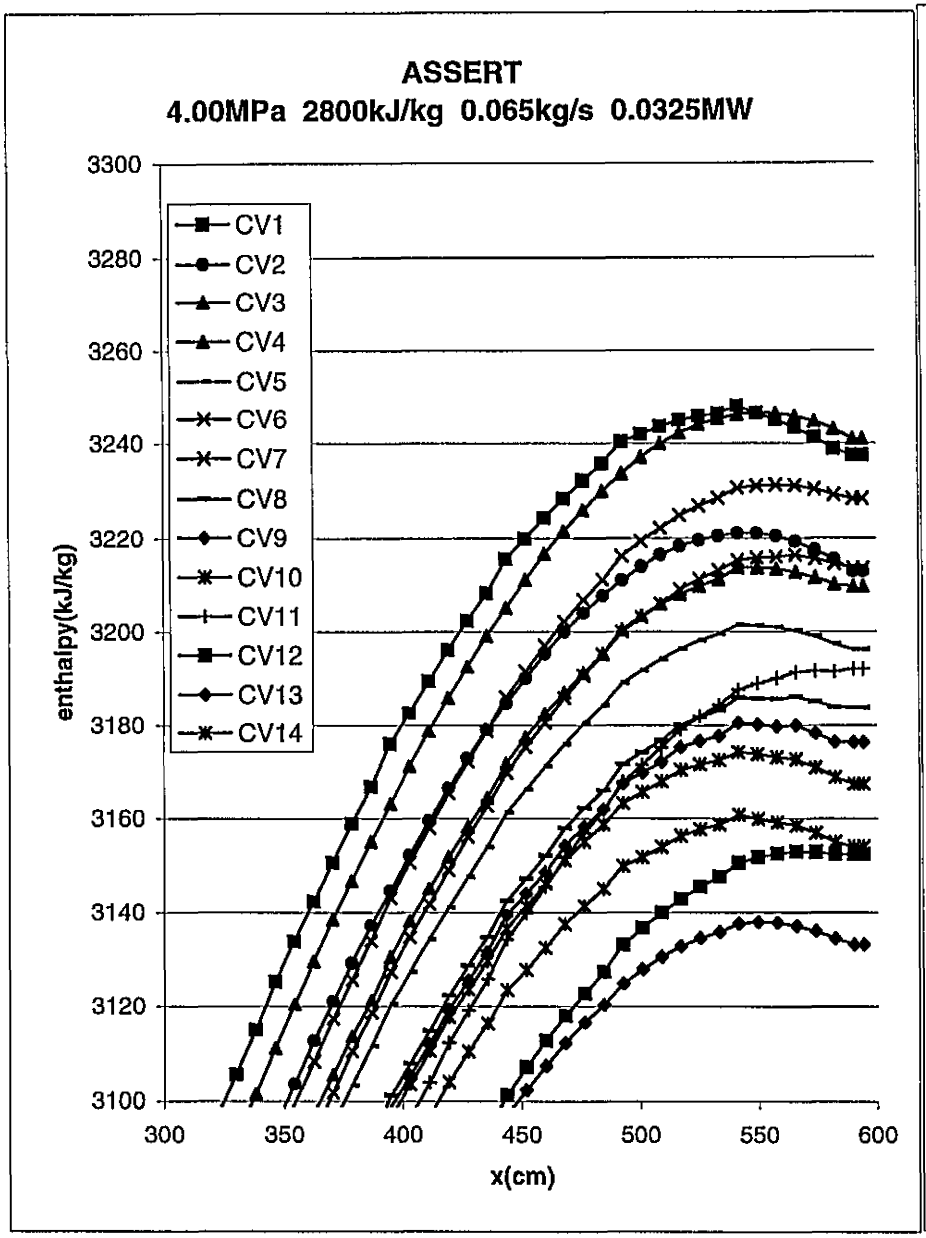

New Model
4.00MPa 2800kJ/kg 0.65kg/s 0.08125MW
2950
2940
2930
2920
2910
2900
2890
2880
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

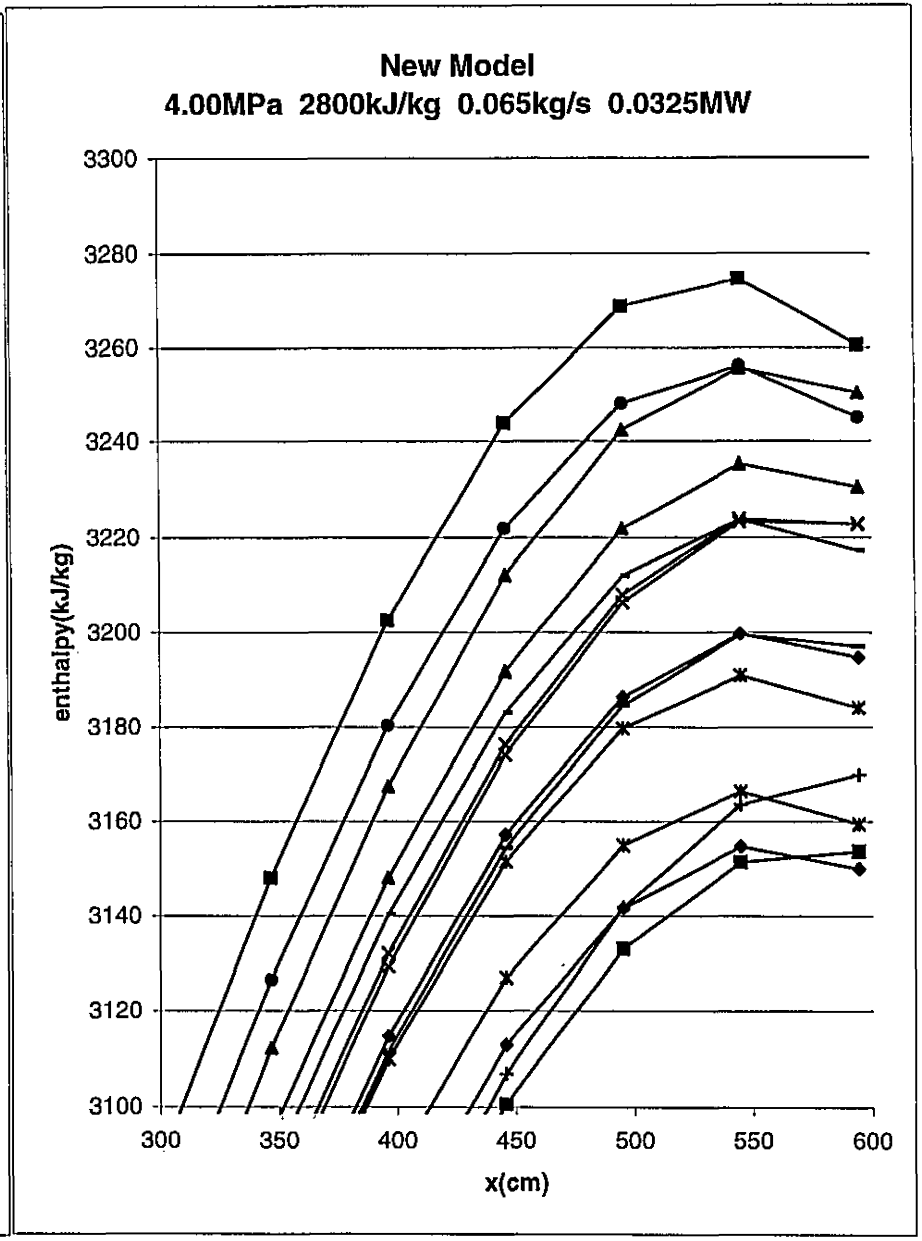

ASSERT
4.00MPa 2800kJ/kg 0.40kg/s 0.20MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3400
3380
3360
3340
3320
3300
3280
3260
3240
3220
3200
300
350
400
450
500
550
600

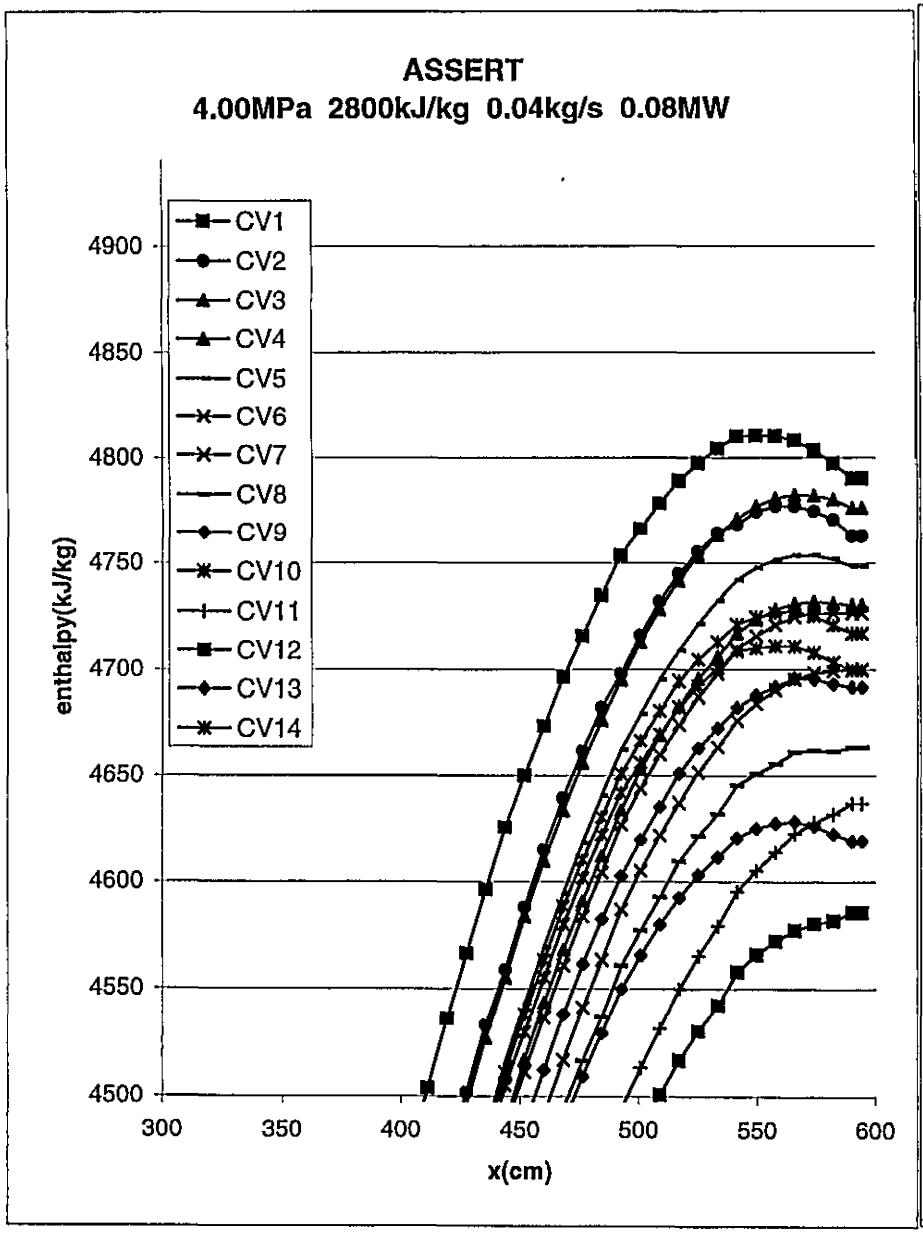

New Model
4.00MPa 2800kJ/kg 0.40kg/s 0.20MW
enthalpy(kJ/kg)
x(cm)
3400
3380
3360
3340
3320
3300
3280
3260
3240
3220
3200
300
350
400
450
500
550
600

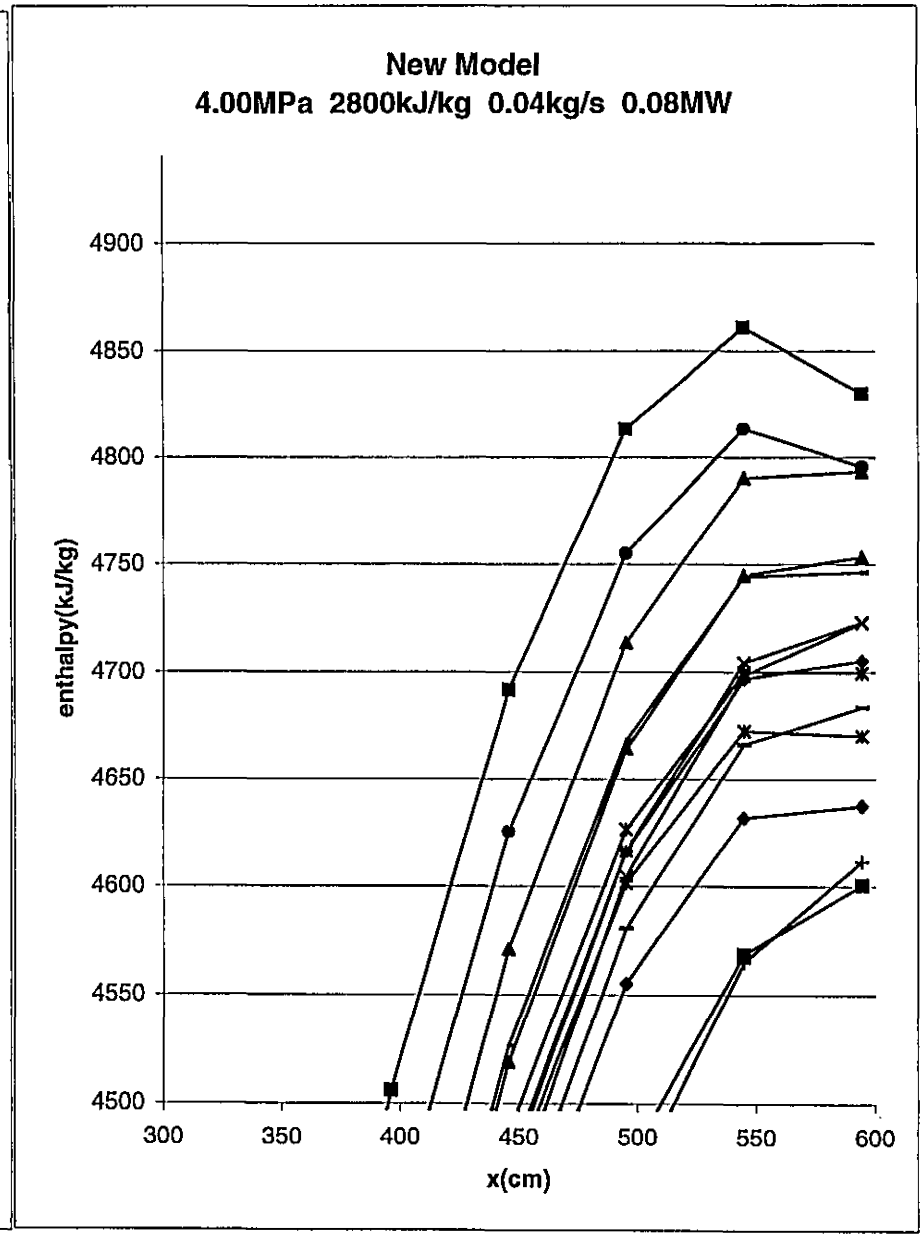

ASSERT
4.00MPa 2800kJ/kg 0.25kg/s 0.5MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
5090
4990
4890
4790
4690
4590
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

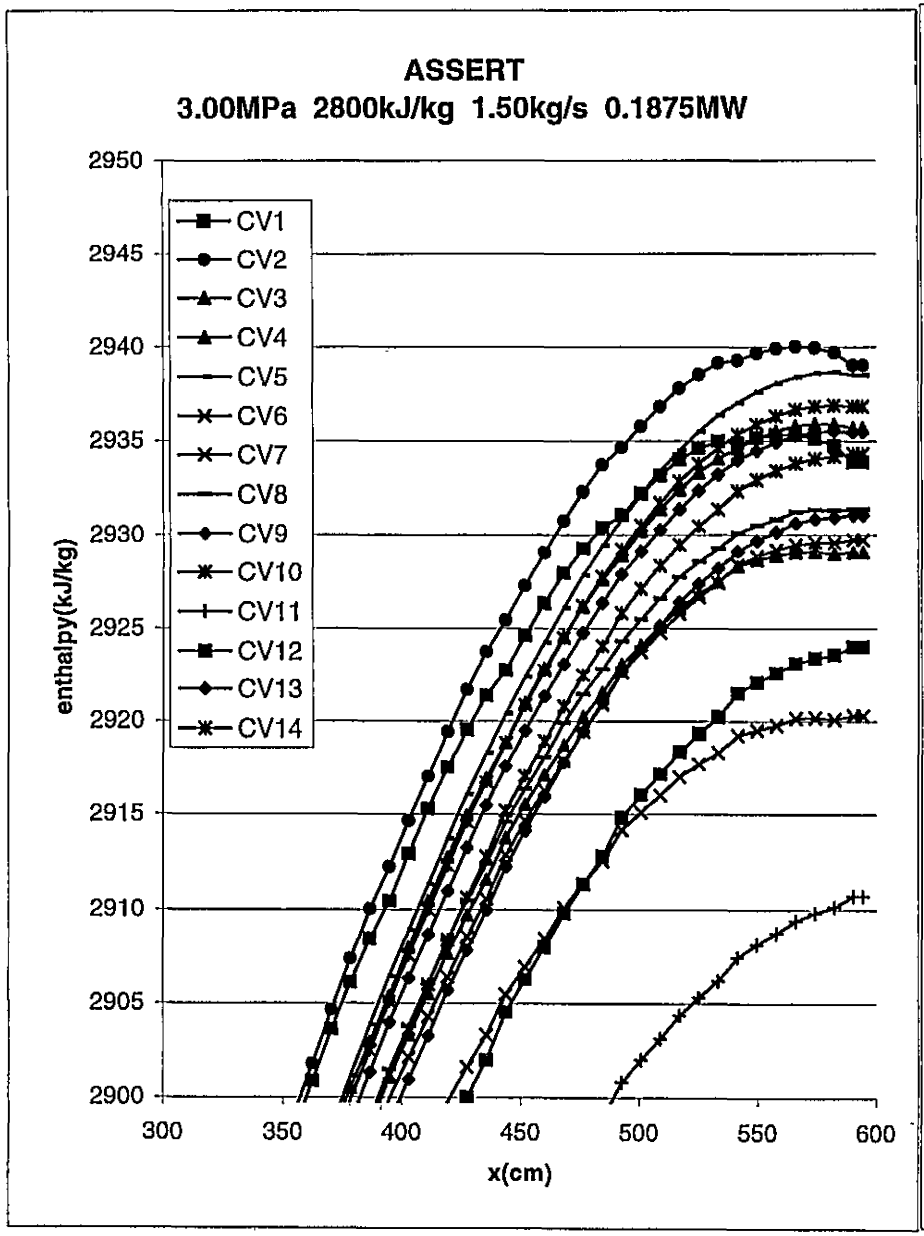

New Model
4.00MPa 2800kJ/kg 0.25kg/s 0.5MW
5090
4990
4890
4790
4690
4590
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

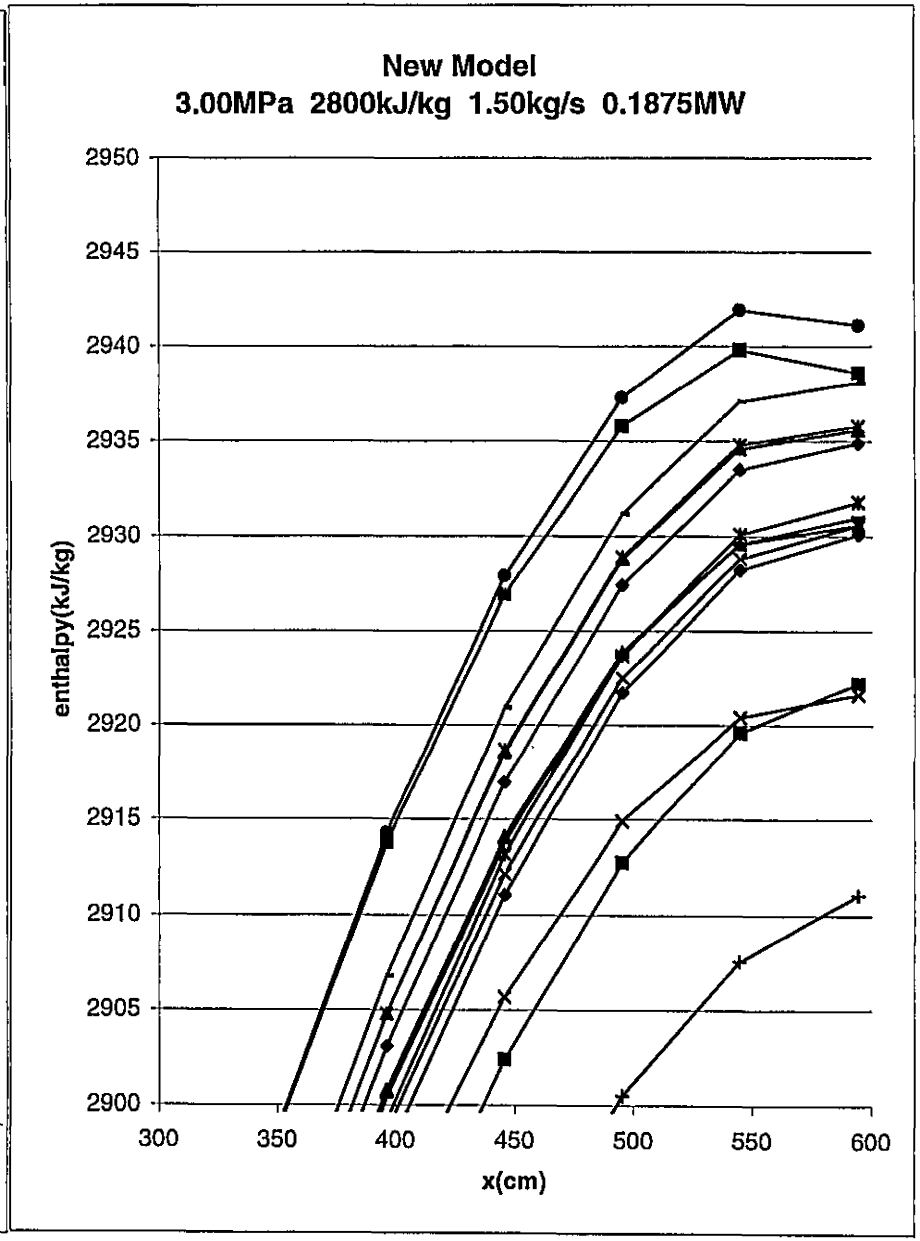

ASSERT
4.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3300
3280
3260
3240
3220
3200
3180
3160
3140
3120
3100
300
350
400
450
500
550
600

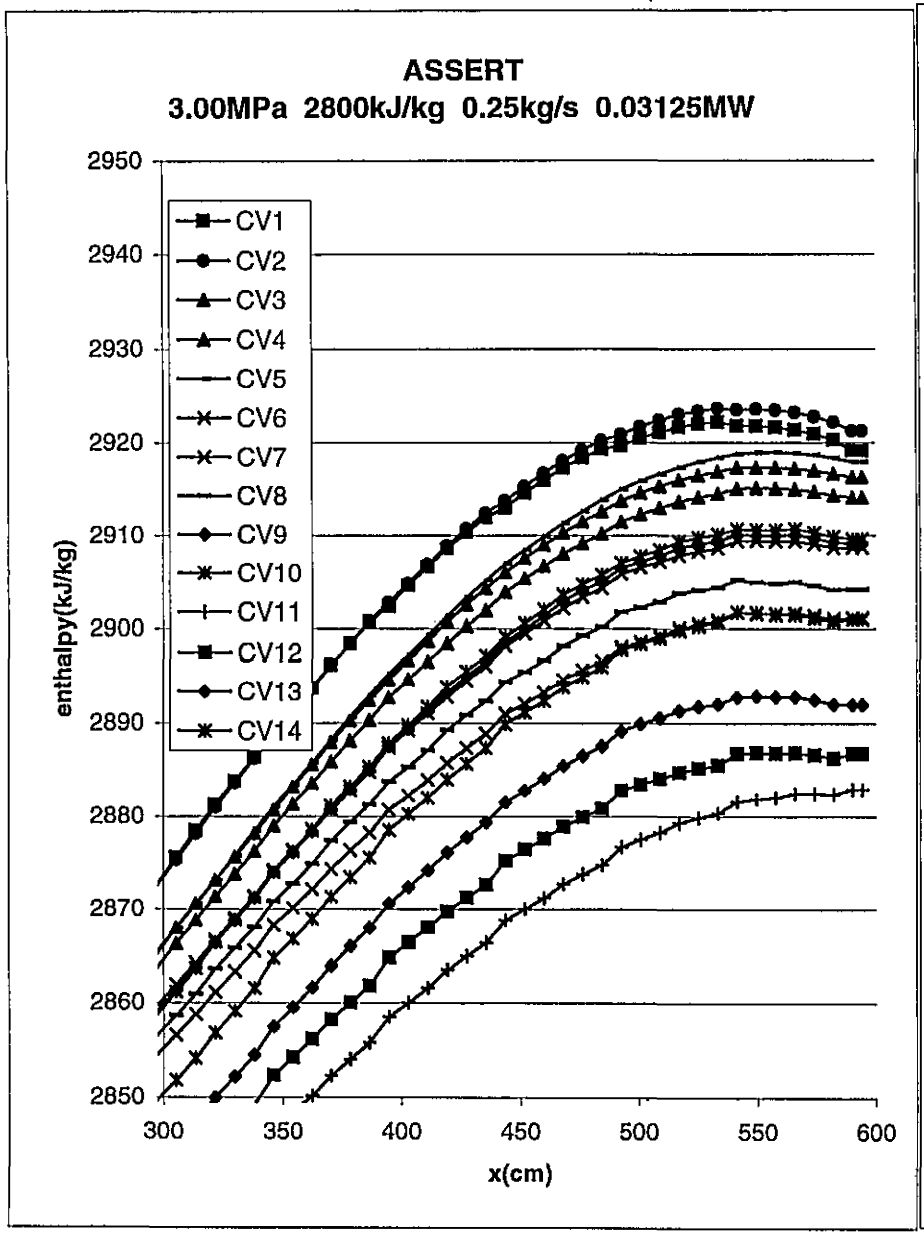

New Model
4.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
enthalpy(kJ/kg)
x(cm)
3300
3280
3260
3240
3220
3200
3180
3160
3140
3120
3100
300
350
400
450
500
550
600

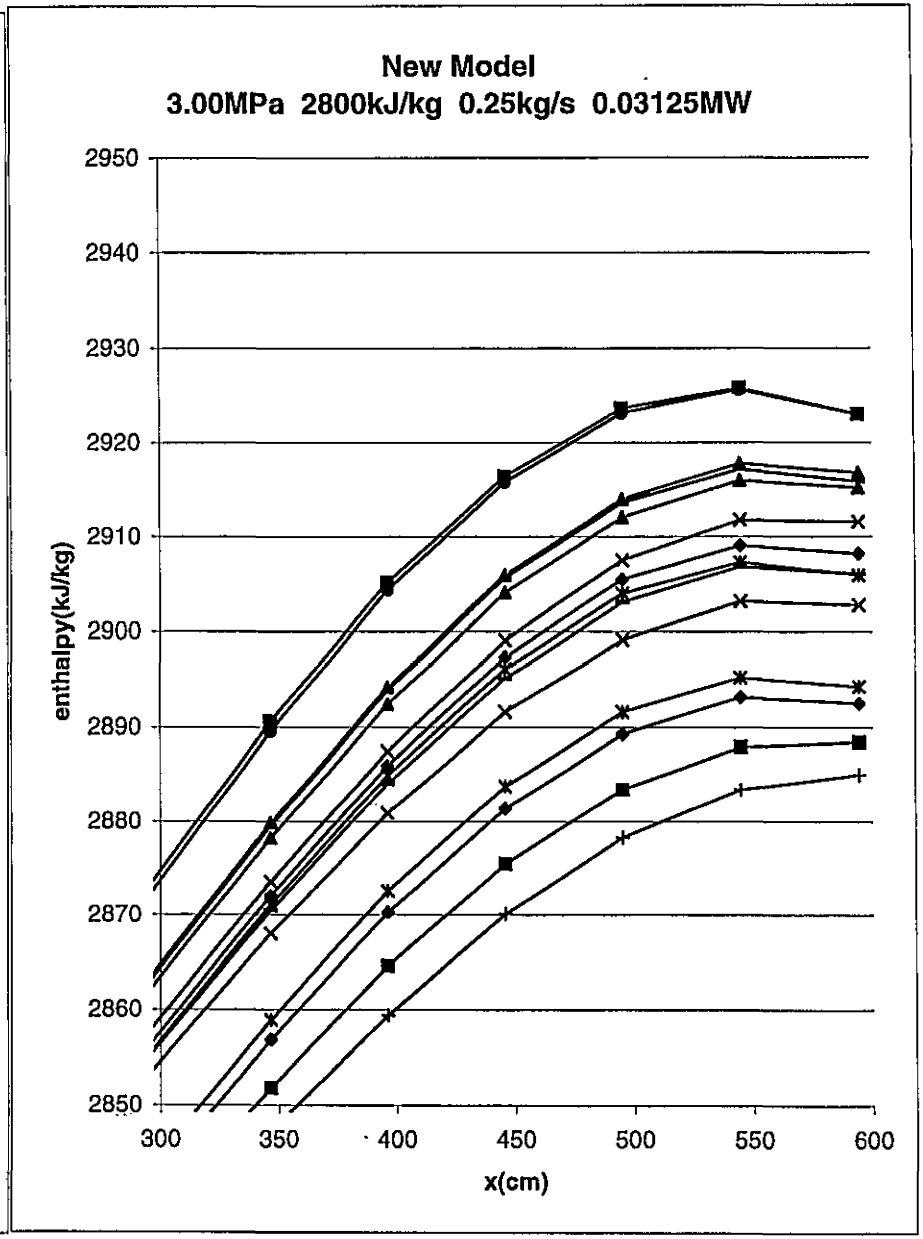

ASSERT
4.00MPa 2800kJ/kg 0.04kg/s 0.08MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
4900
4850
4800
4750
4700
4650
4600
4550
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

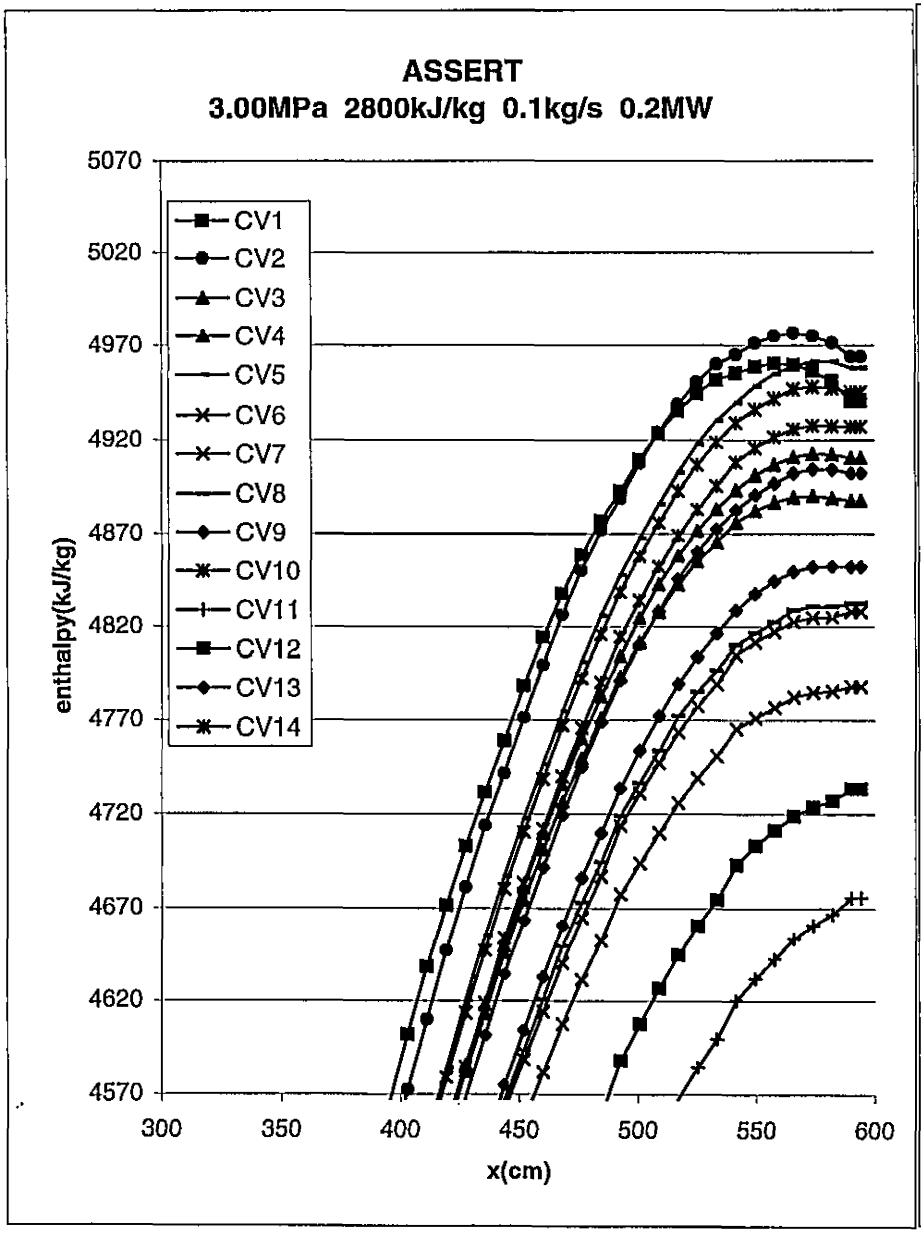

New Model
4.00MPa 2800kJ/kg 0.04kg/s 0.08MW
4900
4850
4800
4750
4700
4650
4600
4550
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

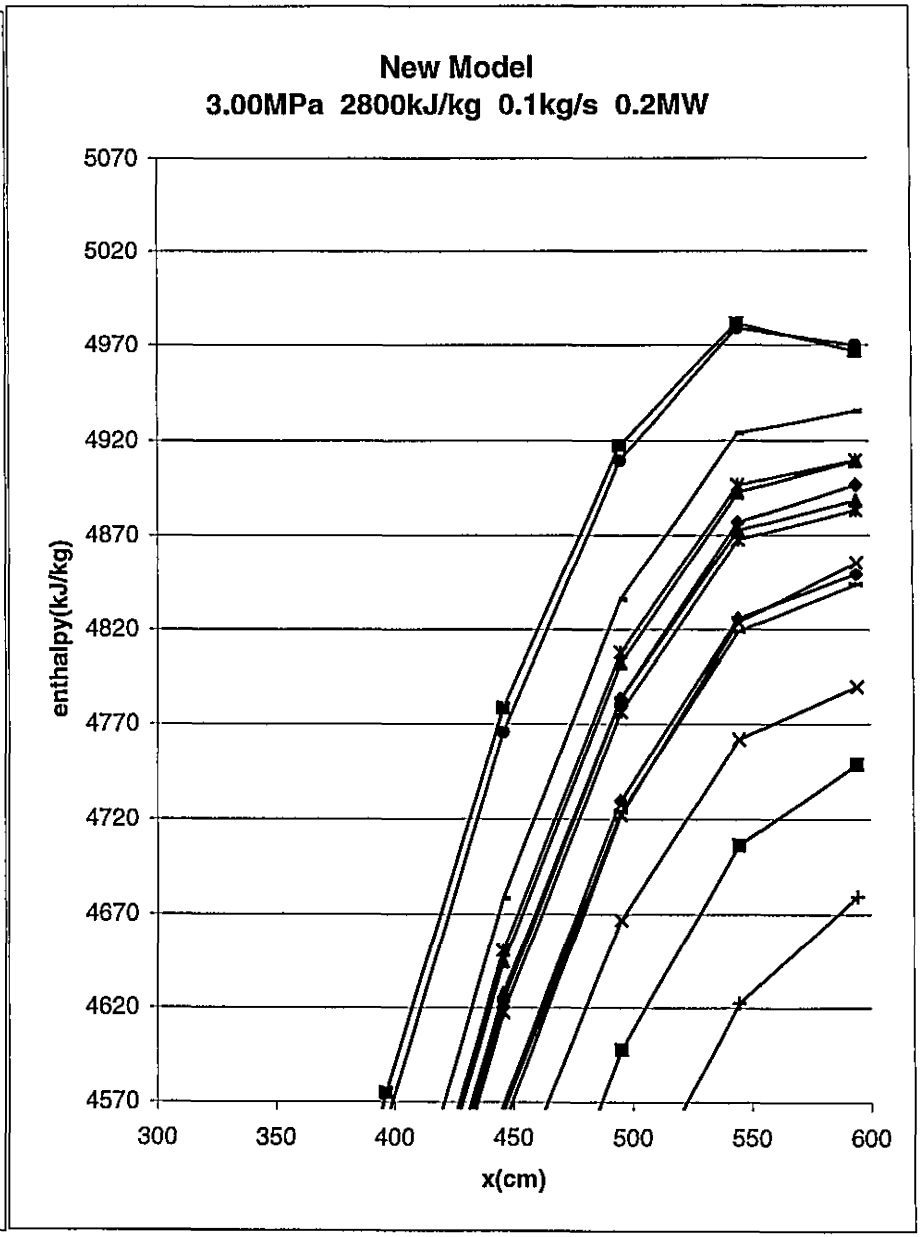

ASSERT
3.00MPa 2800kJ/kg 1.50kg/s 0.1875MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

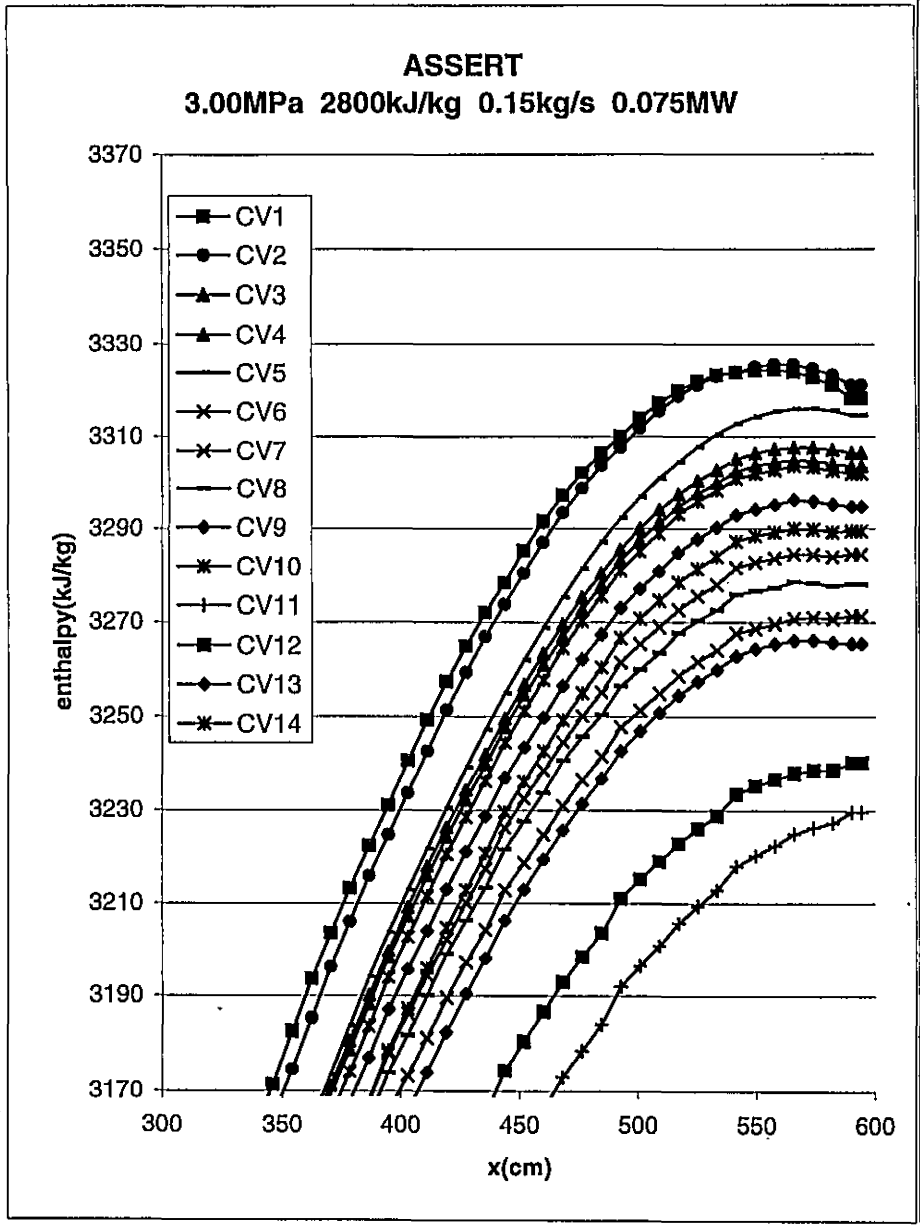

New Model
3.00MPa 2800kJ/kg 1.50kg/s 0.1875MW
enthalpy(kJ/kg)
x(cm)
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

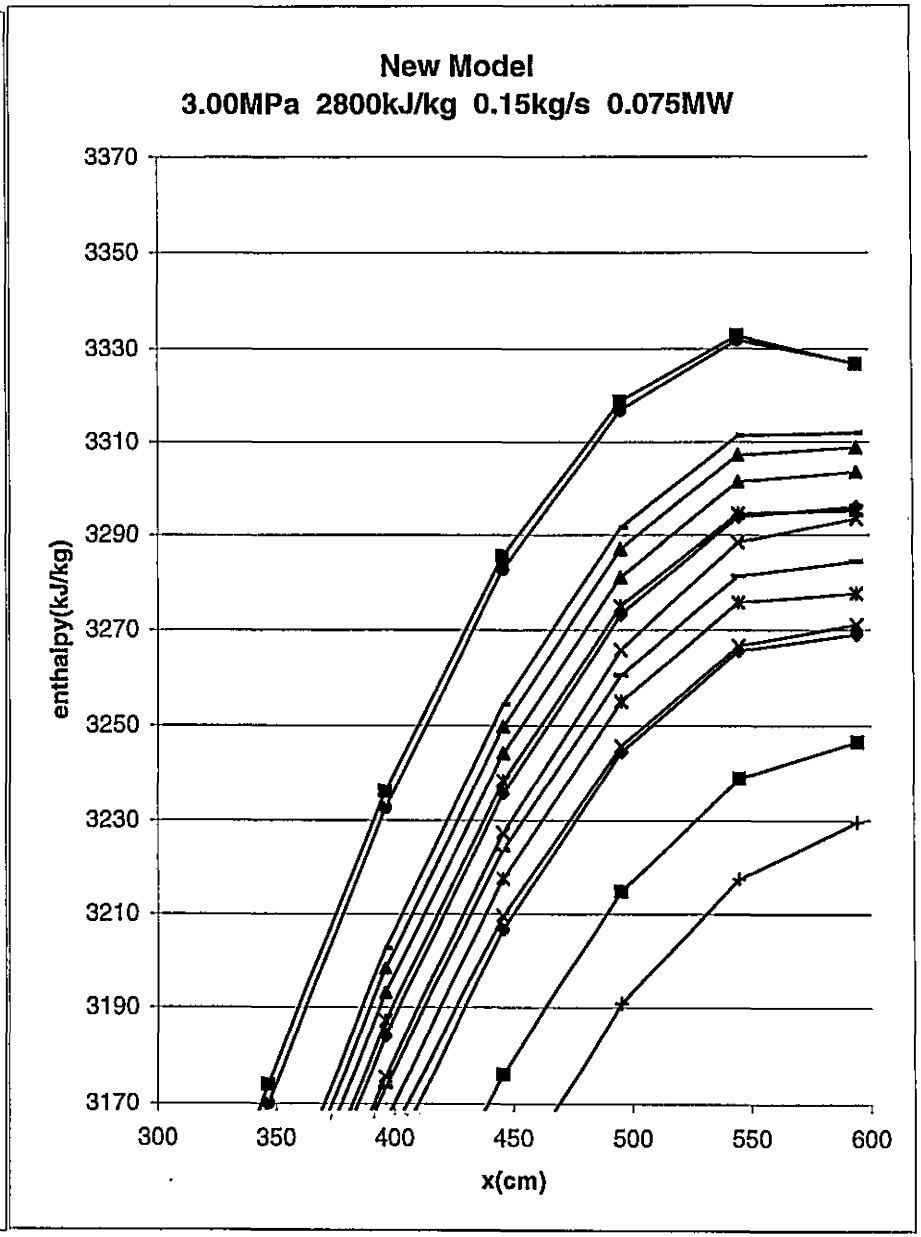

ASSERT
3.00MPa 2800kJ/kg 0.25kg/s 0.03125MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
enthalpy(kJ/kg)
2950
2940
2930
2920
2910
2900
2890
2880
2870
2860
2850
300
350
400
450
500
550
600
x(cm)

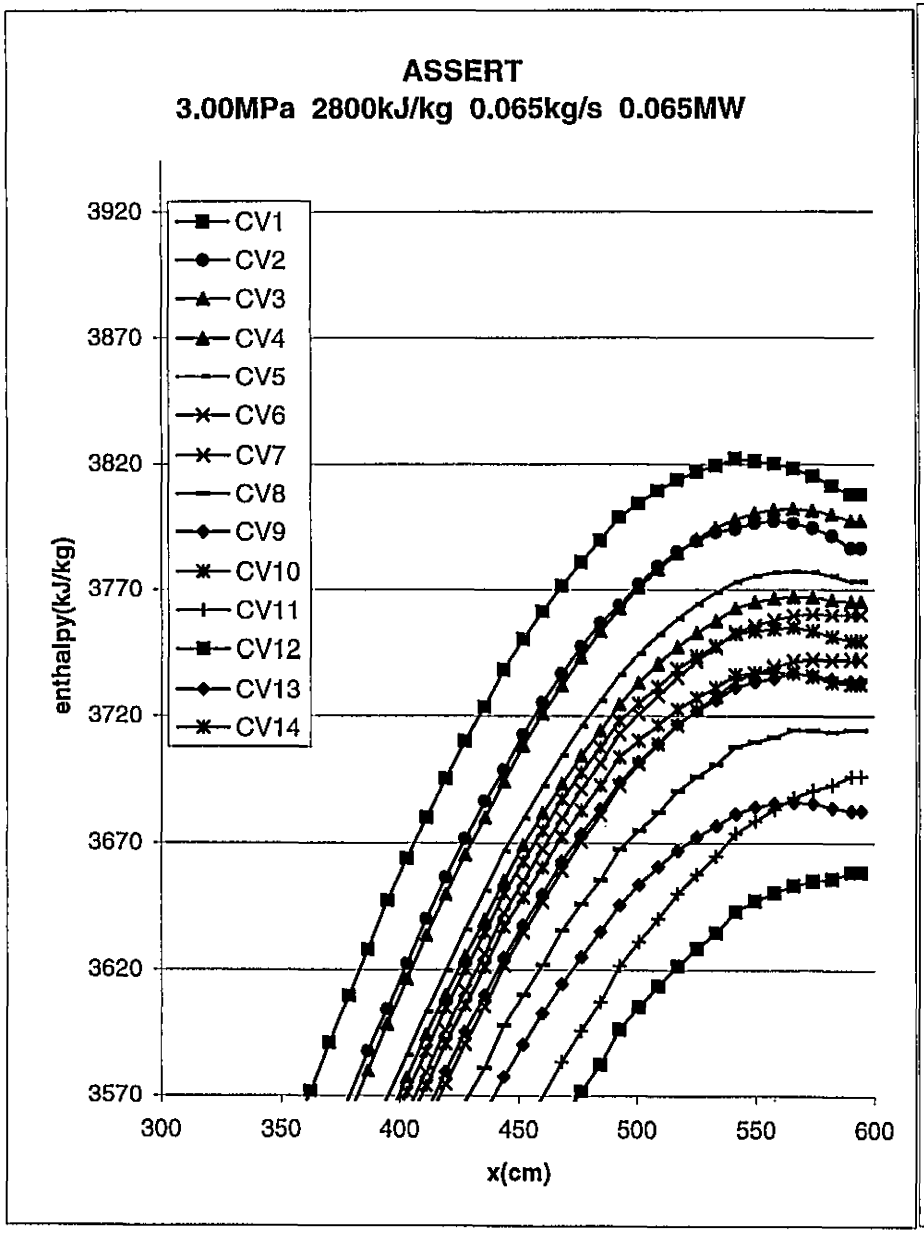

New Model
3.00MPa 2800kJ/kg 0.25kg/s 0.03125MW
enthalpy(kJ/kg)
2950
2940
2930
2920
2910
2900
2890
2880
2870
2860
2850
300
350
400
450
500
550
600
x(cm)

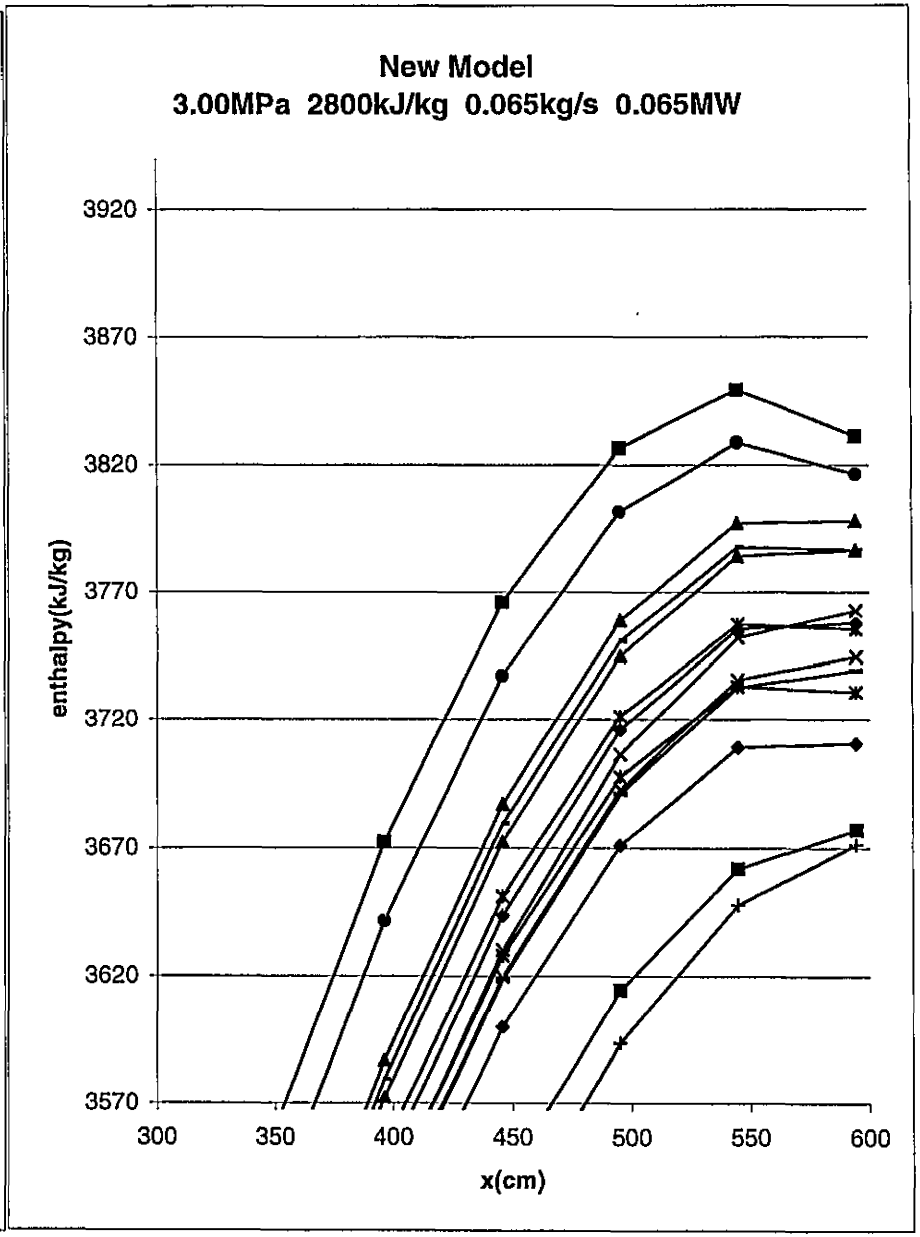

ASSERT
3.00MPa 2800kJ/kg 0.1kg/s 0.2MW
5070
5020
4970
4920
4870
4820
4770
4720
4670
4620
4570
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

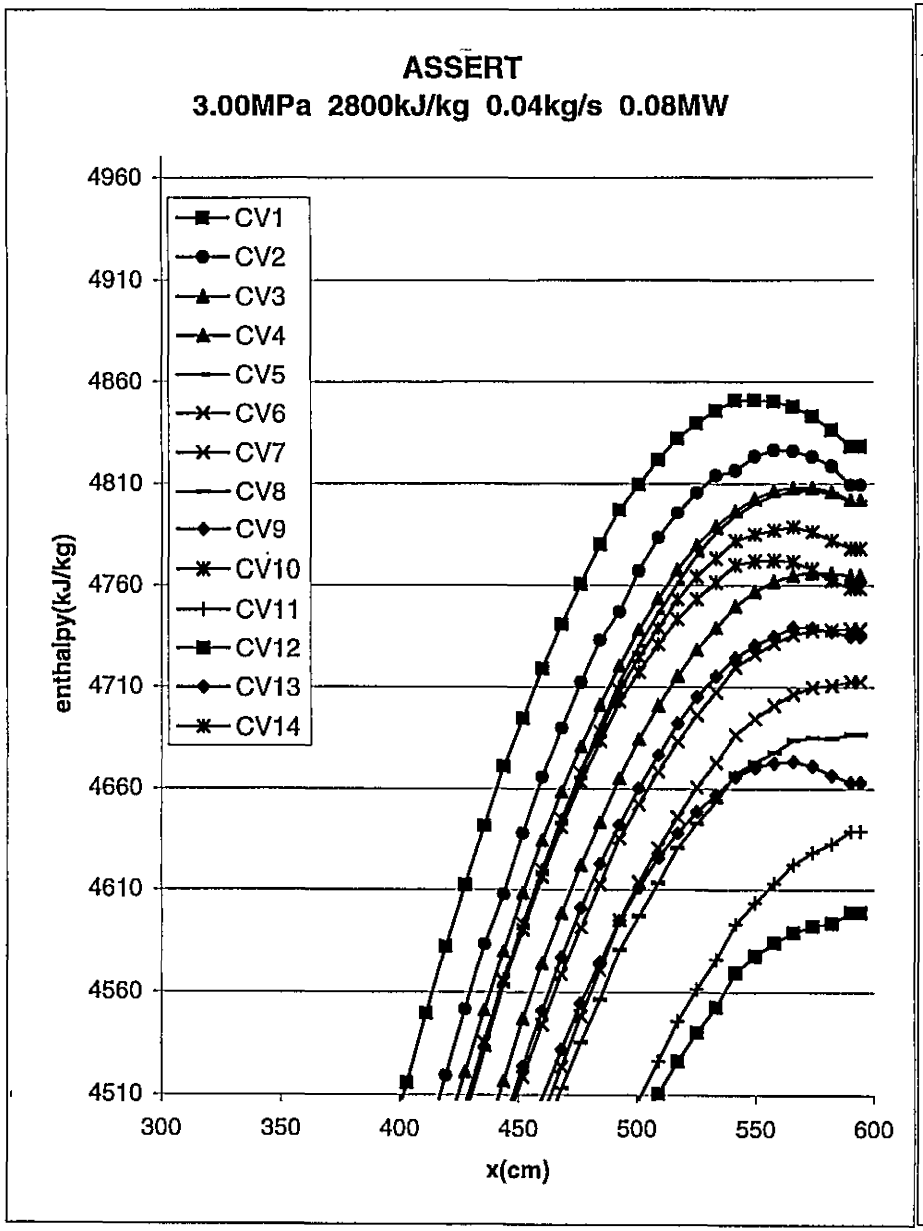

New Model
3.00MPa 2800kJ/kg 0.1kg/s 0.2MW
5070
5020
4970
4920
4870
4820
4770
4720
4670
4620
4570
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

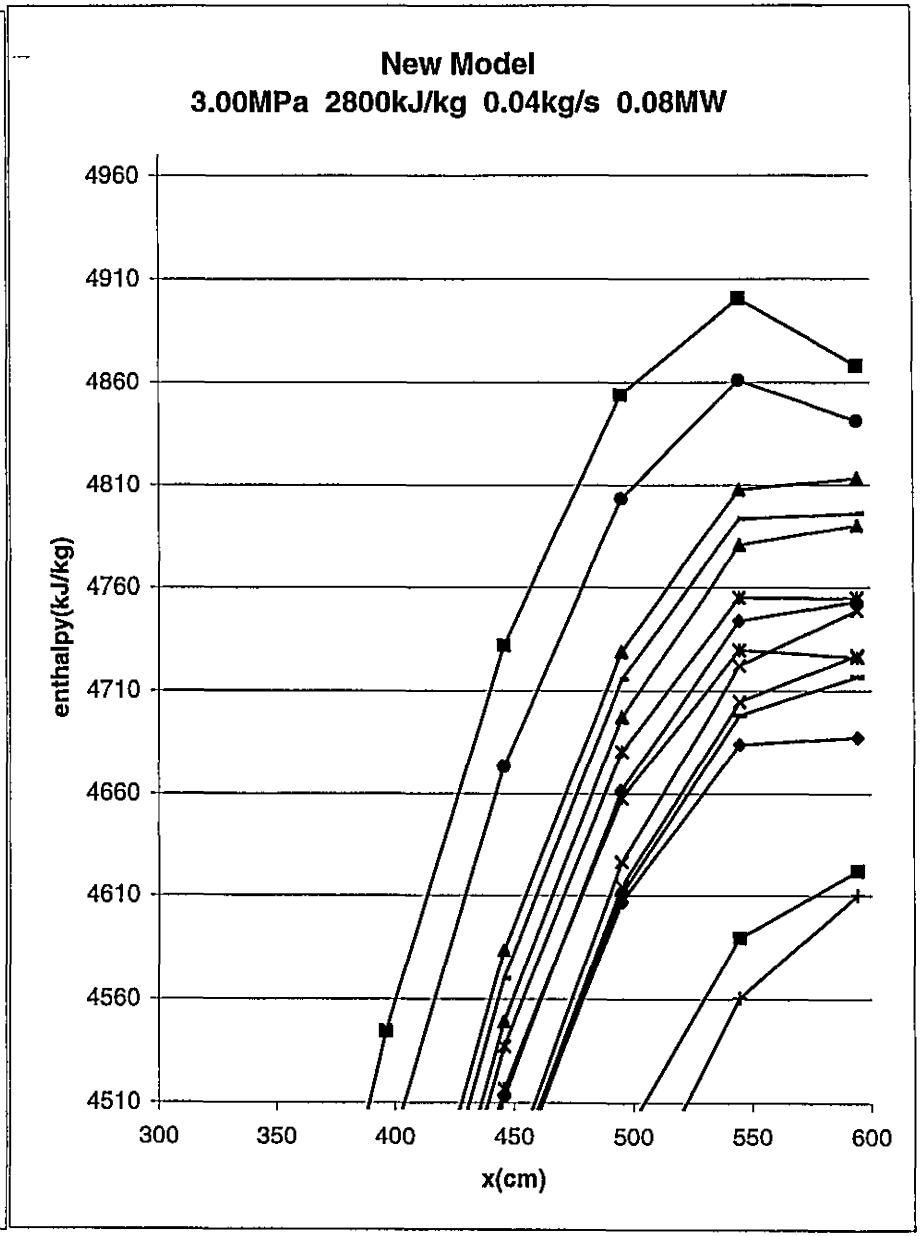

ASSERT
3.00MPa 2800kJ/kg 0.15kg/s 0.075MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3370
3350
3330
3310
3290
3270
3250
3230
3210
3190
3170
300
350
400
450
500
550
600

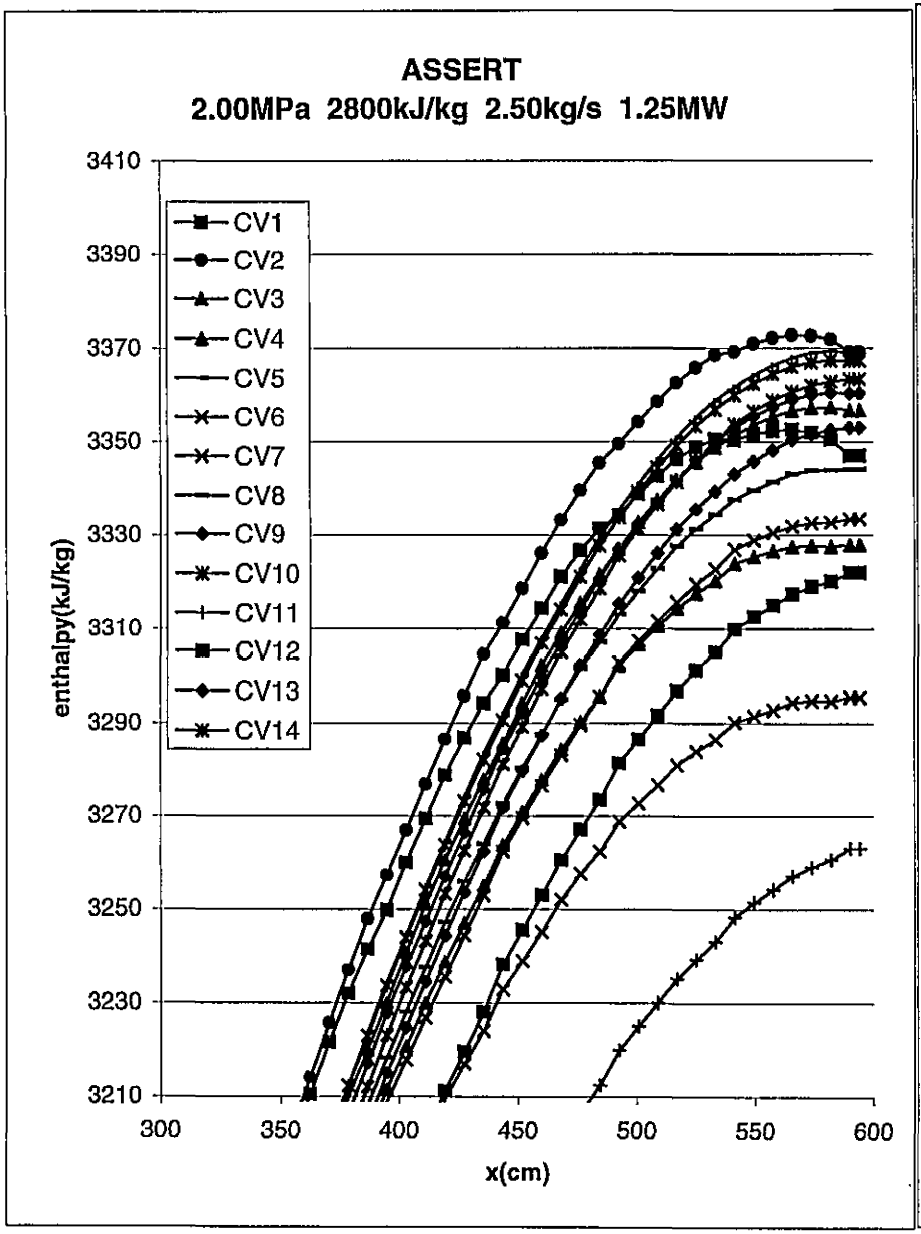

New Model
3.00MPa 2800kJ/kg 0.15kg/s 0.075MW
enthalpy(kJ/kg)
x(cm)
3370
3350
3330
3310
3290
3270
3250
3230
3210
3190
3170
300
350
400
450
500
550
600

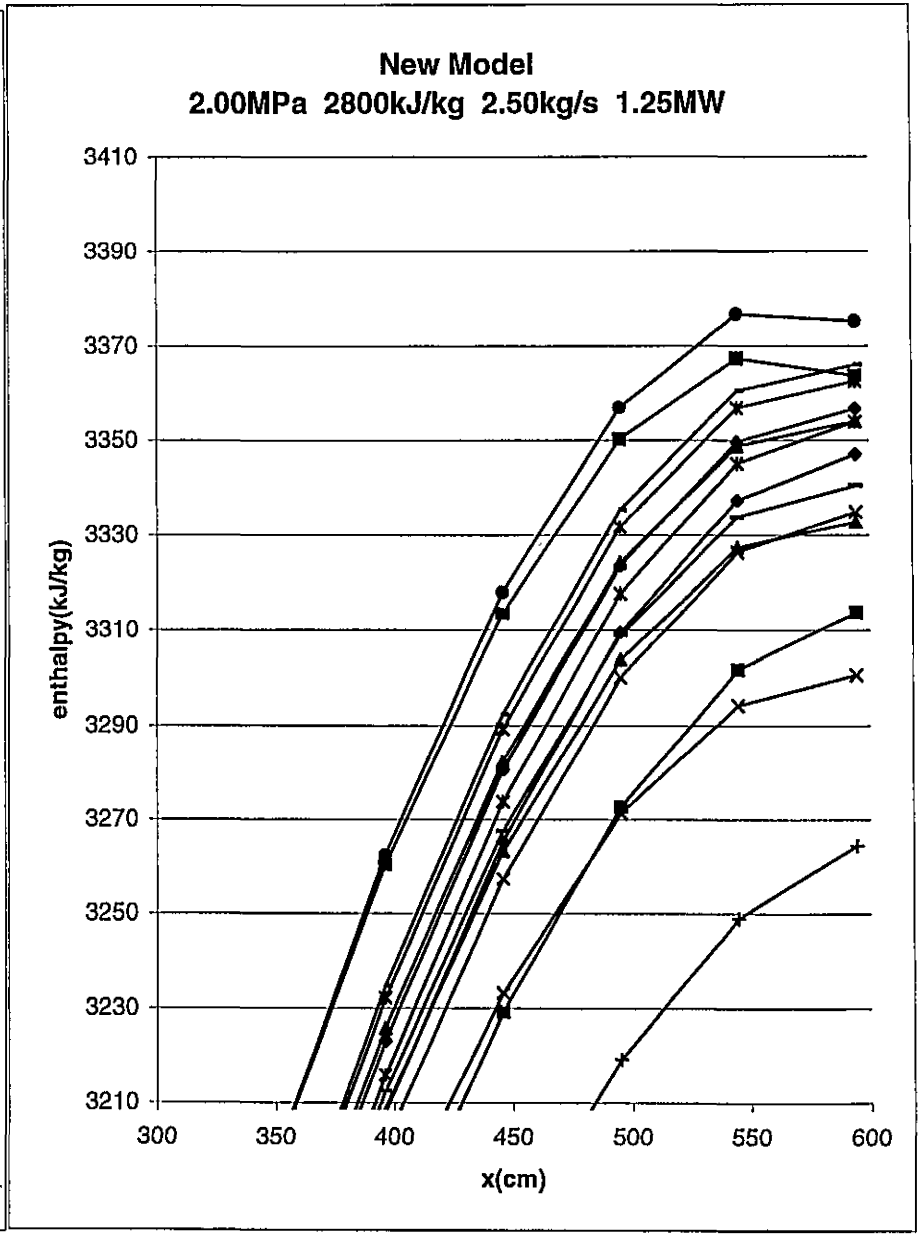

ASSERT
3.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
3920
3870
3820
3770
3720
3670
3620
3570
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

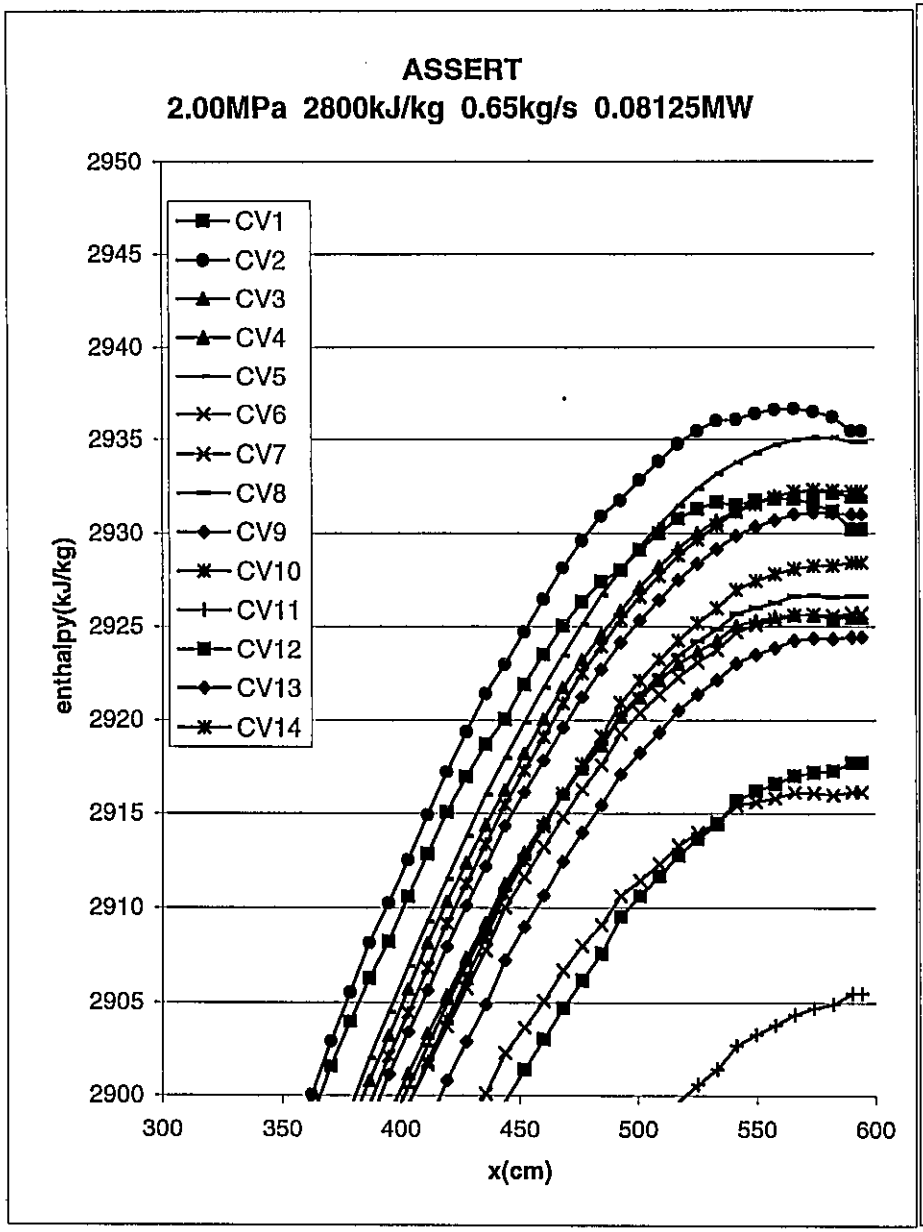

New Model
3.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
3920
3870
3820
3770
3720
3670
3620
3570
300
350
400
450
500
550
600
x(cm)

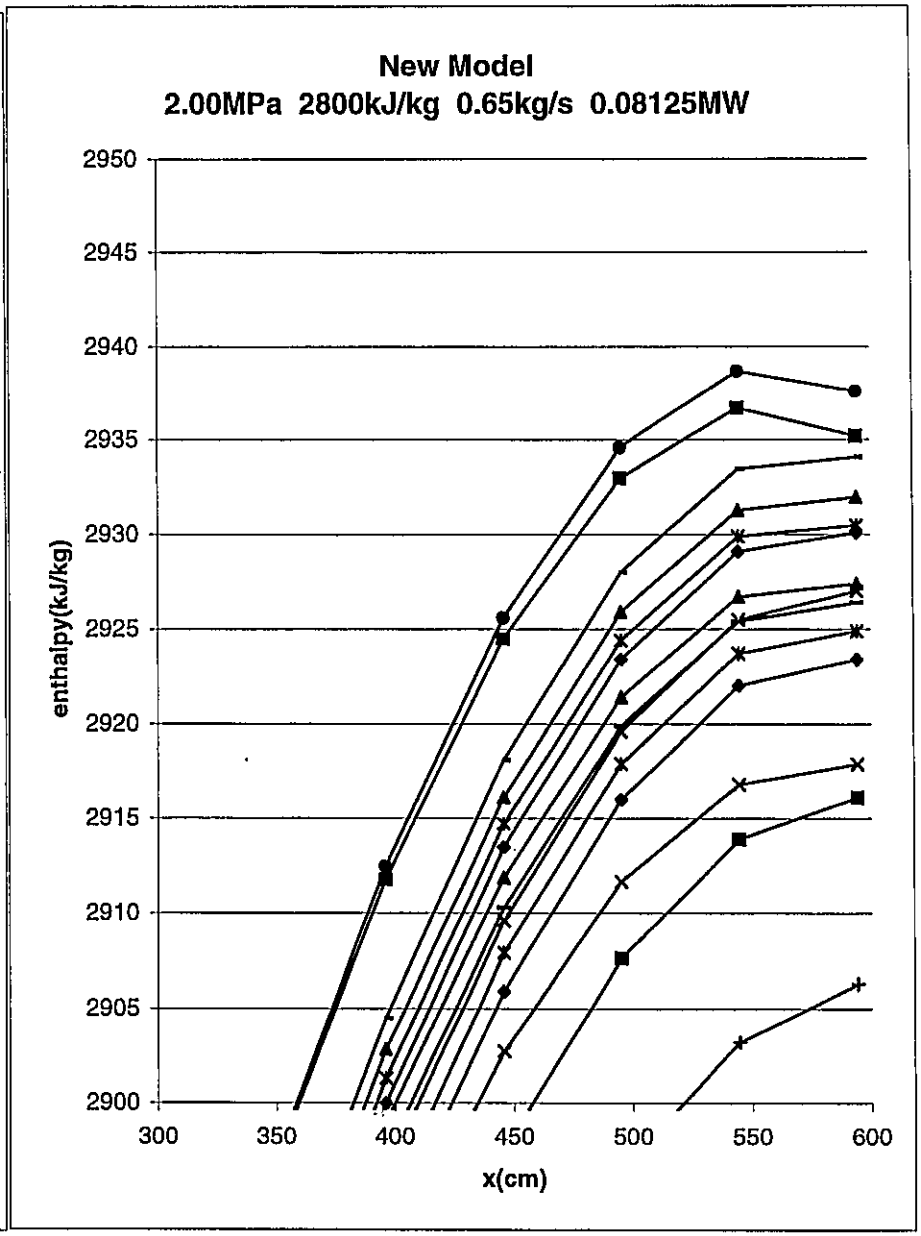

ASSERT
3.00MPa 2800kJ/kg 0.04kg/s 0.08MW
enthalpy(kJ/kg)
4960
4910
4860
4810
4760
4710
4660
4610
4560
4510
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

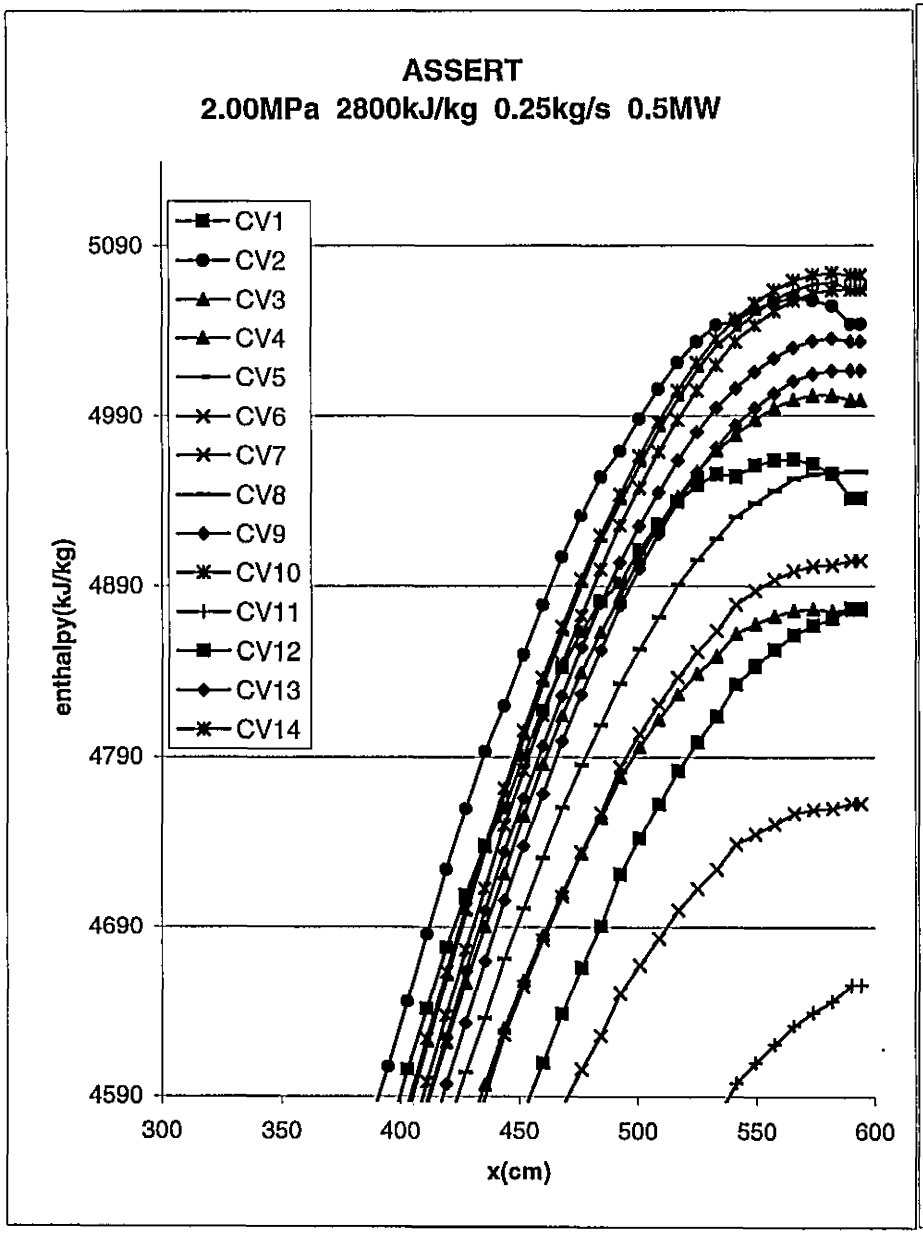

New Model
3.00MPa 2800kJ/kg 0.04kg/s 0.08MW
enthalpy(kJ/kg)
4960
4910
4860
4810
4760
4710
4660
4610
4560
4510
300
350
400
450
500
550
600
x(cm)

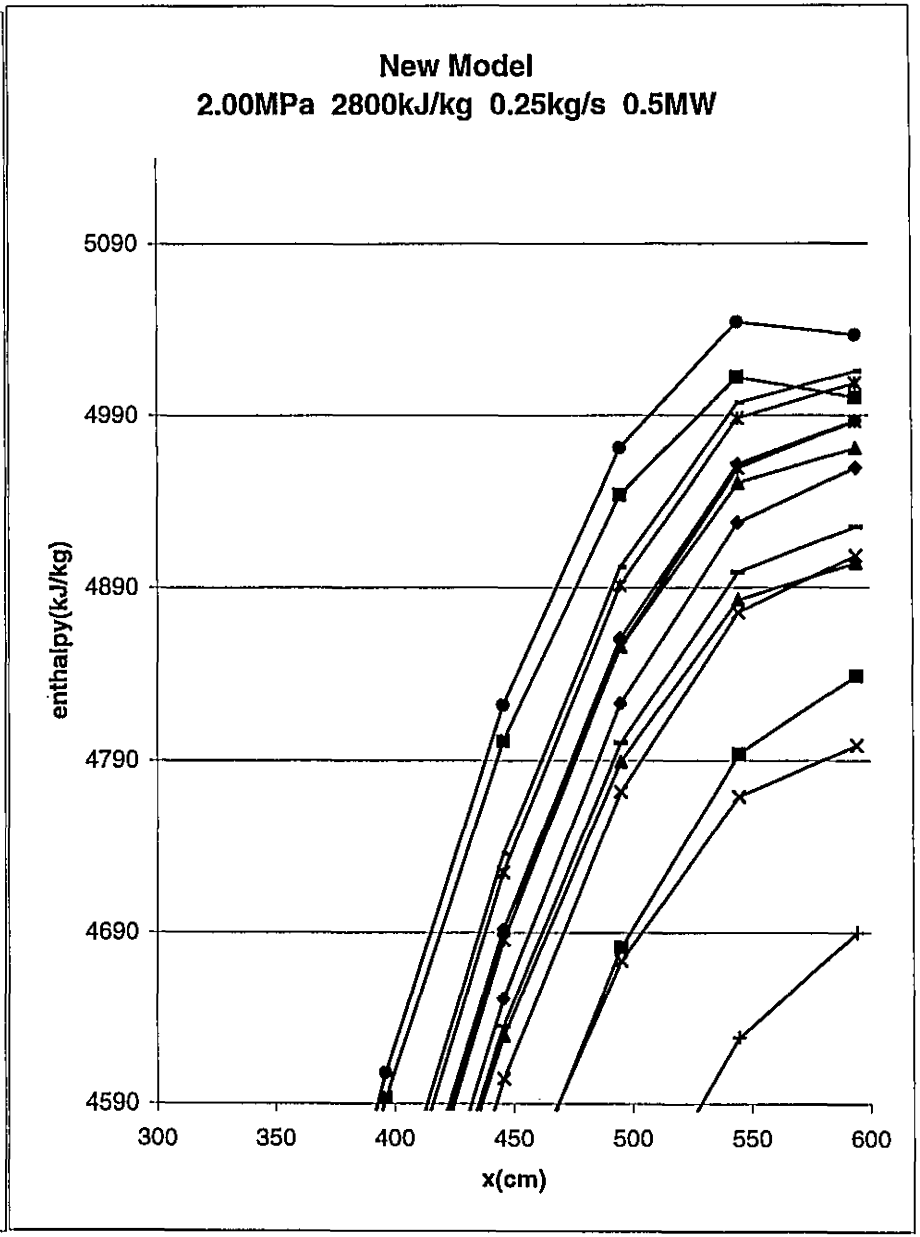

ASSERT
2.00MPa 2800kJ/kg 2.50kg/s 1.25MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3410
3390
3370
3350
3330
3310
3290
3270
3250
3230
3210
300
350
400
450
500
550
600

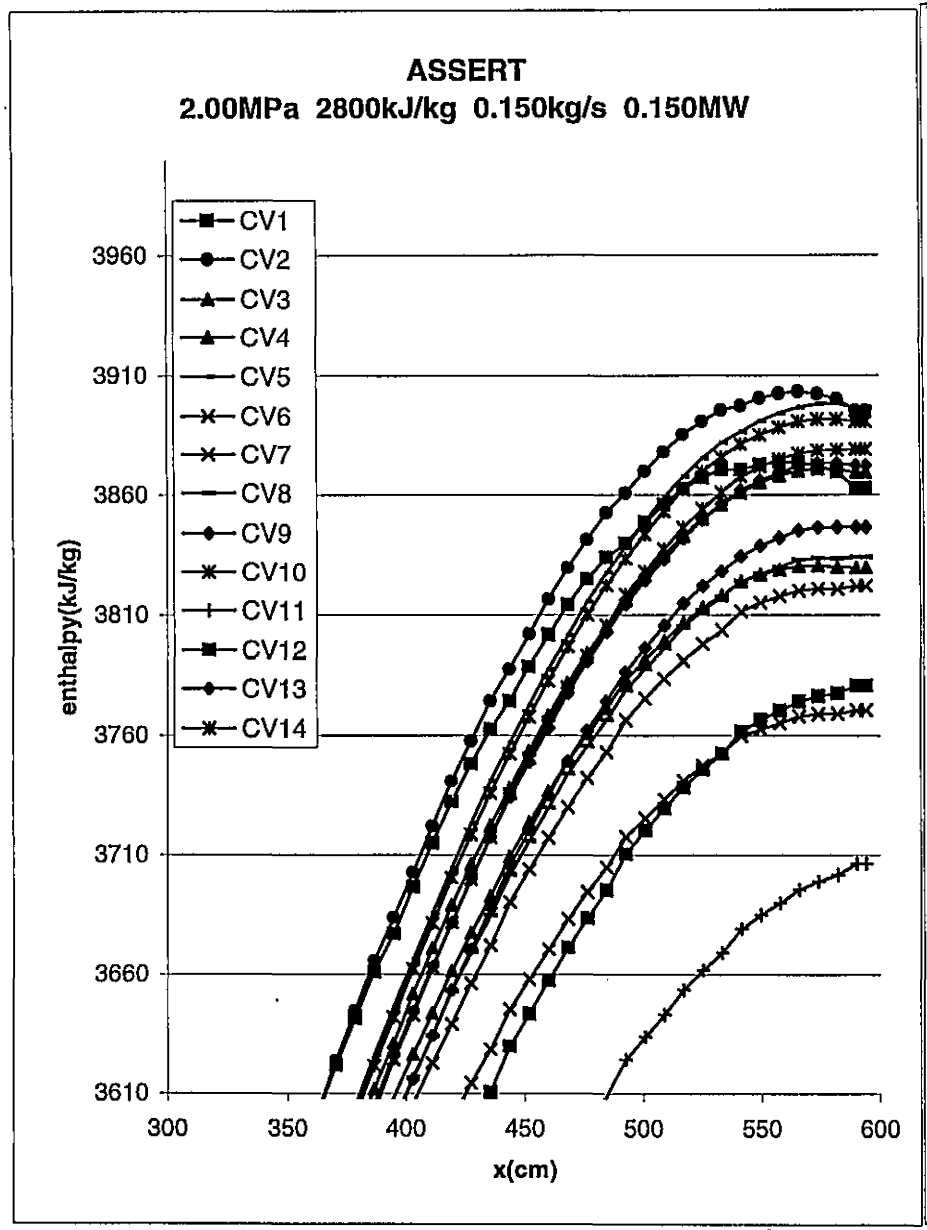

New Model
2.00MPa 2800kJ/kg 2.50kg/s 1.25MW
enthalpy(kJ/kg)
x(cm)
3410
3390
3370
3350
3330
3310
3290
3270
3250
3230
3210
300
350
400
450
500
550
600

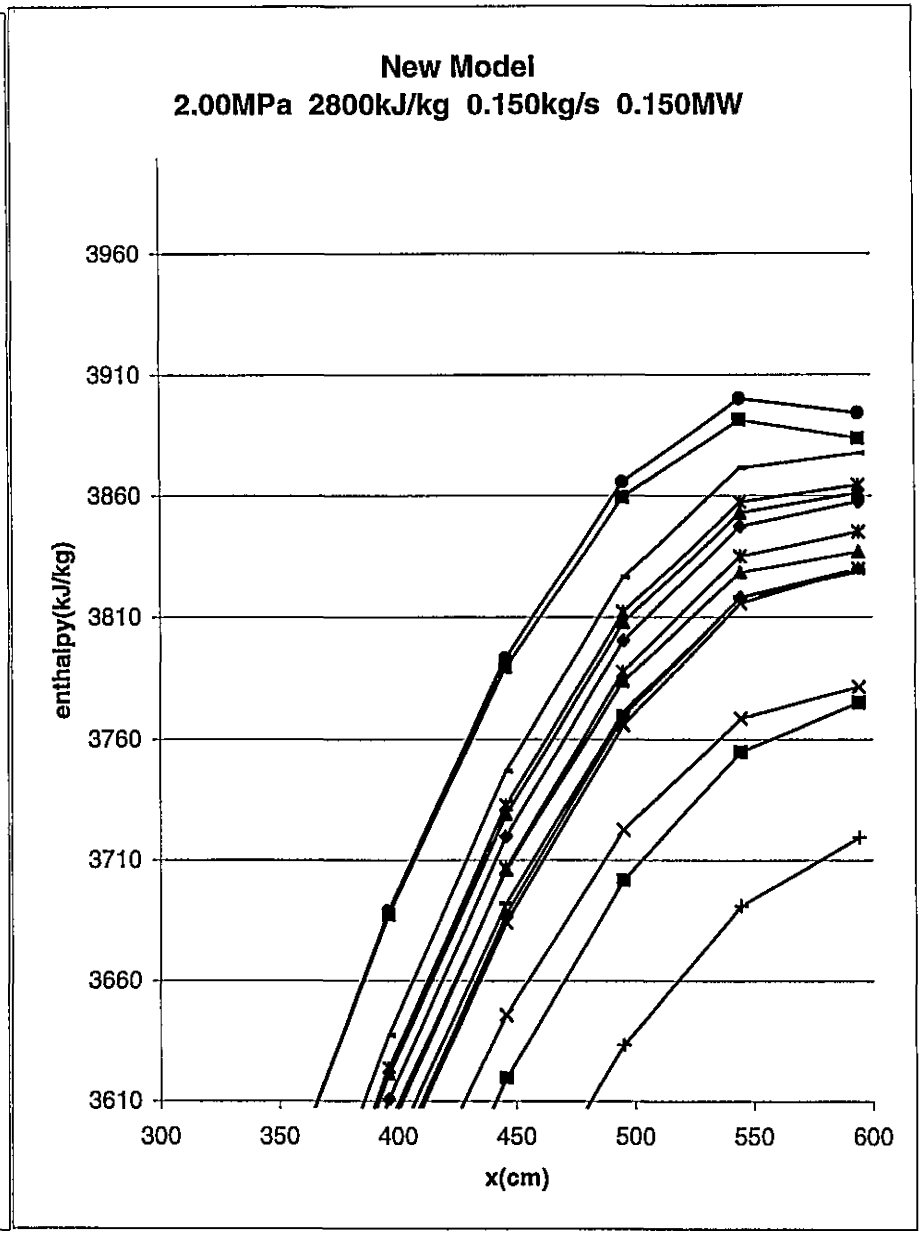

ASSERT
2.00MPa 2800kJ/kg 0.65kg/s 0.08125MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

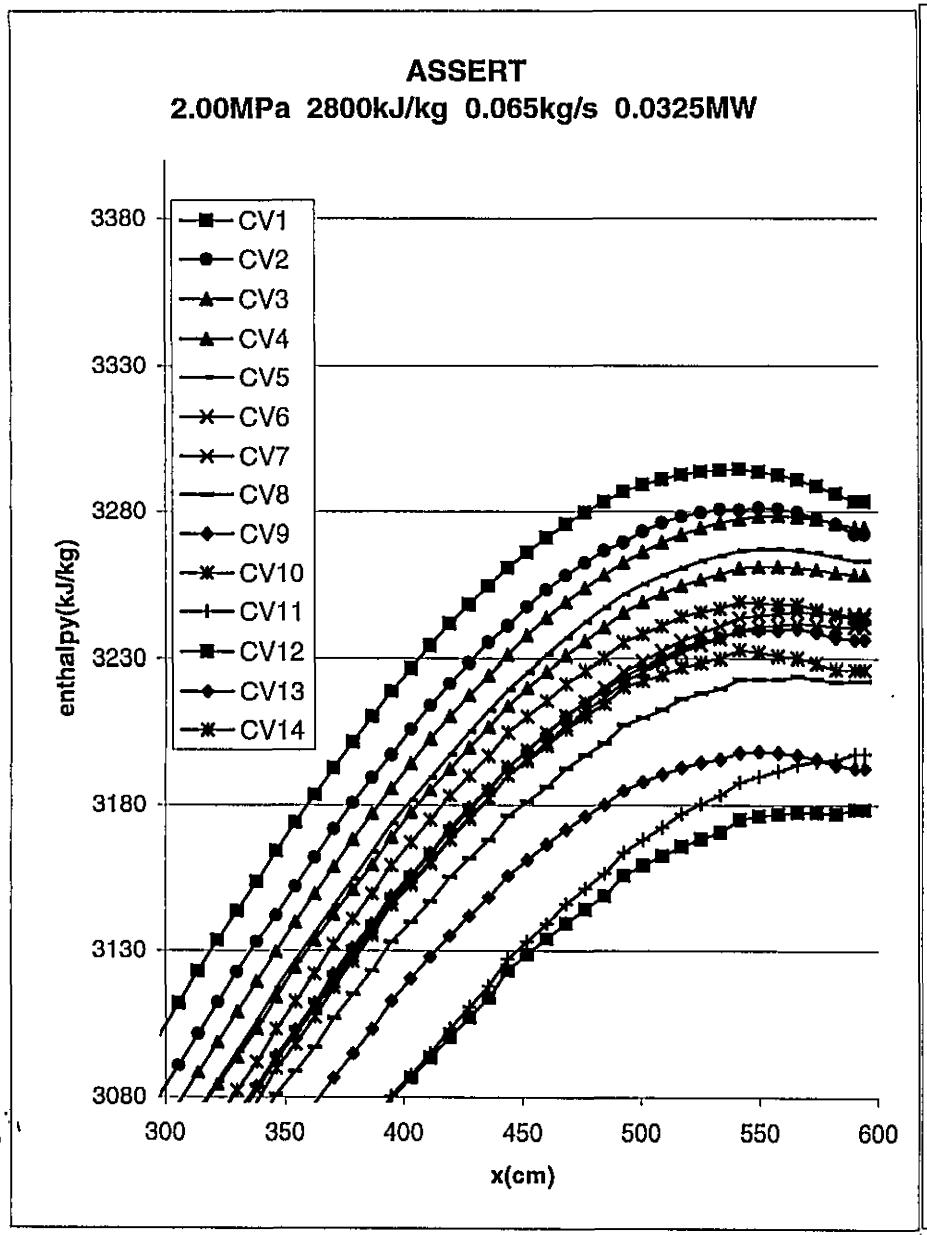

New Model
2.00MPa 2800kJ/kg 0.65kg/s 0.08125MW
enthalpy(kJ/kg)
x(cm)
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

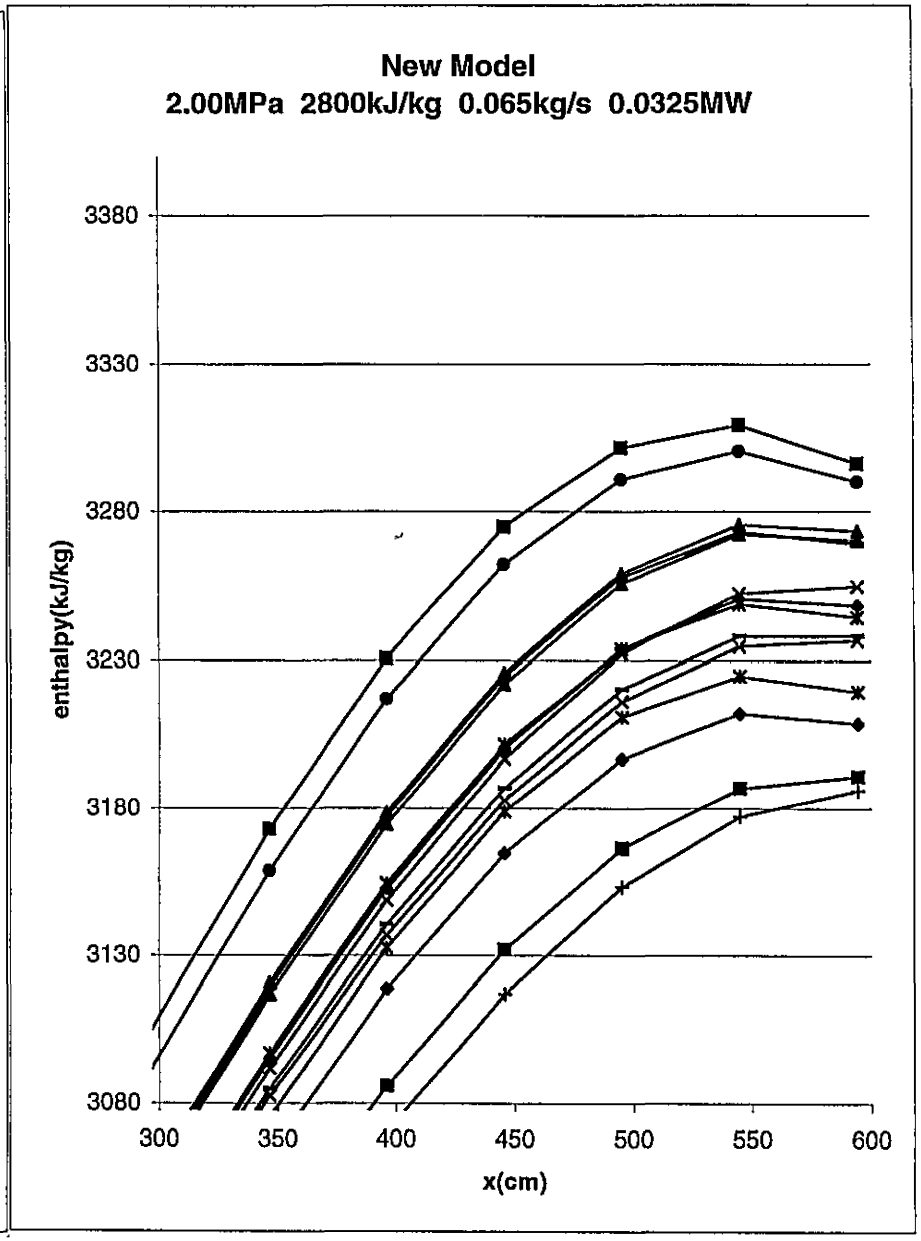

ASSERT
2.00MPa 2800kJ/kg 0.25kg/s 0.5MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
5090
4990
4890
4790
4690
4590
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

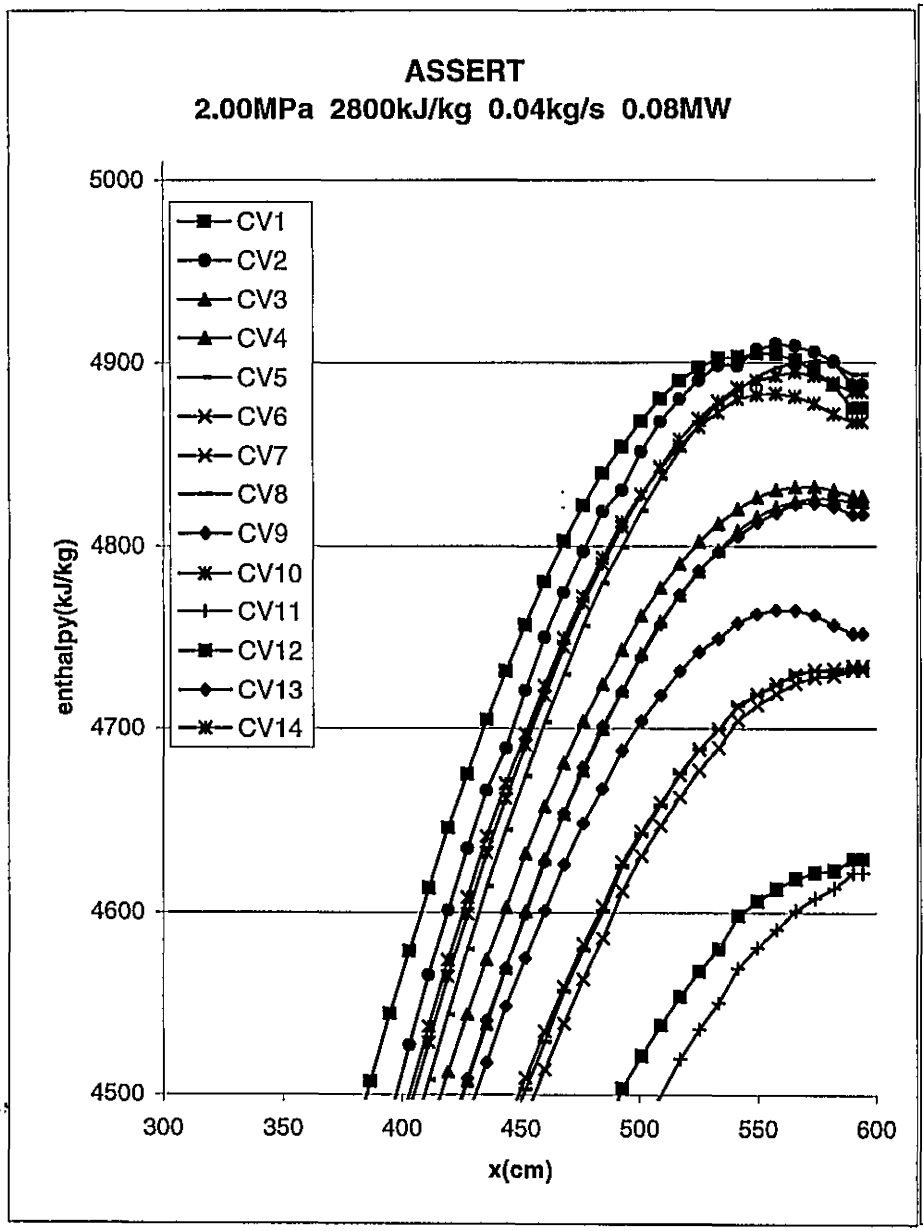

New Model
2.00MPa 2800kJ/kg 0.25kg/s 0.5MW
5090
4990
4890
4790
4690
4590
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

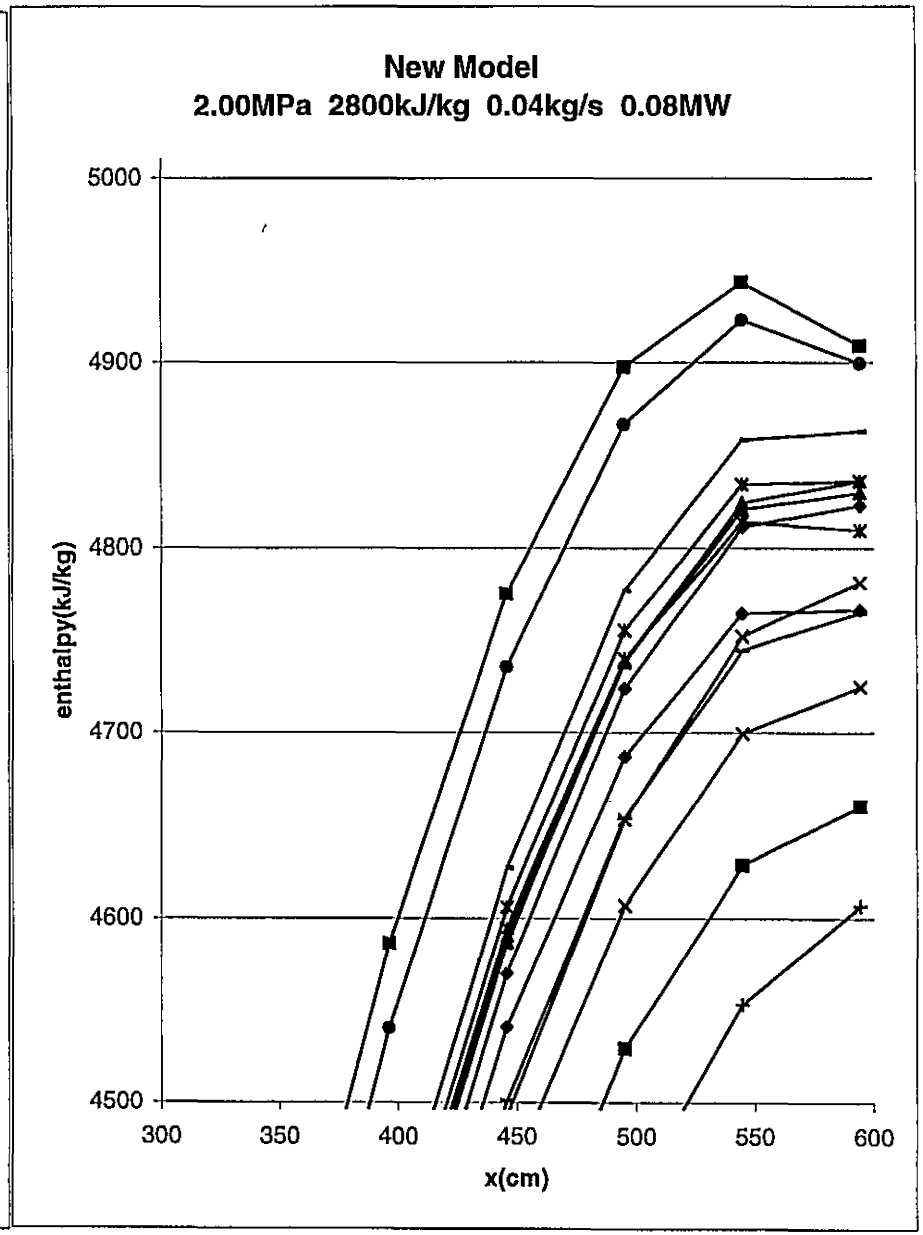

ASSERT
2.00MPa 2800kJ/kg 0.150kg/s 0.150MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3960
3910
3860
3810
3760
3710
3660
3610
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

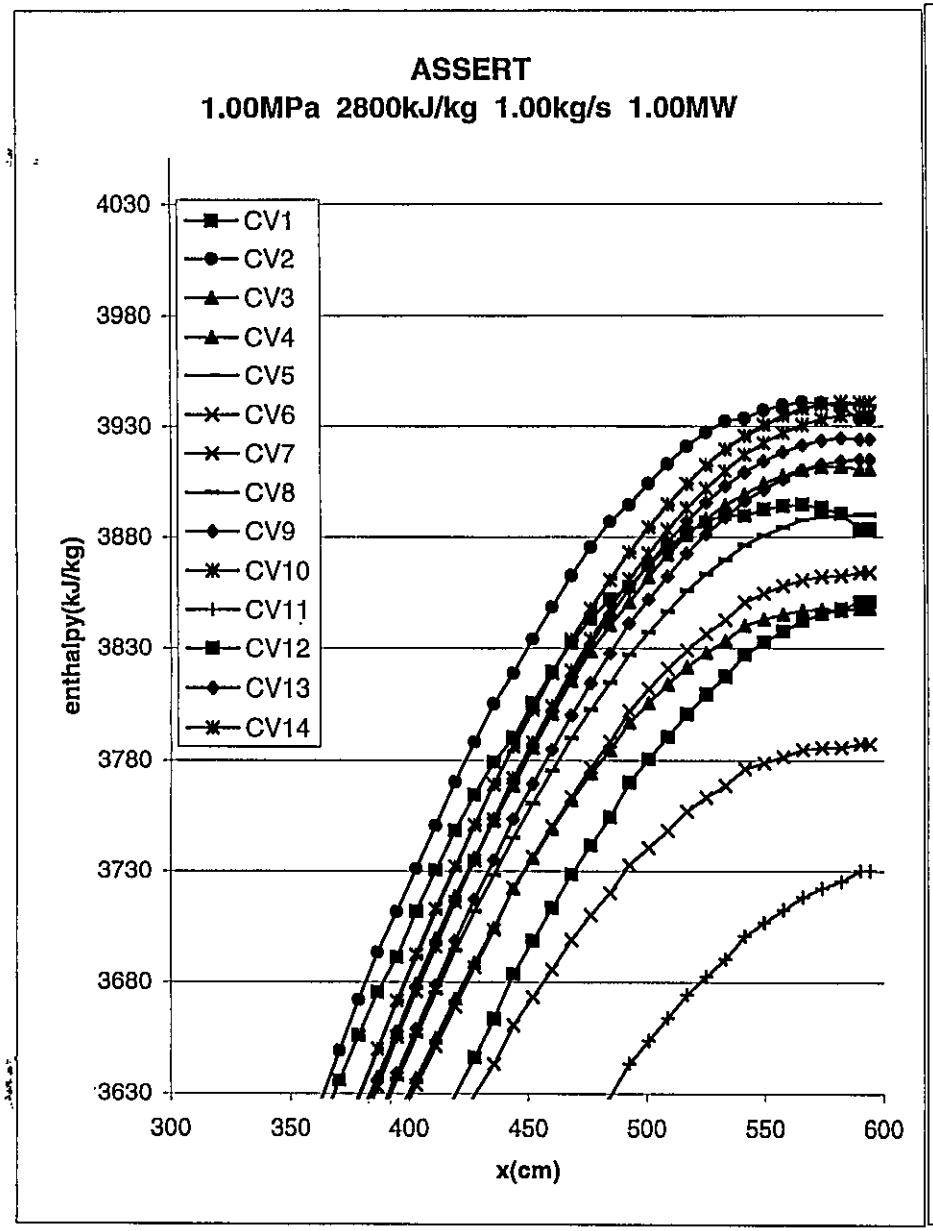

New Model
2.00MPa 2800kJ/kg 0.150kg/s 0.150MW
3960
3910
3860
3810
3760
3710
3660
3610
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

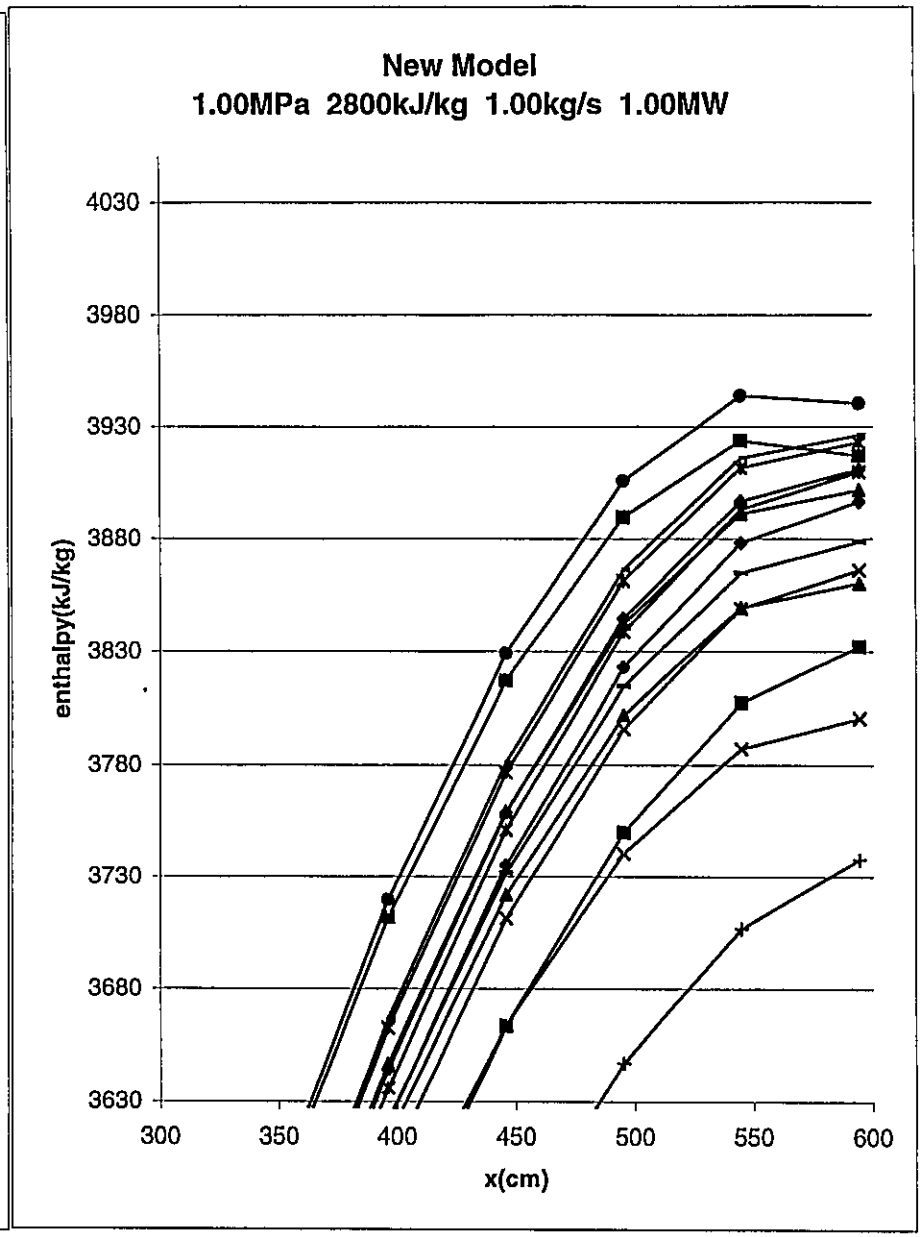

ASSERT
2.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3380
3330
3280
3230
3180
3130
3080
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

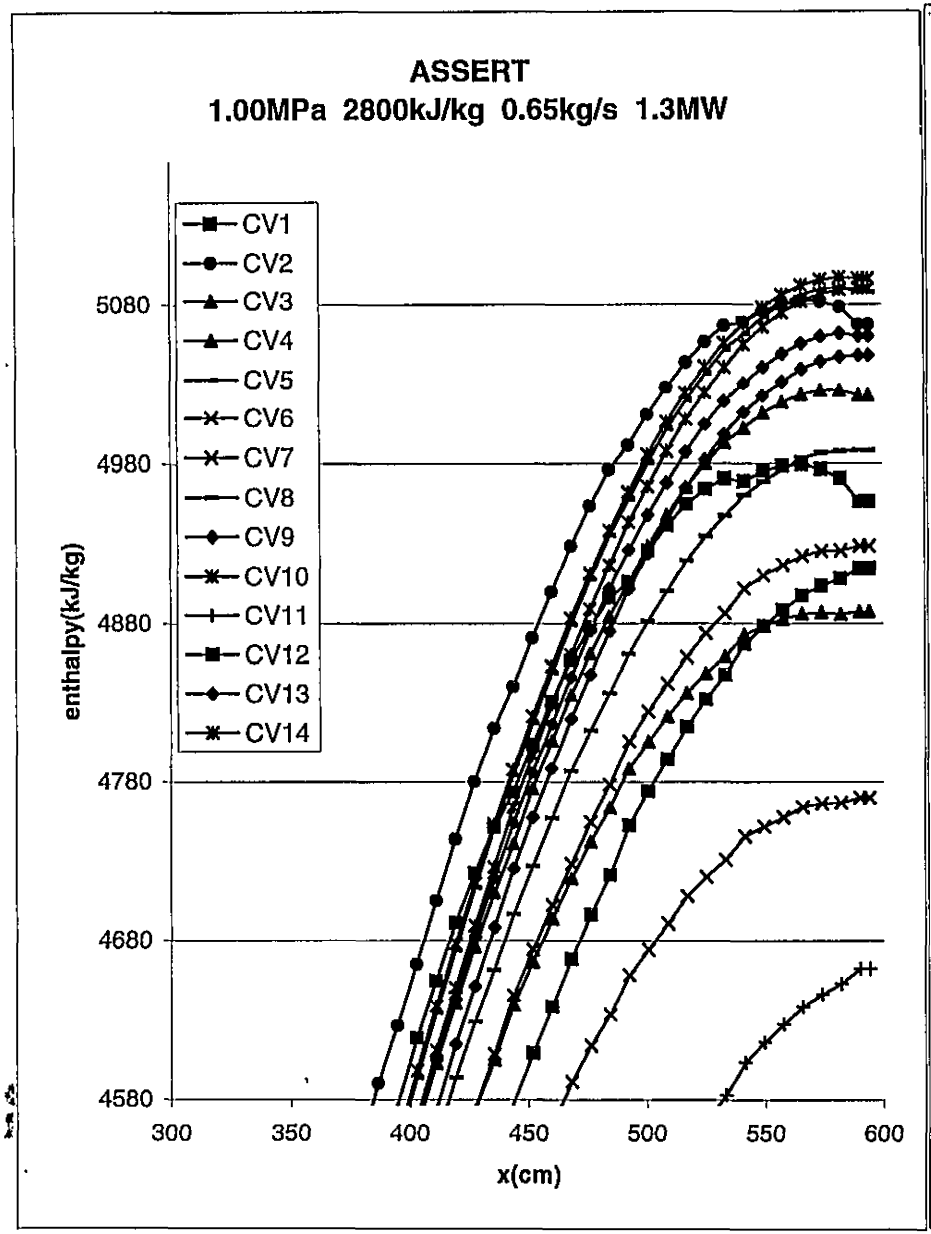

New Model
2.00MPa 2800kJ/kg 0.065kg/s 0.0325MW
3380
3330
3280
3230
3180
3130
3080
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

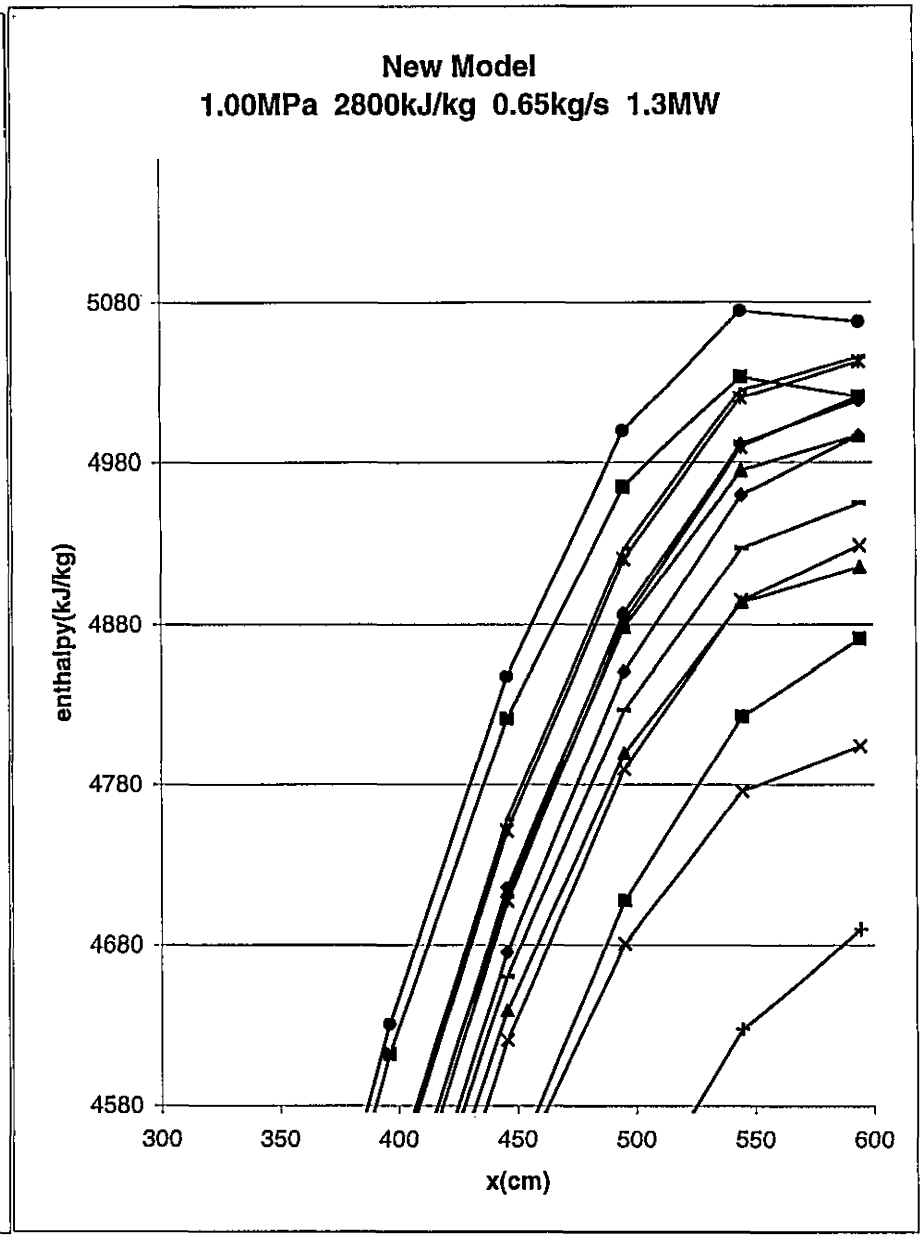

ASSERT
2.00MPa 2800kJ/kg 0.04kg/s 0.08MW
5000
4900
4800
4700
4600
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

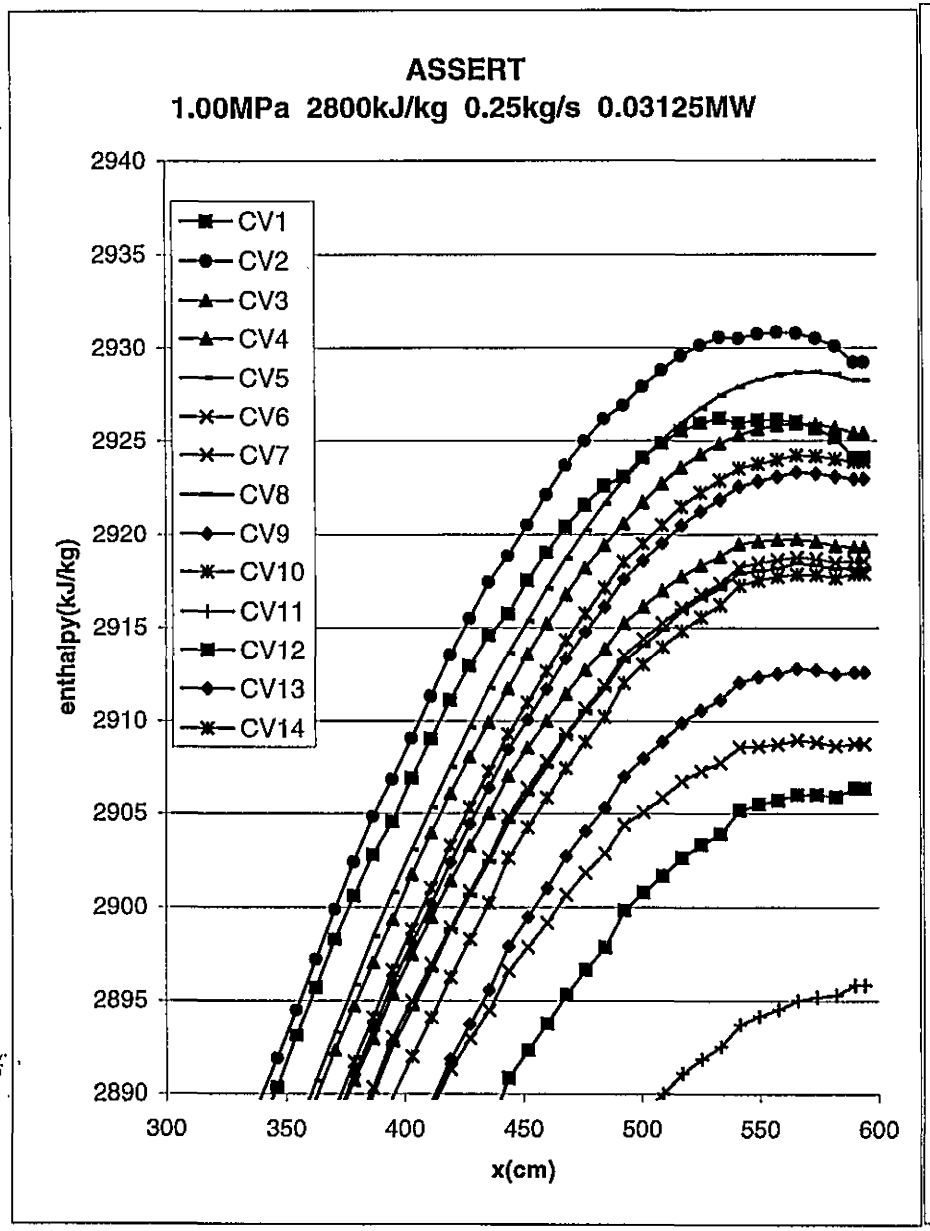

New Model
2.00MPa 2800kJ/kg 0.04kg/s 0.08MW
5000
4900
4800
4700
4600
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

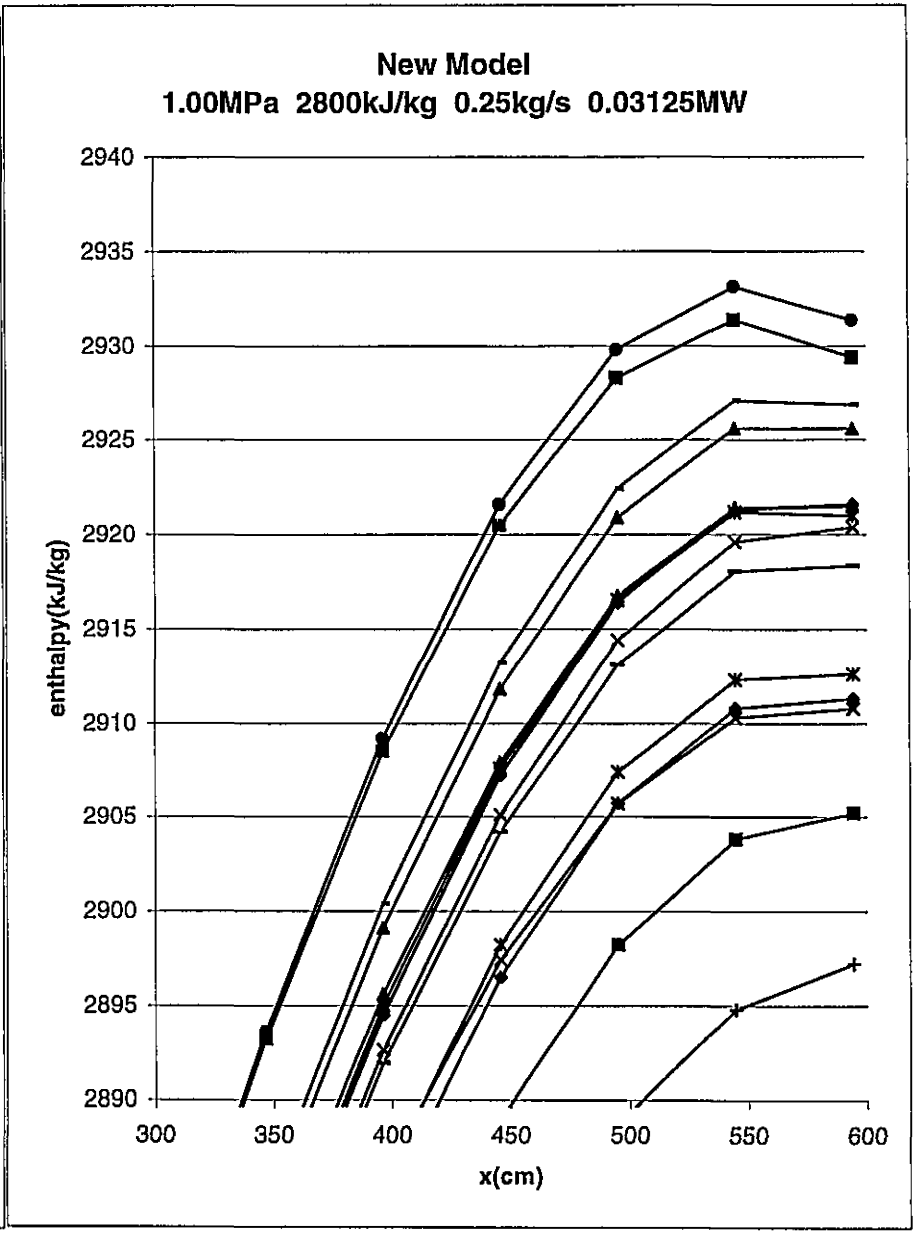

ASSERT
1.00MPa 2800kJ/kg 1.00kg/s 1.00MW
enthalpy(kJ/kg)
4030
3980
3930
3880
3830
3780
3730
3680
3630
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

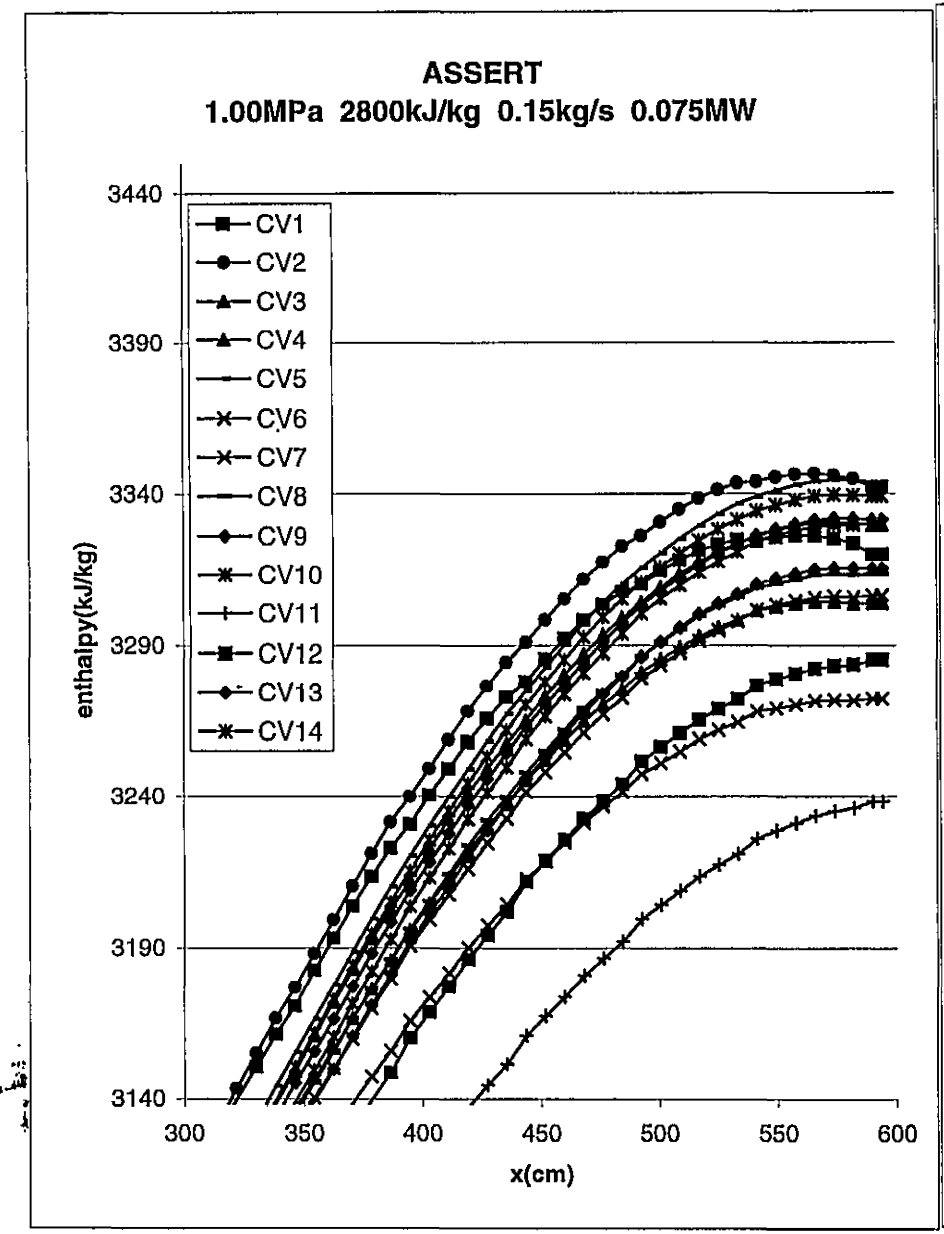

New Model
1.00MPa 2800kJ/kg 1.00kg/s 1.00MW
enthalpy(kJ/kg)
4030
3980
3930
3880
3830
3780
3730
3680
3630
300
350
400
450
500
550
600
x(cm)

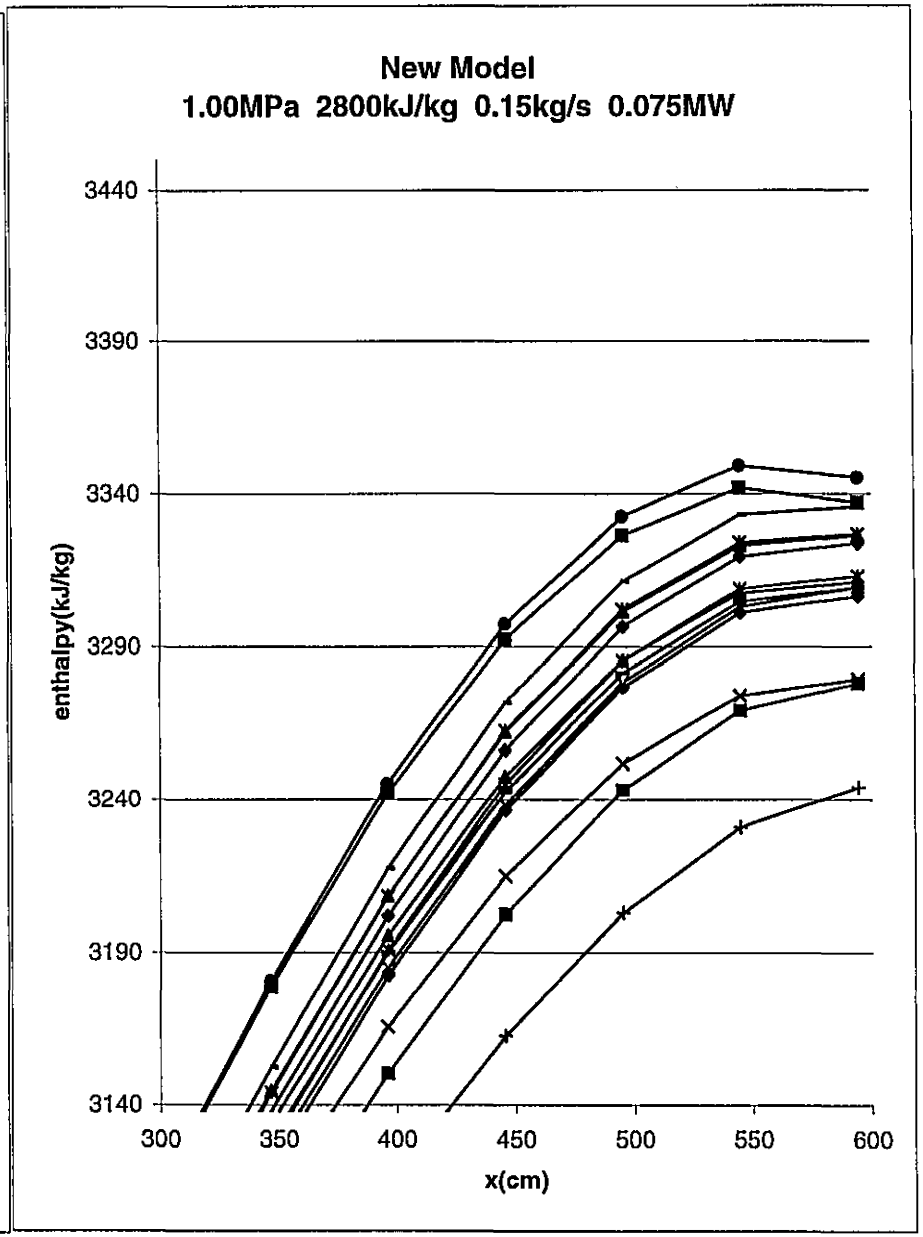

ASSERT
1.00MPa 2800kJ/kg 0.65kg/s 1.3MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

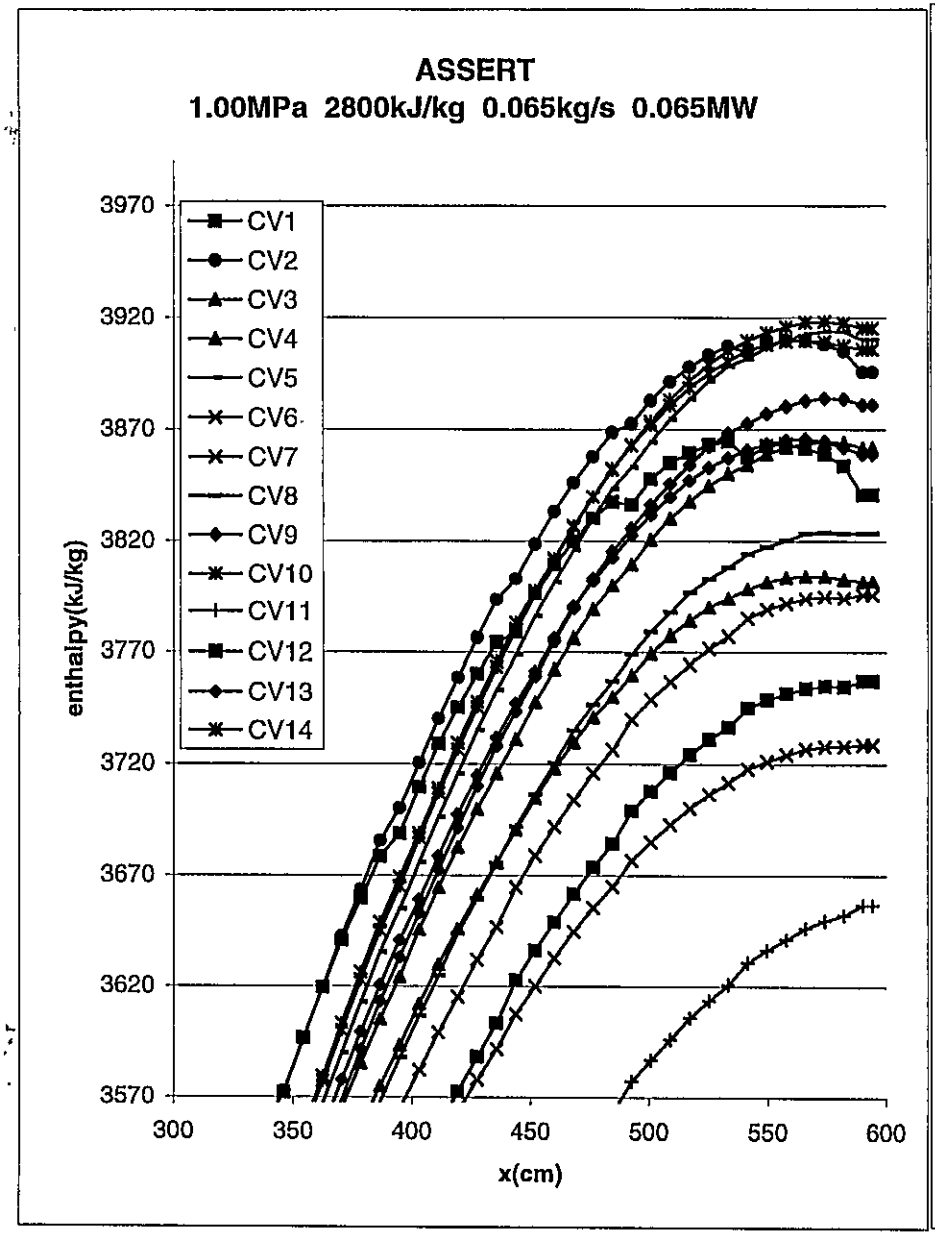

New Model
1.00MPa 2800kJ/kg 0.65kg/s 1.3MW
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

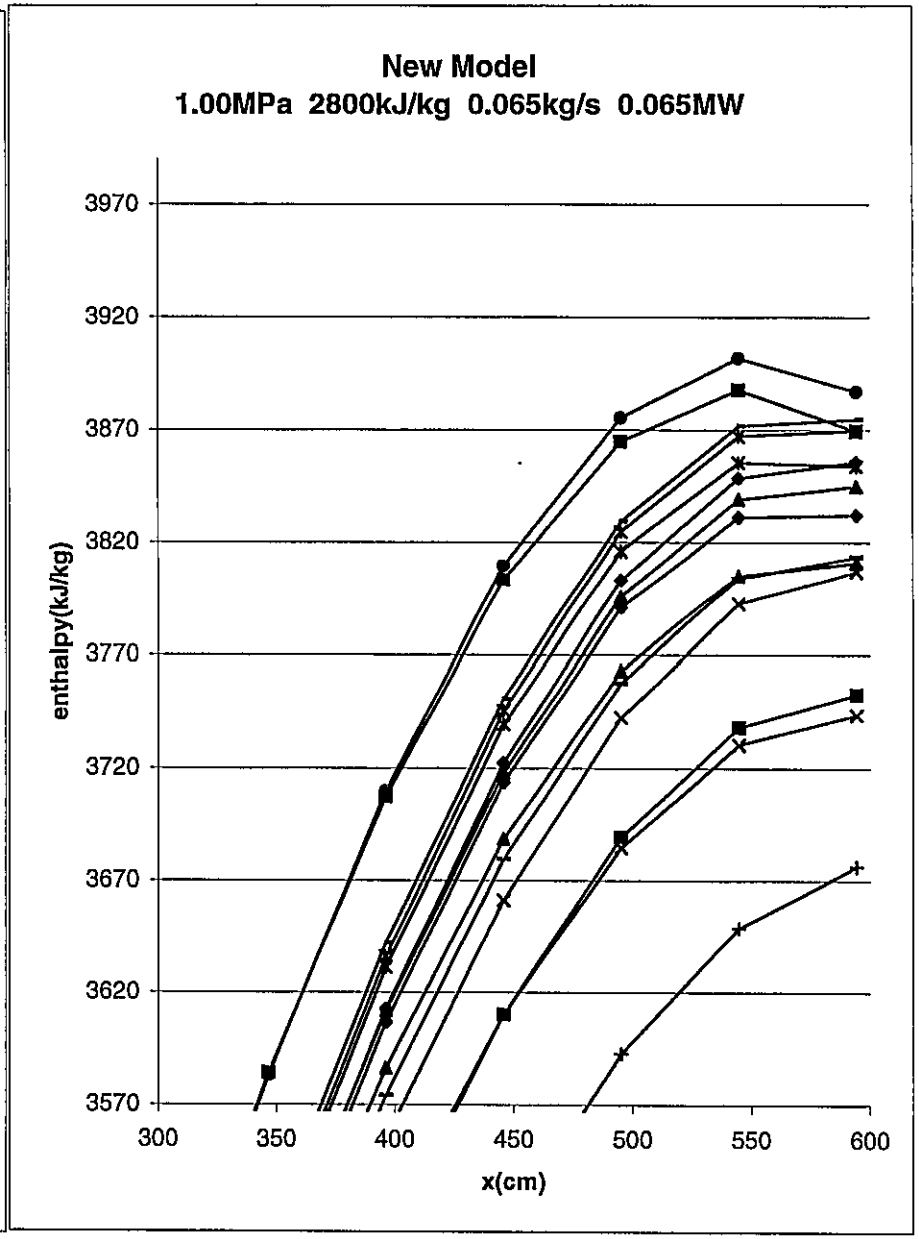

ASSERT
1.00MPa 2800kJ/kg 0.25kg/s 0.03125MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
2940
2935
2930
2925
2920
2915
2910
2905
2900
2895
2890
300
350
400
450
500
550
600

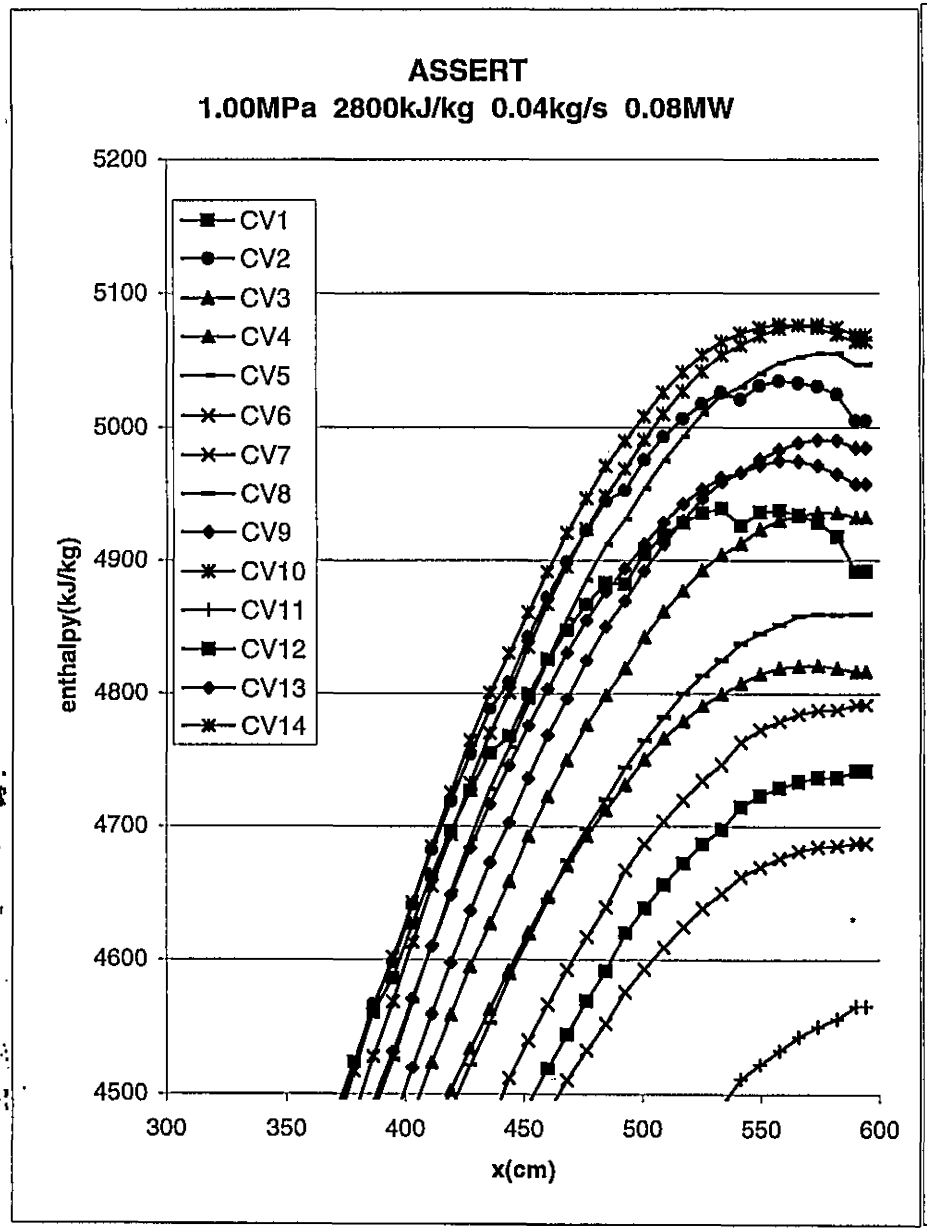

New Model
1.00MPa 2800kJ/kg 0.25kg/s 0.03125MW
enthalpy(kJ/kg)
x(cm)
2940
2935
2930
2925
2920
2915
2910
2905
2900
2895
2890
300
350
400
450
500
550
600

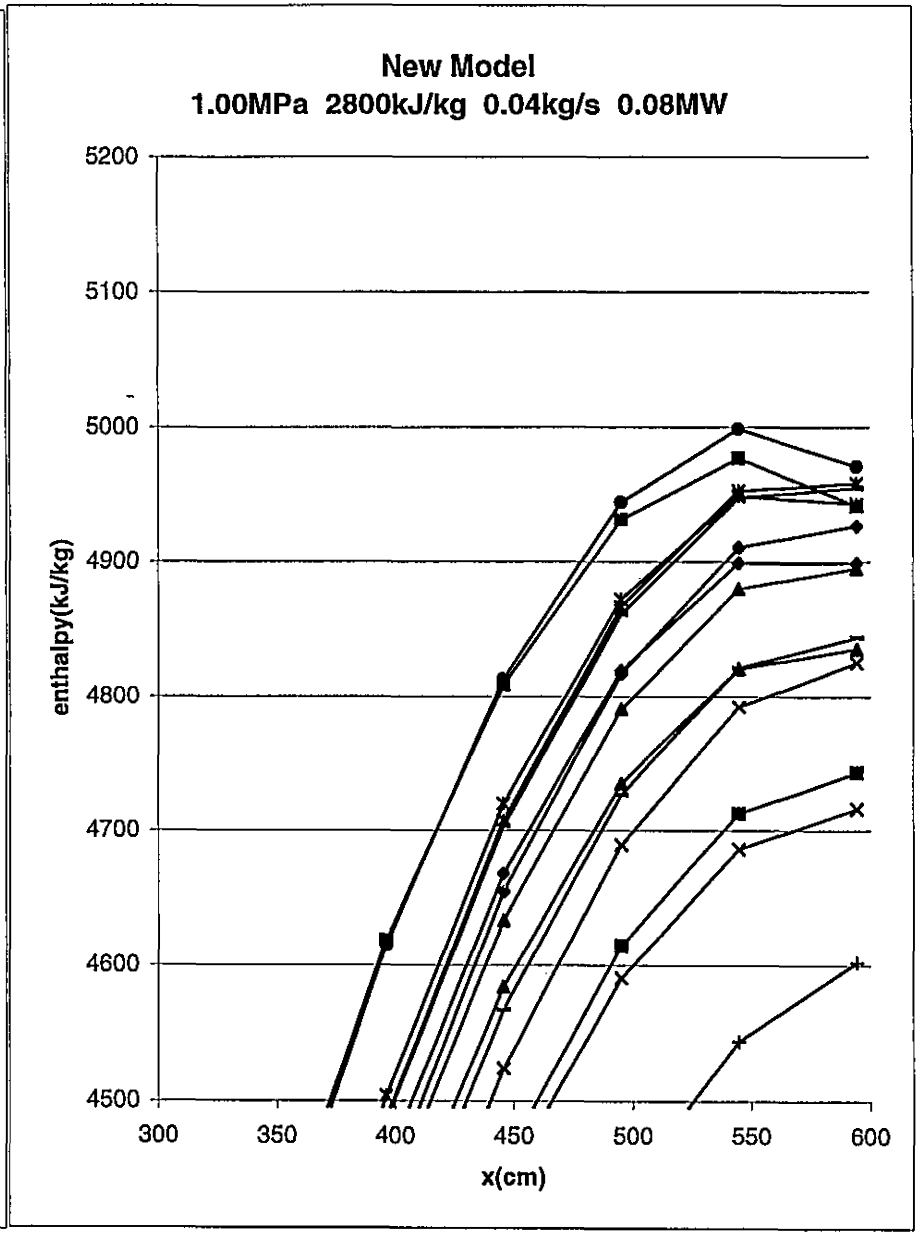

ASSERT
1.00MPa 2800kJ/kg 0.15kg/s 0.075MW
enthalpy(kJ/kg)
3440
3390
3340
3290
3240
3190
3140
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

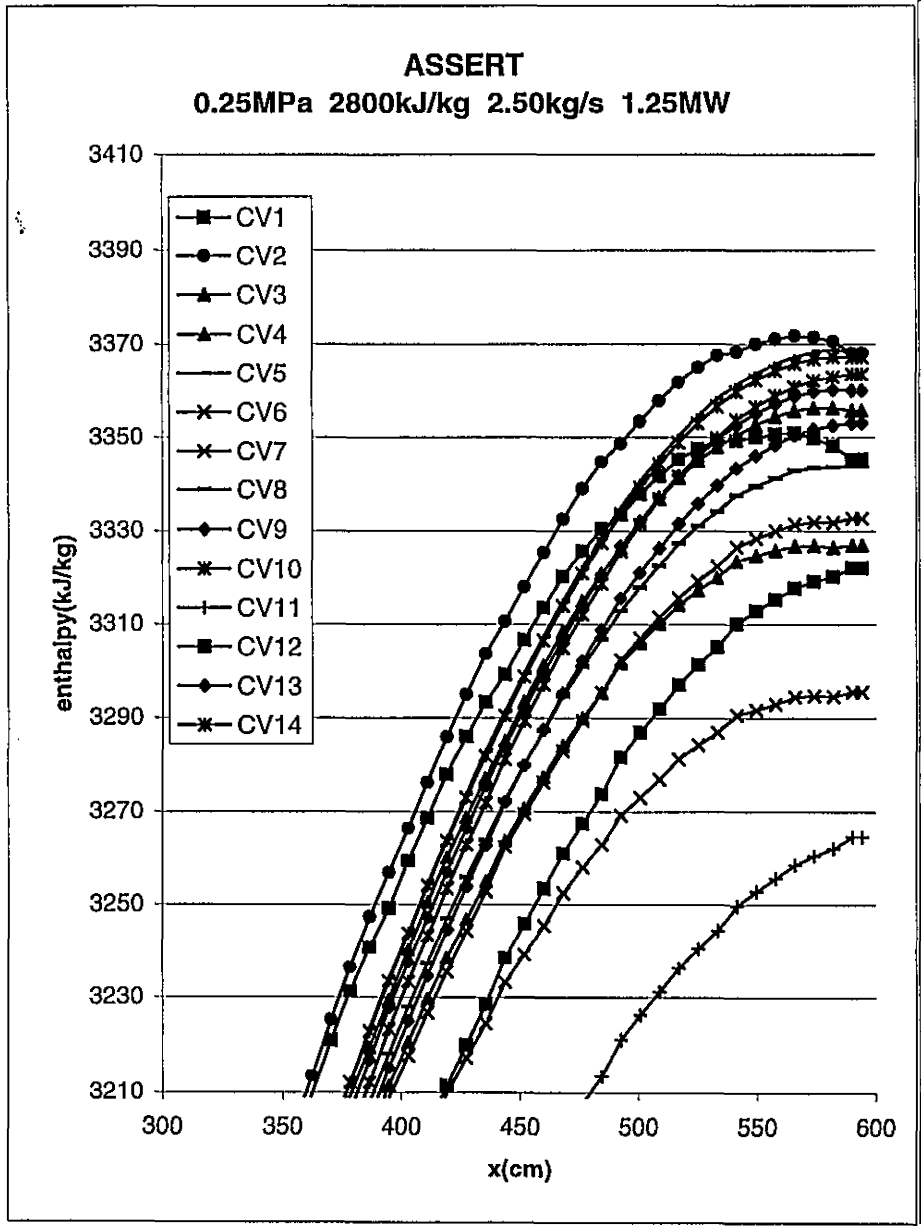

New Model
1.00MPa 2800kJ/kg 0.15kg/s 0.075MW
enthalpy(kJ/kg)
3440
3390
3340
3290
3240
3190
3140
300
350
400
450
500
550
600
x(cm)

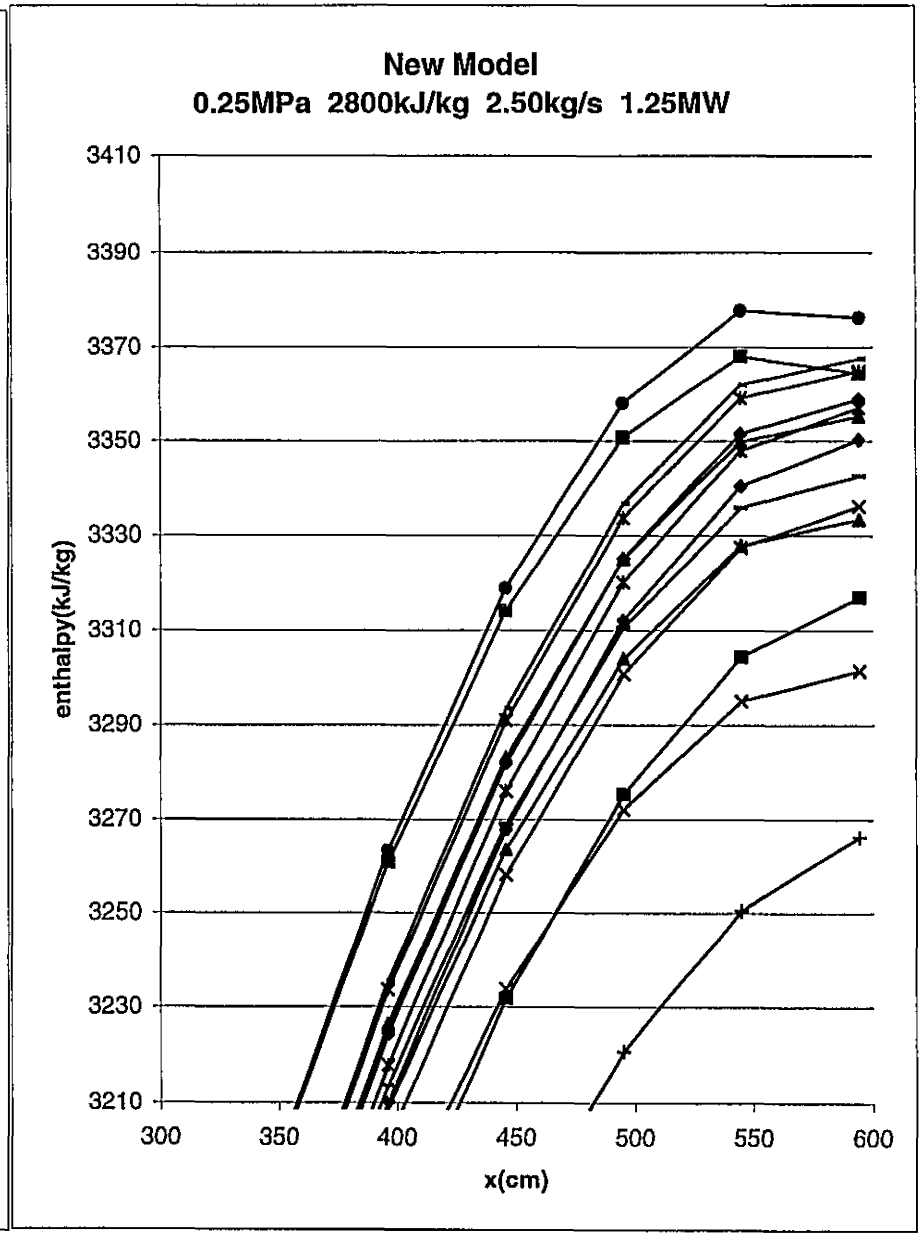

ASSERT
1.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
3970
3920
3870
3820
3770
3720
3670
3620
3570
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

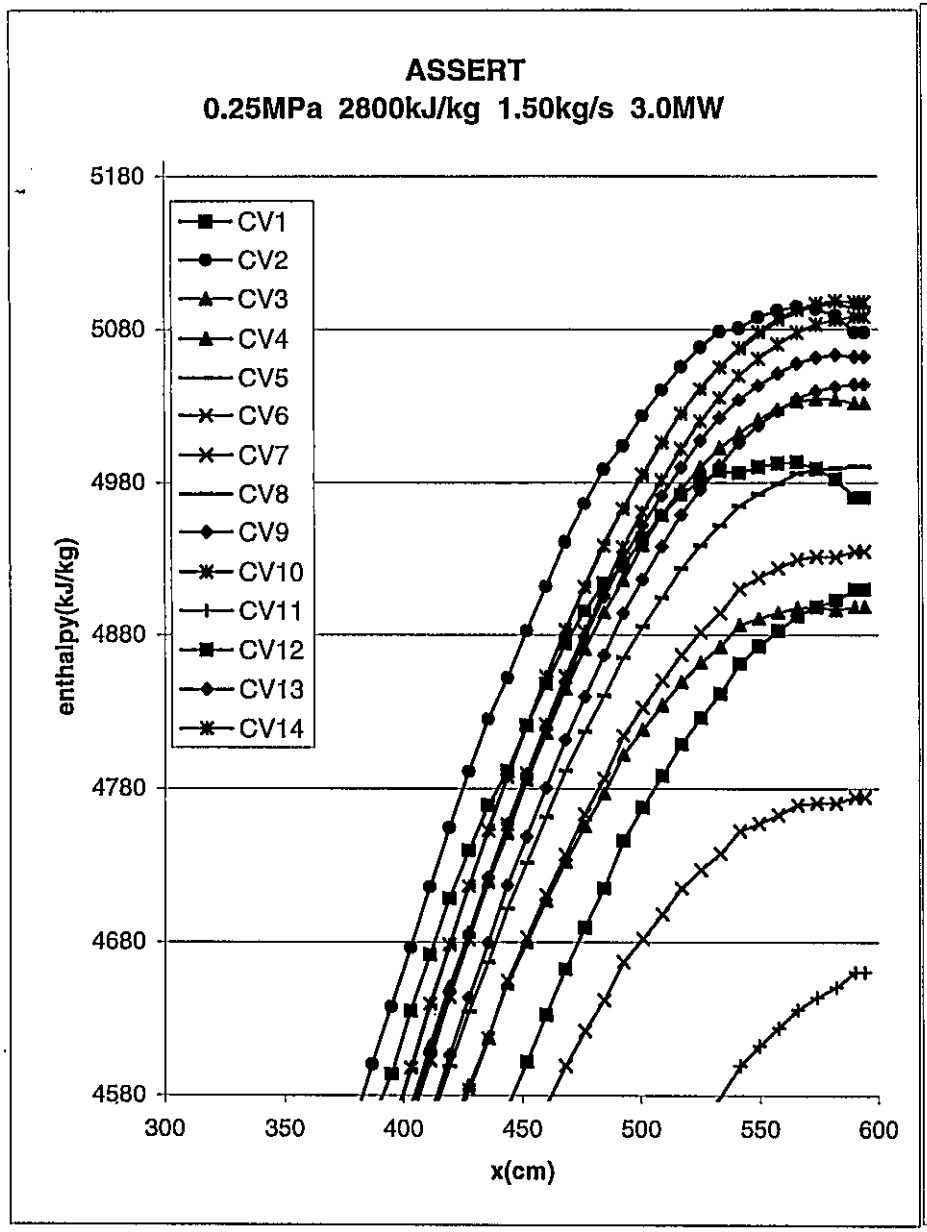

New Model
1.00MPa 2800kJ/kg 0.065kg/s 0.065MW
enthalpy(kJ/kg)
3970
3920
3870
3820
3770
3720
3670
3620
3570
300
350
400
450
500
550
600
x(cm)

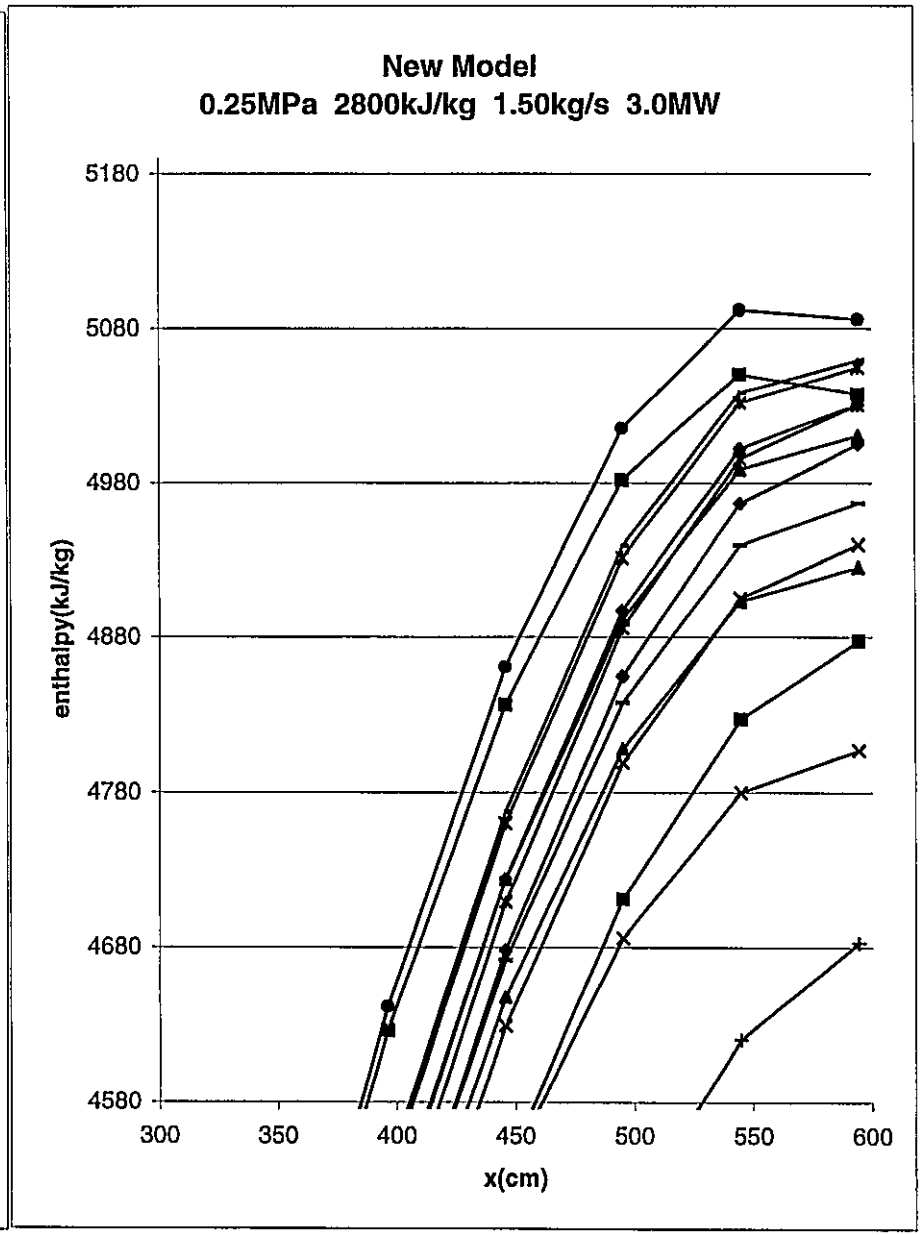

ASSERT
1.00MPa 2800kJ/kg 0.04kg/s 0.08MW
5200
5100
5000
4900
4800
4700
4600
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

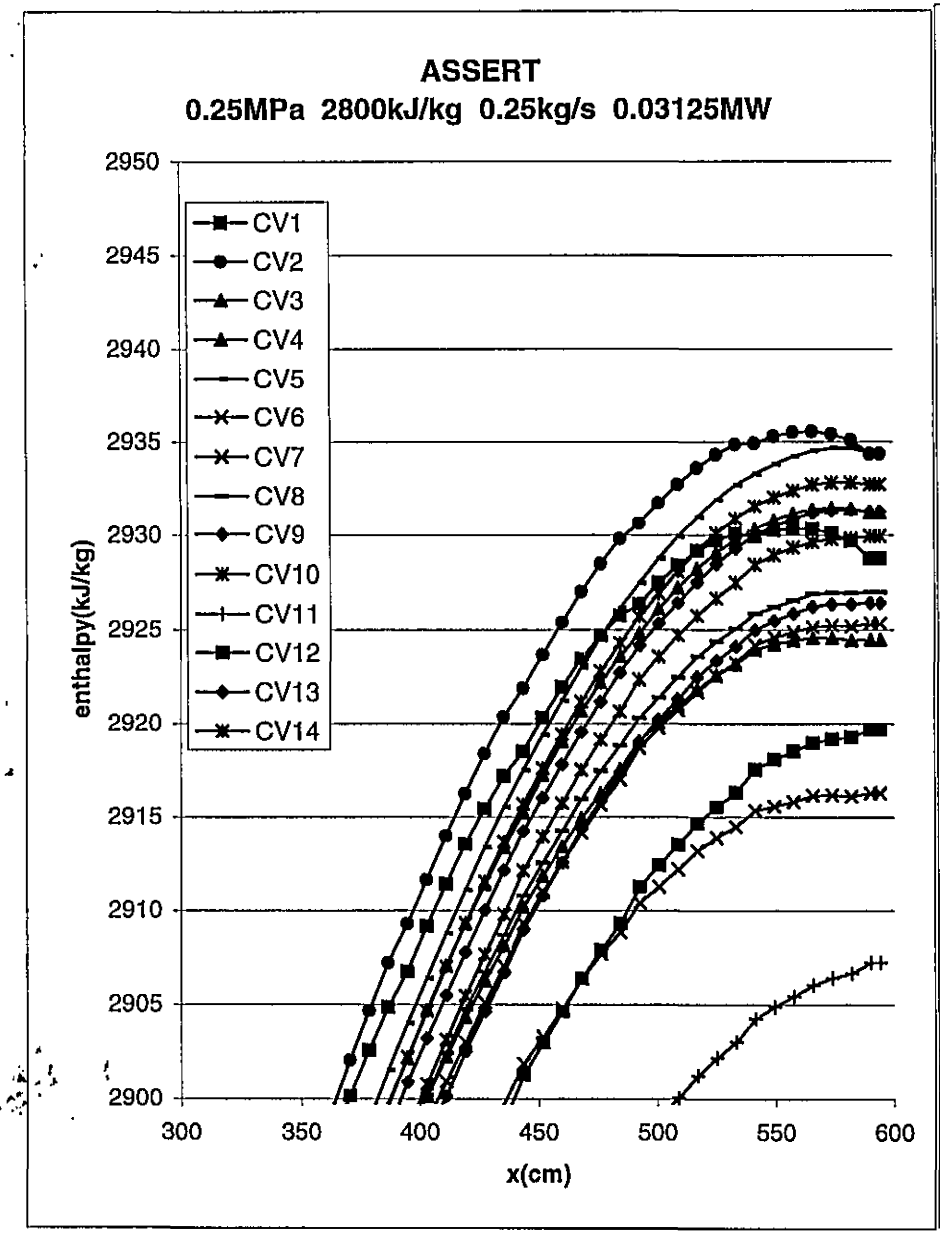

New Model
1.00MPa 2800kJ/kg 0.04kg/s 0.08MW
5200
5100
5000
4900
4800
4700
4600
4500
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

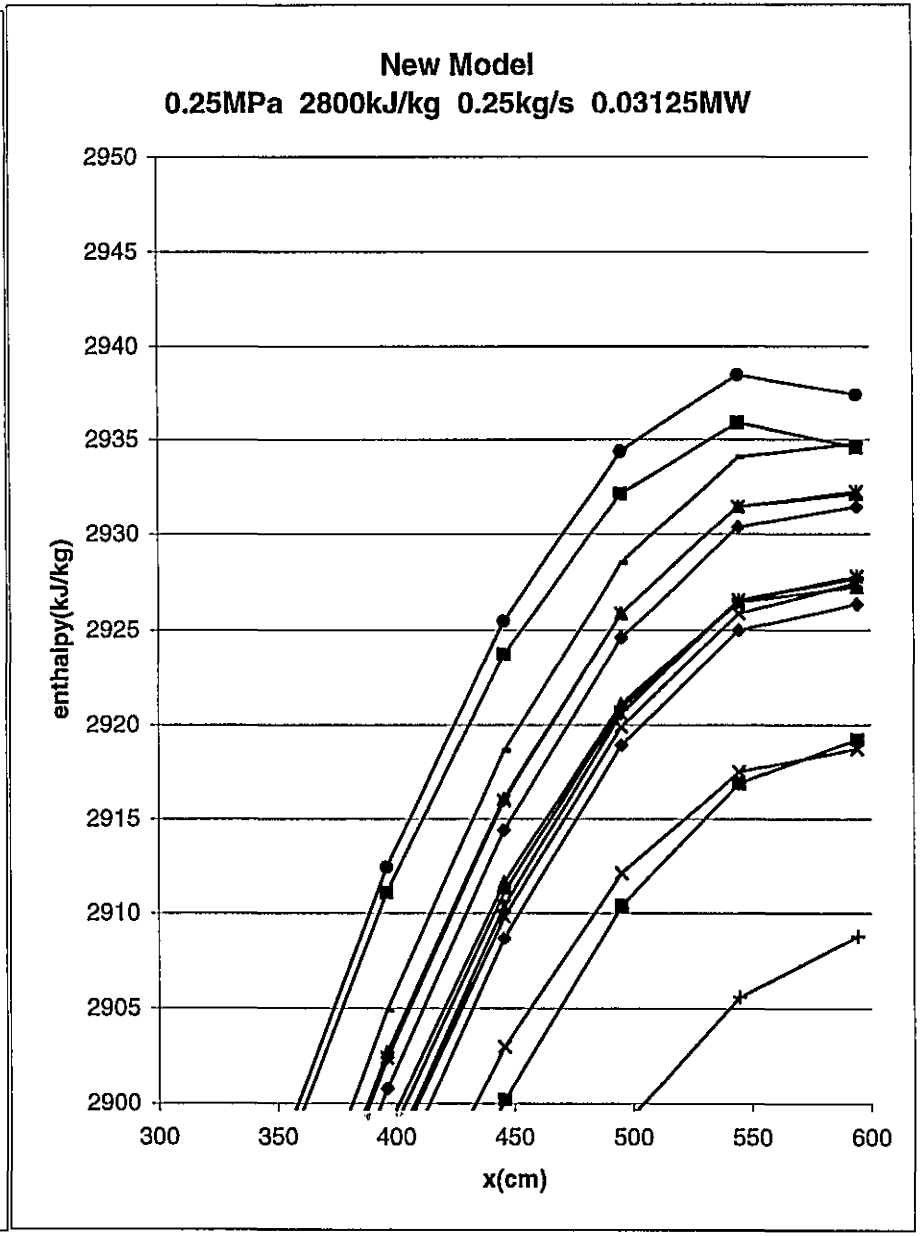

ASSERT
0.25MPa 2800kJ/kg 2.50kg/s 1.25MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3410
3390
3370
3350
3330
3310
3290
3270
3250
3230
3210
300
350
400
450
500
550
600

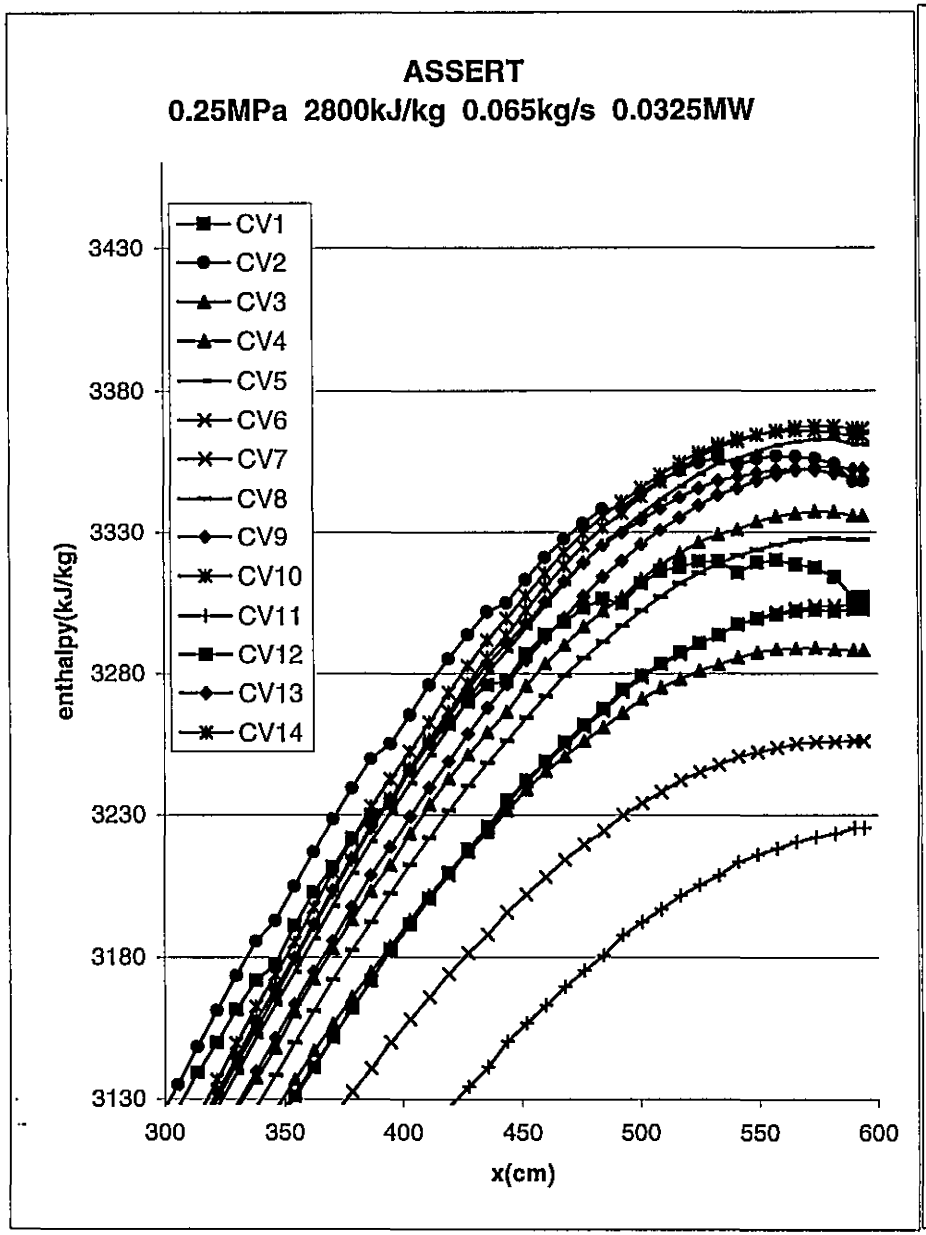

New Model
0.25MPa 2800kJ/kg 2.50kg/s 1.25MW
enthalpy(kJ/kg)
x(cm)
3410
3390
3370
3350
3330
3310
3290
3270
3250
3230
3210
300
350
400
450
500
550
600

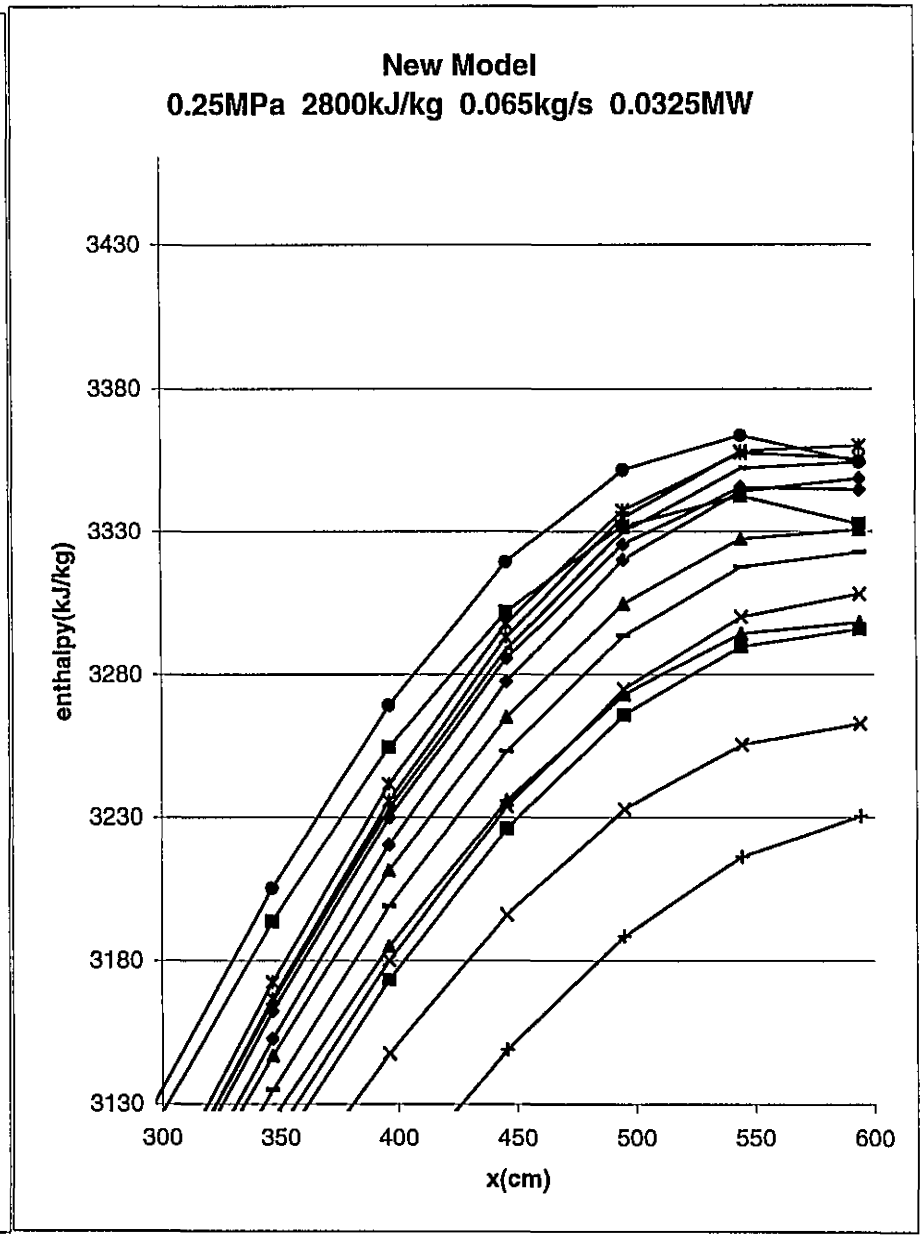

ASSERT
0.25MPa 2800kJ/kg 1.50kg/s 3.0MW
5180
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14

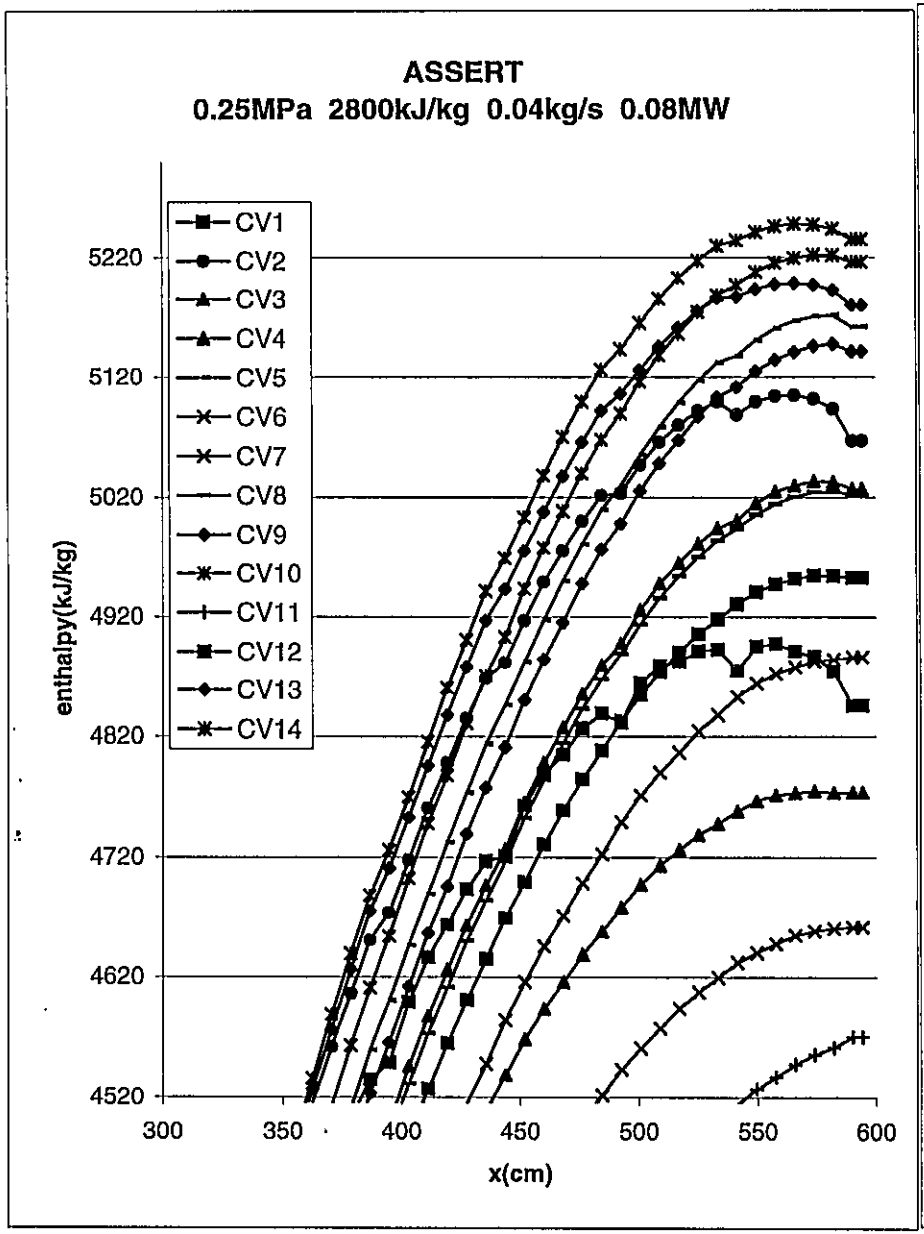

New Model
0.25MPa 2800kJ/kg 1.50kg/s 3.0MW
5180
5080
4980
4880
4780
4680
4580
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)

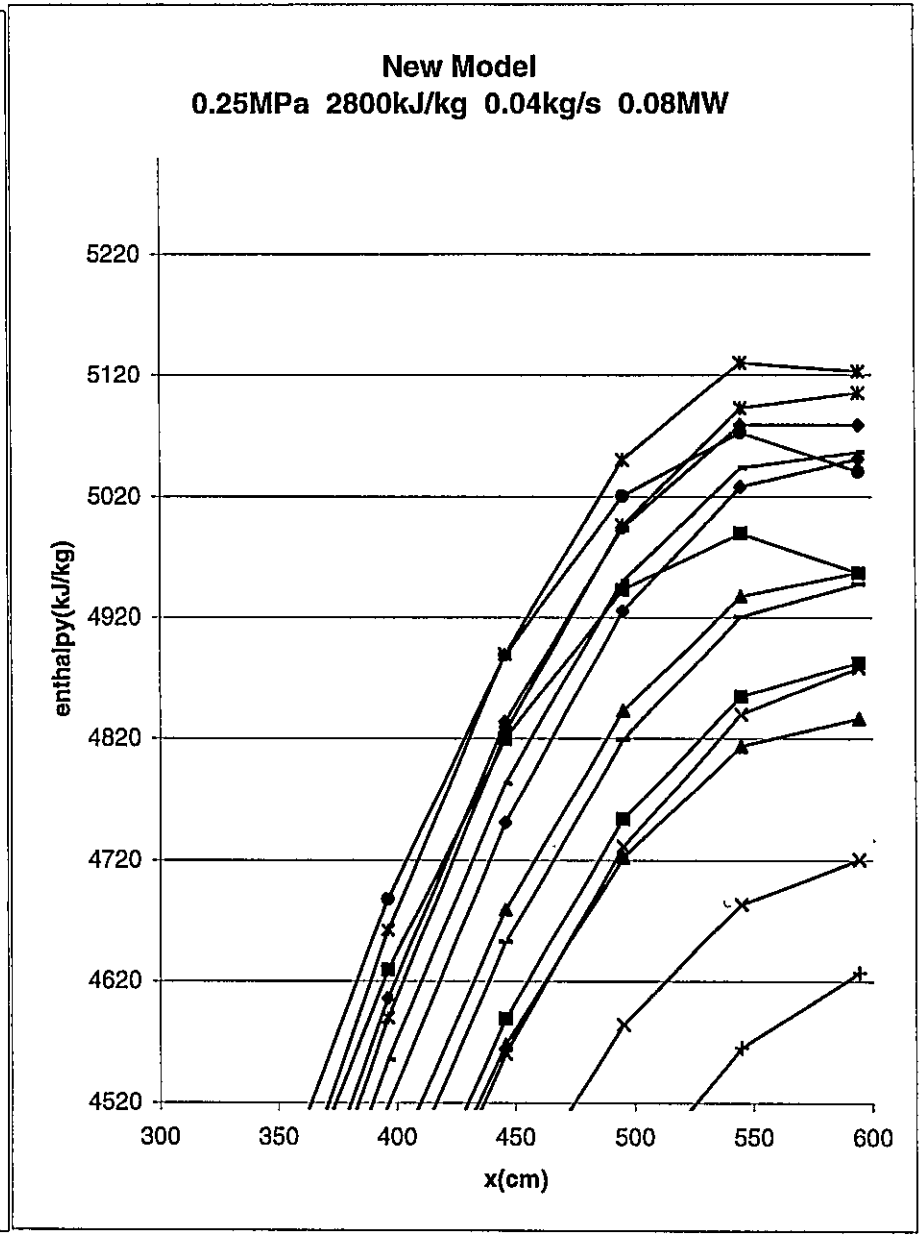

ASSERT
0.25MPa 2800kJ/kg 0.25kg/s 0.03125MW
enthalpy(kJ/kg)
x(cm)
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

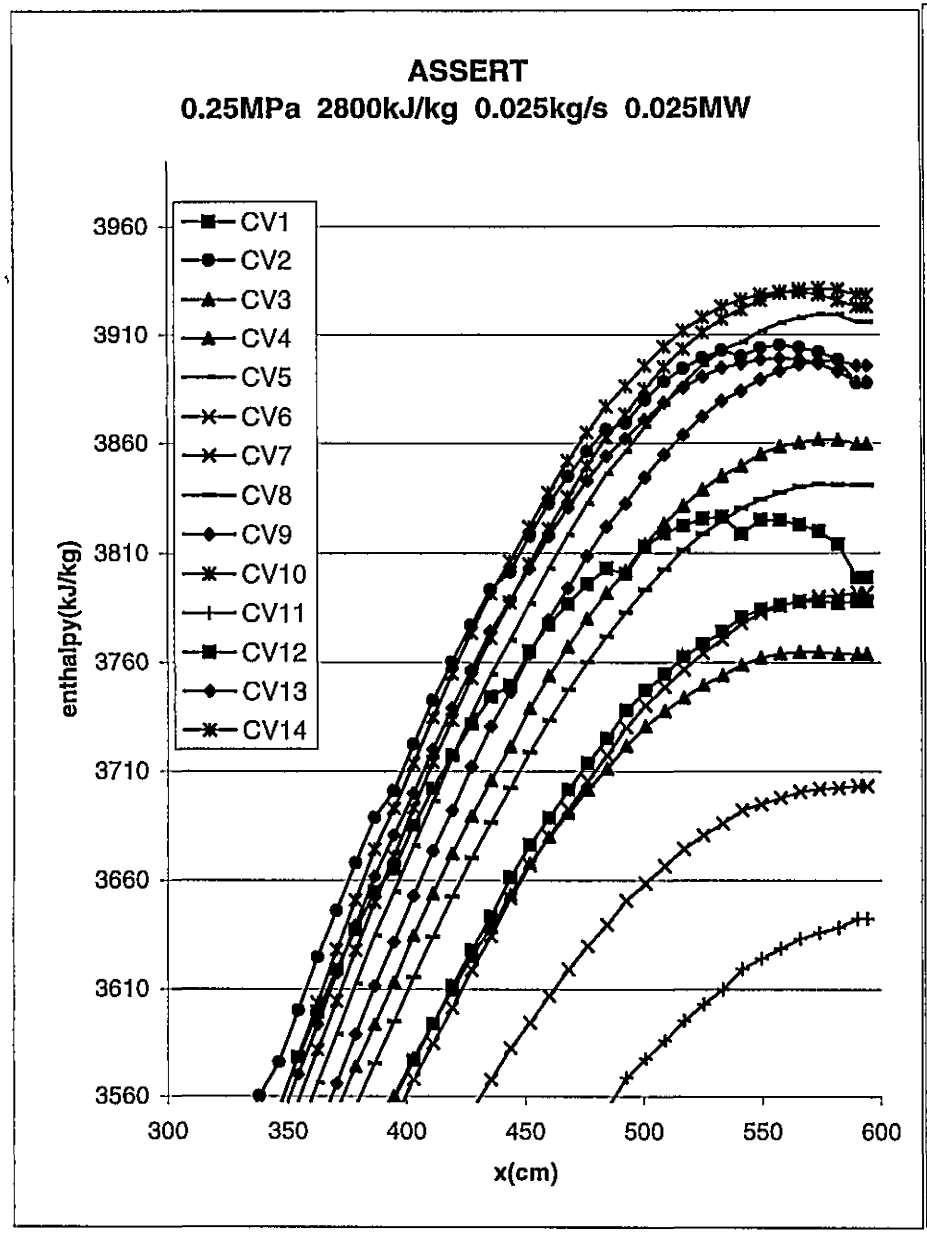

New Model
0.25MPa 2800kJ/kg 0.25kg/s 0.03125MW
enthalpy(kJ/kg)
x(cm)
2950
2945
2940
2935
2930
2925
2920
2915
2910
2905
2900
300
350
400
450
500
550
600

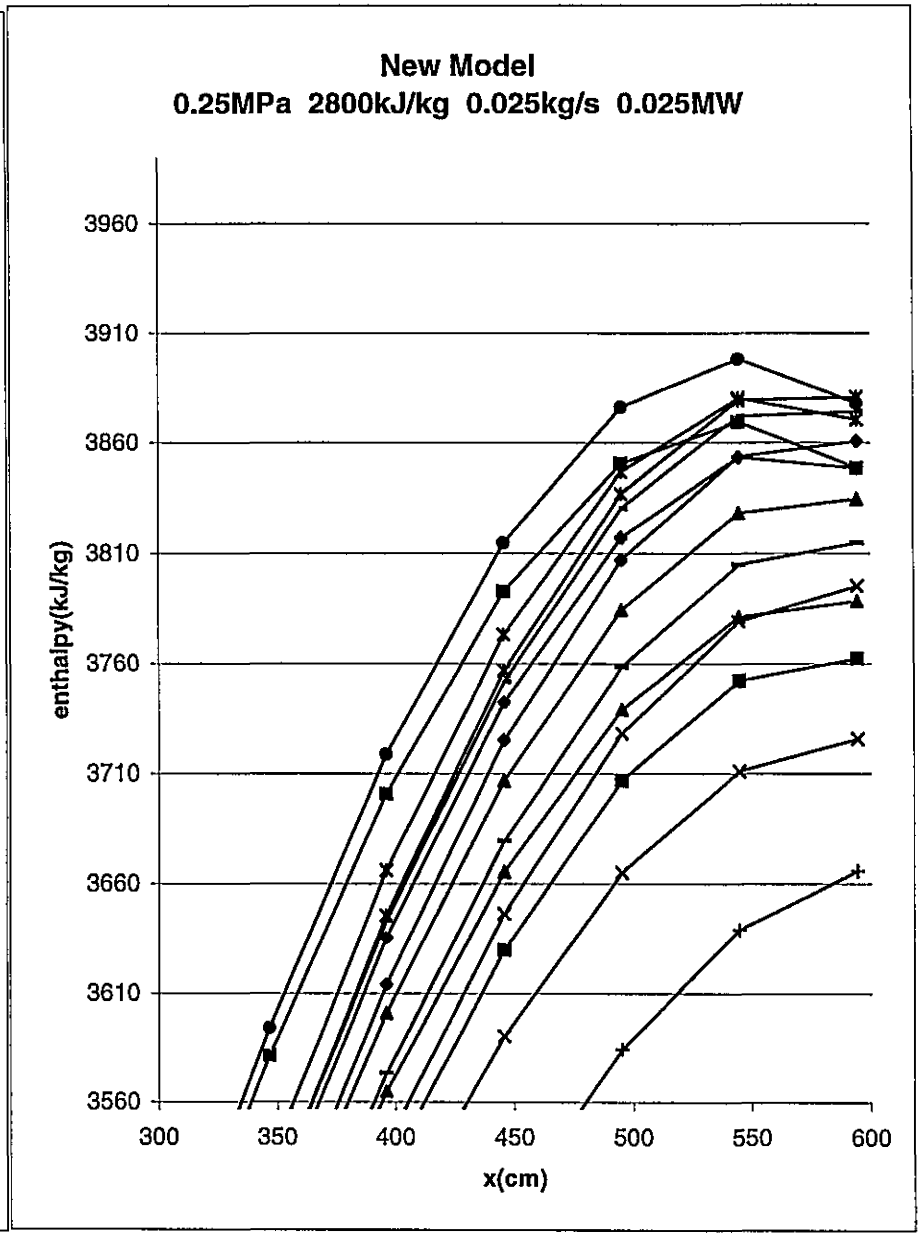

ASSERT
0.25MPa 2800kJ/kg 0.065kg/s 0.0325MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
3430
3380
3330
3280
3230
3180
3130
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)


New Model
0.25MPa 2800kJ/kg 0.065kg/s 0.0325MW
3430
3380
3330
3280
3230
3180
3130
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)


ASSERT
0.25MPa 2800kJ/kg 0.04kg/s 0.08MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
5220
5120
5020
4920
4820
4720
4620
4520
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)


New Model
0.25MPa 2800kJ/kg 0.04kg/s 0.08MW
5220
5120
5020
4920
4820
4720
4620
4520
enthalpy(kJ/kg)
300
350
400
450
500
550
600
x(cm)


ASSERT
0.25MPa 2800kJ/kg 0.025kg/s 0.025MW
CV1
CV2
CV3
CV4
CV5
CV6
CV7
CV8
CV9
CV10
CV11
CV12
CV13
CV14
enthalpy(kJ/kg)
3960
3910
3860
3810
3760
3710
3660
3610
3560
300
350
400
450
500
550
600
x(cm)


New Model
0.25MPa 2800kJ/kg 0.025kg/s 0.025MW
enthalpy(kJ/kg)
3960
3910
3860
3810
3760
3710
3660
3610
3560
300
350
400
450
500
550
600
x(cm)

Appendix A

Sample ASSERT PV V3.0

Input Card

FILE: 37_ballooned_input.inp Jaff Robertson 2002.05.24

llooned PT case

ecord i: Run Control (I5)

MAXT: maximum CPU time (seconds), not used
/
)9

ecord ii: Case Control (I5,2A2,I1,A)

ASE - problem case number:
= 0: stop (default)
> 0: begin new case

NTIN - input data units option:
= SI: use SI units for input (default)
= BR: use British units for input

NTOUT - output data units option:
= SI: use SI units for output (default)
= BR: use British units for output

CHOPT - input data summary print option:
= 0: print only new input data (default)
= 1: print all input data
= 2: print only operating conditions
> 2: print all input data, then stop

EXT - output text for case identification (maximum of 64 characters)

KASE: case ID number
/ UNTIN=SI: use SI units for input
/ / UNTOUT=SI: use SI units for output
/ / / ECHOPT=1: print all input data
/ / /_/ TEXT: case description
/ /// /
)0SISI1 ASSERT-PV V3R0 Case 15.00: 37-Rod CANDU, With Feeders (Constant DP)

roup 1: Fluid Properties

PROP - number of entries to be read in the property table (if NPACK=0)

PACK - source of the property data:
= 0: use property tables specified in the input data (Record 1.1) for
saturation properties and extrapolate for steam properties
= 1: compute all properties by calls to the HLWP property package

PLOOK - property evaluation switch:
= 0: evaluate all properties at the local (axial) pressure of the
reference subchannel
= 1: evaluate all properties at the reference (exit) pressure

FLUID - fluid (coolant) type switch, used by CHF and HLWP:
= 0: light water
= 1: light water
= 2: heavy water
= 3: user-specified fluid type

NPROP=0: no property table entries to read (ignored as NPACK=1)
/ NPACK=1: use HLWP property package
/ / NPLOOK=0: local subchannel 1 pressure
GROUP / / / NFLUID=2: heavy water coolant
/ / / /
0 1 0 2

cord 1.1: Saturated liquid and vapour properties (include if NPACK=0)
(I), TT(I), VVF(I), VVG(I), HHF(I), HHG(I), UUF(I), KKF(I), SSIGMA(I),
JG(I), KKG(I), (I = 1, NPROP) (E5.2, F5.1, 9F10.0)
- presure (MPa or psia)
- saturation temperature (degC or degF)
VF - liquid specific volume (m3/kg or ft3/lb)
VG - vapour specific volume (m3/kg or ft3/lb)
HF - liquid saturation enthalpy (kJ/kg or Btu/lb)
HG - vapour saturation enthalpy (kJ/kg or Btu/lb)
UF - liquid viscosity (kg/m.s or lb/ft.h)

```
* KKF   - liquid thermal conductivity (W/m.K or Btu/h.ft.F)
* SSIGMA - surface tension (N/m or lbf/ft)
* UUG   - vapour viscosity (kg/m.s or lb/ft.h) (optional)
* KKG   - vapour thermal conductivity (W/m.K or Btu/h.ft.F) (optional)
* Sample entry:
*9.0 303.9  0.001418  0.020510   1369.53   2742.05  0.000085  0.545403  0.014028
*
*=============================================================================
* Group 2:  Friction Factor and Two-Phase Flow Options
*-----------------------------------------------------------------------------
* NOPTN  - thermal equilbrium options:
*    = 0: thermal equilibrium
*    = 1: liquid thermal non-equilibrium (recommended)
*    = 2: vapour thermal non-equilibrium
*    = 3: liquid and vapour thermal non-equilibrium
*
* NFRIC  - turbulent single-phase friction factor options:
*    = 0: Blasius-type correlation of the form f = a*Re^b + c (Record 2.1)
*    = 1: Selander's approximation to the Colebrook-White formula (Record 2.2)
*    = 2: Colebrook-White implicit formula (recommended) (Record 2.2)
*    = 3: Chen's approximation to the Colebrook-White formula (Record 2.2)
*    = 4: Miller's approximation to the Colebrook-White formula (Record 2.2)
*    = 5: fully rough friction factor (Record 2.2)
*
* NMULT  - 2-phase friction multiplier options:
*    = 0: homogeneous model
*    = 1: Armand correlation, not recommended
*    = 2: homogeneous model with Owen's viscosity
*    = 3: homogeneous model with McAdam's viscosity
*    = 5: user-supplied polynomial (function of quality, up to 6th order)
*         (Record 2.5)
*    = 6: Friedel correlation (recommended)
*    = 8: Lorenc-Leung correlation, without subcooling
*
* NVISCW - heated-wall viscosity correction to friction factor:
*    = 0: wall viscosity correction not applied (recommended)
*    = 1: include wall viscisity correction (not operational)
*
* LAMNF  - laminar single-phase friction factor:
*    = 0: no laminar friction factor (default)
*    > 0: laminar correlation of the form f = a*Re^b + c (Record 2.3)
*
* N6     - rod-to-coolant single-phase heat-transfer (HT) coefficient:
*    = 0: Dittus-Boelter correlation (default)
*    > 0: read a user-supplied correlation (Record 2.4)
*
* NGRAV  - lateral momentum equation gravity term option:
*    = 0: no gravity terms (default)
*    = 1: include gravity terms (recommended)
*    = 2: include gravity terms in single-phase flow region only
*    = 3: include gravity terms in single-phase and two-phase bubbly flow
*         regions only (i.e., up to a void fraction of 30%)
*
* NUREL  - axial relative velocity option:
*    = 0: equal phasic velocities (mechanical/dynamic equilibrium) (default)
*    = 1: unequal phasic velocities (mechanical non-equilibrium) (recommended)
*    =-1: same as NUREL=1, but user supplies coefficients (Record 2.6)
*
* NVREL  - lateral relative velocity option:
*    = 0: equal phasic velocities (no slip) (default)
*    = 1: Ohkawa-Lahey buoyancy-drift model
*    = 2: modified Wallis buoyancy-drift model, accounts for gap width
*    = 3: modified Wallis buoyancy-drift model, accounts for gap width and
*         pressure effects (recommended)
*    < 0: same as ABS(NVREL), but user supplies coefficients (Record 2.7)
*
* NKMULT - 2-phase form loss multiplier options:
*    = 0: homogeneous model (recommended)
*    = 1: same as friction multiplier (NMULT)
*----------
*              NOPTN=1: liquid thermal non-equilibrium, vapour equilibrium
*            /   NFRIC=2: Colebrook-White 1-phase friction factor (Re>4000)
*           /   /   NMULT=0
*          /   /   /   NVISCW=0: no heated-wall viscosity correction
*         /   /   /   /   LAMNF=1: laminar friction f = a*Re^b + c
*        /   /   /   /   /   N6=0: Dittus-Boelter 1-phase HT coef.
*       /   /   /   /   /   /Re>=2000, .66>P>22 (MPa) liquid water
*      /   /   /   /   /   /   NGRAV=1: include gravity terms
*     /   /   /   /   /   /   /   NUREL=1: axial relative velocity
*    /   /   /   /   /   /   /   /   NVREL=1: transverse relative vel.
* NGROUP /   /   /   /   /   /   /   /   NKMULT=0: homog. K mult.
*  /   /   /   /   /   /   /   /   /   /
```

```
 1  2  0  0  1  0  1  1  1  0
---------------------------
ecord 2.1: Turbulent friction factor coefficients (include if NFRIC=0)
A(I), BB(I), CC(I), (I = 1, 4)                    (12F5.0)
  - f = AA*Re^BB + CC, for up to 4 subchannel types (Record 4.1)
1-.148 .101-.148
---------------------------
ecord 2.2: Subchannel roughness height (include if NFRIC>0)
UFHGT(I), (I = 1, 4)                              (6F10.0)
  - roughness height (cm or in), for up to 4 subchannel types
  - superceded by rod and PT roughness heights from Group 5
0005
---------------------------
ecord 2.3: Laminar friction factor coefficients (include if LAMNF>0)
AL(I), BBL(I), CCL(I), (I = 1, 4)                 (12F5.0)
  - f = AAL*Re^BBL + CCL, for up to 4 subchannel types
) -1.0 0.0
---------------------------
ecord 2.4: Single-phase HT correlation coefficients (include if N6>0)
H(I), (I = 1, 4)                                  (12E5.0)
  - h = (K/D)*(AH(1) * Re^AH(2) * Pr^AH(3) + AH(4))
6  0.0  0.0  0.0
---------------------------
ecord 2.5: Two-phase friction multiplier coefficients (include if NMULT=5)
F, AF(I), (I = 1, 7)                              (I5, 7E10.5)
  - TPMULT = AF(1) + AF(2) * X + AF(3) * X^2 + ... + AF(NF) * X^(NF-1)
  ...
---------------------------
ecord 2.6: Axial relative velocity coefficients (include if NUREL=-1)
UR(1) - coefficient of Ohkawa-Lahey mulitplier (default: 2.9)
UR(2) - exponent of Ohkawa-Lahey mulitplier (default: 1.0)   (12F5.0)
 1.5
---------------------------
ecord 2.7: Lateral relative velocity coefficients (include if NVREL<0)
 NVREL = -1, then                                 (12F5.0)
AVR(1) - coefficient in bubble-rise velocity (default: 1.5)
AVR(2) - exponent of Ohkawa-Lahey rise velocity multiplier (default: 2.0)
 NVREL = -2 or NVREL = -3, then
AVR(1) - coefficient of Wallis-type bubble rise velocity (default: 2.2)
AVR(2) - exponent of Wallis-type buoyancy-drift multiplier (default: 1.5)
AVR(3) - void fraction at bubble-to-slug transition (default: 0.35)
AVR(4) - exponent in gap size factor (default: 4.0)
 NVREL = -3, then
AVR(5) - void fraction at slug-to-annular transition (default: 0.7)
AVR(6) - exponent in pressure factor of buoyancy multiplier (default: 0.6)
 1.5  0.35 4.0  0.7  0.6
---------------------------
ecord 2.8: Two-phase heat transfer and CHF options        (10I5)
HTC   - heat-transfer correlation option:
= 0: forced convection Dittus-Boelter correlation (default if NOPTN=0)
  1: use correlation specified by NHFW (default if NOPTN>0)
    (recommended for calculations up to CHF)
  2: same as NHTC=1, plus simple PDO heat transfer (not recommended)
  3: same as NHTC=1, plus detailed PDO heat transfer
    (recommended for PDO calculations)

CHF   - critical heat flux (CHF) option:
= 0: no CHF calculations (default)
= 1: Govan & Hewitt film-dryout model for annualar flow
= 2: Weisman & Pei DNB model for bubbly flow
= 3: Biasi CHF correlation
= 5: LW-T-1996 table look-up method
= 9: use options 2, 3 and 5
=10: use MAPLE-X10 CHF calculations (also activates MAPLE simulations)

HFW   - two-phase wall-to-fluid heat transfer options:
= 0: Chen correlation (default)
= 1: Ahmad model and OSV criteria (recommended)
= 2: Rouhani-Axelsson model and ONB criteria
= 3: Chen correlation with refined interfacial area and reduced liquid
    superheat (ONB)
= 4: Hancox-Nicoll model and OSV criteria
= 5: Maroti model (ONB)
= 6: Lahey-Moody model (with Ahmad's OSV criteria)

CK1   - CHF table correction factor for subchannel hydraulic diameter
= 0: no correction applied (default)
= 1: apply diameter correction factor (recommended with NCK2=1)

CK2   - CHF table correction factor for inter-element gap
= 0: no correction applied (default)
= 1: apply gap correction factor (recommended with NCK1=1)
```

```
*
* NCK3  - CHF correction for flow stratification in a horizontal channel:
*    = 0: no correction applied (default)
*    = 1: apply orientation correction factor (recommended)
*
* NCK4  - CHF enhancement factor for bundle appendages:
*    = 0: no enhancement applied (default)
*    = 1: apply CHF enhancement (recommended)
*
* NCK5  - option for boiling length average (BLA) heat flux with CHF table:
*    = 0: BLA approach not used (default, recommended)
*    = 1: use BLA heat flux
*---------
*        NHTC=0: forced conv Dittus-Boelter only
*       /  NCHF=0: no CHF calcs
*      /  /  NHFW=0: original Chen wall-fluid heat transfer
*     /  /  /  NCK1=0: for hyd. dia., no corr.
*    /  /  /  /  NCK2=0:for gap, no corr.
*   /  /  /  /  /  NCK3=0: for flow stratfification, no corr.
*  /  /  /  /  /  /  NCK4=0: for CHF enhance, no corr.
* /  /  /  /  /  /  /  NCK5=0: no BLA correction to CHF
* /  /  /  /  /  /  /  /
  0  0  0  0  0  0  0  0
*
*=============================================================================
* Group 3:  Axial Heat Flux Distribution
*-----------------------------------------------------------------------------
*   Exit-skewed cosine axial heat flux profile from the Stern 37-element CHF
* experiments (Ref. 5).  This profile reflects the unheated bundle end-plates
* at the beginning and the end of the channel, each 0.75 cm long.
*   The heat fluxes are normalized by the average heat flux over the HEATED
* length.
*-------------------------------------------------
* NQXOPT - axial heat flux profile options:
*    = 0: axial power factors (Record 3.1)
*    = 1: power factors for each axial zone (Record 3.2)
*    = 2: CANDU bundle power factors for each axial zone (Record 3.3)
*
* NAX   - number of entries in heat flux table
*---------
*       NQXOPT=0: use axial power factors from Record 3.1
* NGROUP /   NAX=27: number of entries in heat flux table
* /  /  /
  3  0  27
*-------------------------------
* Record 3.1:  Axial heat flux table (include if NQXOPT=0)
* Y(I), AXIAL(I), (I = 1, NAX)                    (12F5.0)
* Y     - relative axial location (X/L), include end-points 0 and 1
* AXIAL  - relative heat flux (local/average)
*    Y       Y       Y       Y       Y       Y
*   / AXIAL / AXIAL / AXIAL / AXIAL / AXIAL / AXIAL
*  /  /  /  /  /  /  /  /  /  /  /
 0.0 0.0 .038 .190 .077 .377 .115 .558 .154 .731 .192 .893
.2311.043 .2691.177 .3081.294 .3461.392 .3851.470 .4231.525
.4621.560 .5001.574 .5381.560 .5771.525 .6151.470 .6541.392
.6921.294 .7311.177 .7691.043 .808 .893 .846 .731 .885 .558
.923 .377 .962 .190  1.0  0.0
*-------------------------------
* Record 3.2:  Nodal heat flux table (include if NQXOPT=1)
* QXJ(I), (I = 1, NAX)                            (12F5.0)
* QXJ   - axial power factor for each axial zone
* 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
*-------------------------------
* Record 3.3:  Bundle heat flux table (include if NQXOPT=2)
* QBNDL(I), (I = 1, NAX)                           (12F5.0)
* QBNDL  - constant axial power factor for heated length of each bundle
* 1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
*=============================================================================
*  Group 4:  Subchannel Layout and Dimension Data
*-----------------------------------------------------------------------------
*  37-Rod Half Bundle                G.M. Waddington  1997-11-19
*  Nominal (cold) dimensions from Ref. 3, eccentric bundle, nominal PT.
*  Radial flux distribution (RFD) from Ref. 4 (burnup of ~50 MWh/kgU).
*  Produced by GENGEOM V4R1 using datagg_candu37_half.
*  Fuel Channel Dimensions:
*    Pressure tube inside diameter     = 10.338 cm
*    Bundle diameter (incl bearing pads) = 10.212 cm
*    Rod diameters                     = 1.312 cm
*      Ring       # Rods  Diameter  Offset   RFD
*    - centre        1     0.0        0     0.784
*    - inner         6     2.980     30     0.815
*    - intermediate  12    5.760     15     0.917
```

```
 - outer       18    8.660    10    1.129
Computed Total Dimensions:
 - flow area       = 16.95852 cm2
 - wetted perimeter  = 92.49163 cm
 - heated perimeter  = 76.25274 cm
 - hydraulic diameter =  0.73341 cm
HOT DIMENSIONS WOULD BE MORE APPROPRIATE FOR CHF CALCULATIONS.
-------------------------------------------------------------------------------
      ***SUBCHANNEL AND ROD GEOMETRY INFORMATION***

CHANL ----> total number of subchannels (N1)
ROD  ----> total number of rods (N2)

      Axial variation data-specification (N3)
XVOPT = 0: no axial variation data specified
    1: user specified axial variation data

      Subchannel geometry:
TYPE  ---> subchannel type up to 4 types, also connected to NFRIC
   ---> subchanne identification number
REASC ---> nominal subchannel area
PERIM ---> nominal subchannel wetted perimeter
PERIM ---> nominal subchannel heated perimeter
C     ---> adjacent subchannel identification number
APWDS ---> the gap width between subchannel I and the adjacent subchannel
APDSS ---> distance between the centroid of subchannel I and the adjacent
      subchannel
APANG ---> angular orientation of centroid through the gap. 0 degree is
      vertical upward for a horizintal channel. For vertical channels,
      any reference is suitable for 0 deg.

      Rod geometry:
FUEL ---> fuel shape and fuel material option
   ---> rod number
ODDIA ---> outer rod diameter. If there is cladding aroud the rod, RODDIA is
      the cladding outer diameter
ODFLX ---> radial heat-flux factor for rod I as a fraction of the average rod
      power
R     ---> identification number of subchannels surrounding Rod I
HI    ---> the fraction of the total rod power input to adjacent subchannel
      (i.e., the fraction of the outer rod perimeter facing facing the
      subchannel identified by LR)

      Axial geometry:
HETA  ---> channel orientation
   = 0: vertical
    90: horizontal
BNDLS ---> number of fuel bundles (this option not fully functional, choose
      the default 1)
NDLTL ---> length of one bundle (this option not fully functional, choose
      the total channel length)
NDLHL ---> heated length of one bundle (this option not fully functional,
      the total channel length choose)
-------------------------------------------------------------------------------
     NCHANL: number of subchannels
    /   NROD: number of rods
roup   /   /   AXVOPT=0: no axial variations
  /   /   /
 34  19   0

- Subchannel Geometry Data (Record 4.1)
     NTYPE: subchannel type
    /  I: subchannel number
   /  /   AREASC: flow area (cm2)
  /  /  /   WPERIM: wetted perimeter (cm)
 /  /  /  /   HPERIM: heated perimeter (cm)
/  /  /  /  /   LC: index of neighbouring subchannel
/  /  /  /  /  /   GAPWDS: gap width (cm)
/  /  /  /  /  /  /   GAPDSS: centroid-to-centroid distance (cm)
/  /  /  /  /  /  /  /   GAPANG: centroid-to-centroid angle (degrees clockwise from vertical)
/  /  /  /  /  /  /  /              LC, GAPWDS, GAPDSS, GAPANG
/  /  /  /  /  /  /  /  _____________/\_____________
/  /  /  /  /  /  /  /  /  /  /  /  /  |  |  |  |
 1.14271.0301.030   2.1780.8603120.0  -5.17801.176 0.00
 2.28542.0612.061   3.1780.8603180.0   7.17801.17660.00
 3.28542.0612.061   4.1780.8603240.0   9.17801.176120.0
 4.14271.0301.030  -11.17801.176180.0
 5.43552.0612.061   6.17951.17690.02  -12.17881.491 0.00
 6.28702.0612.061   7.17951.176150.0   14.1788.999130.00
 7.87104.1224.122   8.17951.176150.0   16.17881.49160.00
 8.28702.0612.061   9.17951.176210.0   18.1788.999190.00
 9.87104.1224.122  10.17951.176210.0   20.17881.491120.0
```

```
1  10.28702.0612.061  11.17951.176270.0  22.1788.9991150.0
1  11.43552.0612.061  -24.17881.491180.0
1  12.43382.0612.061  13.17041.25488.46  -25.19181.400 0.00
1  13.37312.0612.061  14.4062.7825147.4  26.19181.15021.84
1  14.47802.0612.061  15.4062.782592.56
1  15.37312.0612.061  16.17041.254151.5  27.19181.14337.77
1  16.86764.1224.122  17.17041.254148.5  28.19181.38059.72
1  17.37312.0612.061  18.4062.7825207.4  29.19181.11981.61
1  18.47802.0612.061  19.4062.7825152.6
1  19.37312.0612.061  20.17041.254211.5  30.19181.10597.51
1  20.86764.1224.122  21.17041.254208.5  31.19181.339119.7
1  21.37312.0612.061  22.4062.7825267.4  32.19181.080141.8
1  22.47802.0612.061  23.4062.7825212.6
1  23.37312.0612.061  24.17041.254271.5  33.19181.073157.7
1  24.43382.0612.061  -34.19181.317180.0
1  25.40992.0581.145  26.24501.708100.1
1  26.81274.1152.290  27.23751.707120.2
1  27.79274.1112.290  28.22331.704140.3
1  28.76214.1052.290  29.20421.701160.3
1  29.72494.0982.290  30.18261.697180.4
1  30.68574.0902.290  31.16111.693200.4
1  31.64914.0832.290  32.14231.690220.3
1  32.61944.0772.290  33.12831.687240.2
1  33.60024.0742.290  34.12091.686260.1
1  34.29682.0361.145
*
*----- Rod Geometry and RFD Data (Record 4.2)
*      IDFUEL: >=0 cylindrical fuel specified (default 0)
*     /  I: rod number
*    /  /   RODDIA: outside diameter of rod (cm)
*   /  /  /   RODFLX: rod heat flux normalized by average heat flux
*  /  /  /  /   LR: subchannel adjacent to rod
* /  /  /  /  /   PHI: fraction of rod power to subchannel LR
*/  /  /  /  /  /   LR  PHI  LR  PHI  LR  PHI  LR  PHI  LR  PHI  LR  PHI
*/  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /  /
   11.3120.784   1.0833   2.1667   3.1667   4.0833
   21.3120.815   1.1667   2.1667   5.2500   6.1666   7.2500
   31.3120.815   2.1667   3.1667   7.2500   8.1666   9.2500
   41.3120.815   3.1667   4.1667   9.2500  10.1666  11.2500
   51.3120.917   5.2500   6.1667  12.2507  13.1541  14.1786
   61.3120.917   6.1667   7.2500  14.1786  15.1541  16.2507
   71.3120.917   7.2500   8.1667  16.2507  17.1541  18.1786
   81.3120.917   8.1667   9.2500  18.1786  19.1541  20.2507
   91.3120.917   9.2500  10.1667  20.2507  21.1541  22.1786
  101.3120.917  10.1667  11.2500  22.1786  23.1541  24.2507
  111.3121.129  12.2493  13.1951  25.2778  26.2778
  121.3121.129  13.1508  14.1428  15.1508  26.2778  27.2778
  131.3121.129  15.1951  16.2493  27.2778  28.2778
  141.3121.129  16.2493  17.1951  28.2778  29.2778
  151.3121.129  17.1508  18.1428  19.1508  29.2778  30.2778
  161.3121.129  19.1951  20.2493  30.2778  31.2778
  171.3121.129  20.2493  21.1951  31.2778  32.2778
  181.3121.129  21.1508  22.1428  23.1508  32.2778  33.2778
  191.3121.129  23.1951  24.2493  33.2778  34.2778
*
*----- Axial Dimensions (Record 4.3)
*      THETA: channel angle from vertical (degrees)
*     /   NBNDLS: number of bundles
*    /  /     BNDLTL: bundle total length (cm)
*   /  /   /     BNDLHL: bundle heated length (cm)
*  /  /    /    /
*  90   1  594.36 594.36     * whole channel
   90  12   49.53  48.03     * bundle lengths
*=============================================================================
*  Group 7:  Appendage Form Loss Data
*-----------------------------------------------------------------------------
*    This group specifies the geometry data for these appendage types:
*    1) junction: bundle endplates
*    2) junction: flow area expansion
*    3) bearing pad plane
*    4) midplane: spacer pads
*    5) midplane: bearing pads
*  from which the pressure loss factors in each subchannel are computed using
*  Idelchik's 1994 thick-edged orifice formula.
*-------------------------------------------
*  J6    - subchannel form loss data options:
*     = 0: no axial form loss factors specified
*     = 2: specify pressure drop coefficients directly (Records 7.1, 7.2 & 7.3)
*     = 4: compute K-factors from geometry, Weisbach correlation (7.1,7.2,7.5)
*     = 5: compute K-factors from geometry, Idelchik formula (7.1,7.2,7.4,7.5)
*
*  NGRID  - number of axial locations for form losses
```

NGRIDT - number of form loss types for which data is supplied
---
J6=5: compute K-factors using Idelchik's orifice formula
/ N2: not used
/ / NGRID: number of axial locations
NGROUP / / / NGRIDT: number of form loss types
/ / / / /
5 0 98 5

Record 7.1: Axial location and appendage type (include if J6>0)
GRIDXL(I), IGRID(I), (I = 1, NGRID) (6(E5.0,I5))
GRIDXL - relative axial location (X/L)
IGRID - loss type at axial location GRIDXL
GRIDXL GRIDXL GRIDXL GRIDXL GRIDXL GRIDXL
/ IGRID / IGRID / IGRID / IGRID / IGRID / IGRID
/ / / / / / / / / / / /
02 1 .0002 2 .0049 3 .0155 3 .0417 4 .0417 5
79 3 .0786 3 .0834 1 .0834 2 .0882 3 .0989 3
51 4 .1251 5 .1512 3 .1619 3 .1667 1 .1667 2
15 3 .1822 3 .2084 4 .2084 5 .2346 3 .2452 3
01 1 .2501 2 .2549 3 .2655 3 .2917 4 .2917 5
79 3 .3286 3 .3334 1 .3334 2 .3382 3 .3489 3
51 4 .3751 5 .4012 3 .4119 3 .4167 1 .4167 2
15 3 .4322 3 .4584 4 .4584 5 .4846 3 .4952 3
01 1 .5001 2 .5048 3 .5155 3 .5417 4 .5417 5
79 3 .5786 3 .5834 1 .5834 2 .5882 3 .5989 3
51 4 .6251 5 .6512 3 .6619 3 .6667 1 .6667 2
15 3 .6822 3 .7084 4 .7084 5 .7346 3 .7452 3
01 1 .7501 2 .7549 3 .7655 3 .7917 4 .7917 5
79 3 .8286 3 .8334 1 .8334 2 .8382 3 .8489 3
51 4 .8751 5 .9012 3 .9119 3 .9167 1 .9167 2
15 3 .9322 3 .9584 4 .9584 5 .9846 3 .9952 3
98 1 .9998 2

Include one copy of Record 7.2, plus Records 7.3, 7.4 or 7.5 (determined
by J6), for each form loss type I (of NGRIDT types).

Record 7.2: Appendage loss factors (include if J6>0)
NGRIDC (I5)
NGRIDC - number of subchannels for which data are supplied

Record 7.3: Appendage loss factors (include if J6=2)
CD(J,I), PSP(J,I), (J = 1, NGRIDC) (I5,2E5.2)
- subchannel number for which data is supplied
CD - loss coefficient in subchannel J, for loss type I (IGRID)
PSP - subchannel area obstruction ratio, for CHF enhancement (Record 2.7)

Record 7.4: Coefficients for Idelchik orifice formula (include if J6=5)
GBC(I,JTERM), JTERM = 1, 3), GBNAME(I) (3E10.5,A20)
GBC(I,1) - coefficient of contraction term for loss type I (square edge: 0.5)
GBC(I,2) - coefficient of length term for loss type I (range: 0.0 to 1.35)
=-1: Idelchik's formula for wall thickness coefficient (recommended)
GBC(I,3) - coefficient of expansion term for loss type I (recommended: 1.0)
GBNAME - descriptive name for loss type (e.g., bundle endplate)

Record 7.5: Appendage geometry data (include if J6=4 or J6=5)
GBDA(J,I), GBDPW(J,I), GBL(J,I), (J = 1, GRIDC) (I5,3E10.5)
- subchannel number for which data is supplied
GBDA - change in subchannel J flow area (i.e., obstruction area) for type I
GBDPW - change in subchannel J wetted perimeter, for loss type I
GBL - length of obstruction in subchannel J for loss type I

Type 1: Bundle Endplate
NGRIDC: number of subchannels with endplate geometry data

GBC(1)=0.5: contraction term coefficient for a sharp edge
/ GBC(2)=-1: use formula for wall thickness coefficient
/ / GBC(3): coefficient of expansion term
/ / / GBNAME: descriptive name
/ / / /
5 -1. 1.0 Bundle Endplate

J: subchannel number
/ GBDA: change in subchannel flow area (cm2)
/ / GBDPW: change in subchannel wetted perimeter (cm)
/ / / GBL: length of obstruction (cm)
/ / /
0.000000 0.00000 0.3175
0.148725 0.01230 0.3175
0.000000 0.00000 0.3175

```
  4 0.000000  0.00000   0.3175
  5 0.154485  0.08160   0.3175
  6 0.013324  0.04090   0.3175
  7 0.102533 -0.32110   0.3175
  8 0.013324  0.04090   0.3175
  9 0.308969  0.16320   0.3175
 10 0.013324  0.04090   0.3175
 11 0.049892 -0.16055   0.3175
 12 0.110192 -0.19765   0.3175
 13 0.037191 -0.15210   0.3175
 14 0.058056 -0.25290   0.3175
 15 0.037191 -0.15210   0.3175
 16 0.220384 -0.39530   0.3175
 17 0.037191 -0.15210   0.3175
 18 0.058056 -0.25290   0.3175
 19 0.037191 -0.15210   0.3175
 20 0.220384 -0.39530   0.3175
 21 0.037191 -0.15220   0.3175
 22 0.058056 -0.25290   0.3175
 23 0.037191 -0.15210   0.3175
 24 0.110192 -0.19765   0.3175
 25 0.037449 -0.13140   0.3175
 26 0.074898 -0.26270   0.3175
 27 0.074898 -0.26240   0.3175
 28 0.074898 -0.26180   0.3175
 29 0.074898 -0.26130   0.3175
 30 0.074898 -0.26070   0.3175
 31 0.074898 -0.26010   0.3175
 32 0.074898 -0.25970   0.3175
 33 0.074898 -0.25930   0.3175
 34 0.037449 -0.12960   0.3175
*
* -- Type 2:  Junction Gap Flow Area Expansion
*   NGRIDC: all subchannels expand into junction gap
*  /
  34
*       GBC(1)=0.38: function of jet expansion rate into gap
*     /       GBC(2)=0: no length term
*    /      /       GBC(3): coefficient of expansion term
*   /      /      /       GBNAME: descriptive name
*  /      /      /      /
  0.38     0.0     1.0   Junction Gap
*
*      J: subchannel number
*     /       GBDA: change in subchannel flow area (cm2)
*    /      /       GBDPW: change in subchannel wetted perimeter (cm)
*   /      /      /       GBL: length of obstruction (cm)
*  /      /      /      /
   1 -0.06710  0.0000   0.657
   2 -0.13420  0.0000   0.657
   3 -0.13420  0.0000   0.657
   4 -0.06710  0.0000   0.657
   5 -0.12975  0.0000   0.657
   6 -0.13407  0.0000   0.657
   7 -0.25950  0.0000   0.657
   8 -0.13407  0.0000   0.657
   9 -0.25950  0.0000   0.657
  10 -0.13407  0.0000   0.657
  11 -0.12975  0.0000   0.657
  12 -0.12975  0.0000   0.657
  13 -0.13090  0.0000   0.657
  14 -0.12888  0.0000   0.657
  15 -0.13090  0.0000   0.657
  16 -0.25924  0.0000   0.657
  17 -0.13090  0.0000   0.657
  18 -0.12888  0.0000   0.657
  19 -0.13090  0.0000   0.657
  20 -0.25924  0.0000   0.657
  21 -0.13090  0.0000   0.657
  22 -0.12888  0.0000   0.657
  23 -0.13090  0.0000   0.657
  24 -0.12962  0.0000   0.657
  25 -0.06295  0.0000   0.657
  26 -0.12589  0.0000   0.657
  27 -0.12586  0.0000   0.657
  28 -0.12582  0.0000   0.657
  29 -0.12580  0.0000   0.657
  30 -0.12582  0.0000   0.657
  31 -0.12585  0.0000   0.657
  32 -0.12591  0.0000   0.657
  33 -0.12598  0.0000   0.657
  34 -0.06300  0.0000   0.657
```

Type 3: Bearing Pad Plane
NGRIDC: number of subchannels obstructed by bearing pads
/
0
GBC(1)=0.2: contraction term coefficient for the bevelled edge
/ GBC(2)=-1: use formula for wall thickness coefficient
/ / GBC(3): coefficient of expansion term
/ / / GBNAME: descriptive name
/ / / /
2 -1. 1.0 Bearing Pad Plane

J: subchannel number
/ GBDA: change in subchannel flow area (cm2)
/ / GBDPW: change in subchannel wetted perimeter (cm)
/ / / GBL: length of obstruction (cm)
/ / / /
5 0.00935 0.0668 2.90
6 0.01869 0.1335 2.90
7 0.01869 0.1335 2.90
8 0.01869 0.1335 2.90
9 0.01869 0.1335 2.90
0 0.01869 0.1335 2.90
1 0.01869 0.1335 2.90
2 0.01869 0.1335 2.90
3 0.01869 0.1335 2.90
4 0.00935 0.0668 2.90

Type 4: Midplane Spacer Pads
NGRIDC: all subchannels are obstructed by midplane spacers
/

GBC(1)=0.2: contraction term coefficient for the rounded edge
/ GBC(2)=-1: use formula for wall thickness coefficient
/ / GBC(3): coefficient of expansion term
/ / / GBNAME: descriptive name
/ / / /
2 -1. 1.0 Midplane Spacer Pads

J: subchannel number
/ GBDA: change in subchannel flow area (cm2)
/ / GBDPW: change in subchannel wetted perimeter (cm)
/ / / GBL: length of obstruction (cm)
/ / / /
0.03567 0.2700 0.8555
0.07134 0.5400 0.8555
0.07134 0.5400 0.8555
0.03567 0.2700 0.8555
0.04756 0.3600 0.8555
0.07134 0.5400 0.8555
0.09512 0.7200 0.8555
0.07134 0.5400 0.8555
0.09512 0.7200 0.8555
0.07134 0.5400 0.8555
0.04756 0.3600 0.8555
0.04756 0.3600 0.8555
0.10292 0.7790 0.8555
0.13450 1.0180 0.8555
0.10292 0.7790 0.8555
0.09512 0.7200 0.8555
0.10292 0.7790 0.8555
0.13450 1.0180 0.8555
0.10292 0.7790 0.8555
0.09512 0.7200 0.8555
0.10292 0.7790 0.8555
0.13450 1.0180 0.8555
0.10292 0.7790 0.8555
0.04756 0.3600 0.8555
0.01189 0.0900 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.02378 0.1800 0.8555
0.01189 0.0900 0.8555

Type 5: Midplane Bearing Pads
NGRIDC: number of subchannels obstructed by midplane bearing pads

```
 10
*      GBC(1)=0.2: contraction term coefficient for the bevelled edge
*     /       GBC(2)=-1: use formula for wall thickness coefficient
*    /      /       GBC(3): coefficient of expansion term
*   /      /      /       GBNAME: descriptive name
*  /      /      /      /
  0.2     -1.     1.0   Midplane Bearing Pads
*
*      J: subchannel number
*     /       GBDA: change in subchannel flow area (cm2)
*    /      /       GBDPW: change in subchannel wetted perimeter (cm)
*   /      /      /       GBL: length of obstruction (cm)
*  /      /      /      /
  25  0.01869  0.1335   2.90
  26  0.03738  0.2670   2.90
  27  0.03738  0.2670   2.90
  28  0.03738  0.2670   2.90
  29  0.03738  0.2670   2.90
  30  0.03738  0.2670   2.90
  31  0.03738  0.2670   2.90
  32  0.03738  0.2670   2.90
  33  0.03738  0.2670   2.90
  34  0.01869  0.1335   2.90
*-------------------------------
* Record 7.6:  Cross-flow form loss data
*    KIJ: inter-subchannel gap loss coefficient (default: 0.5)
*   /
 1.0
*===============================================================================
* Group 9:  Calculation Control
*-------------------------------------------------------------------------------
* NSKIPX - output print option for axial nodes:
* = 0 or 1: print at all axial nodes
*     > 1: print every NSKIPX axial nodes
*
* NSKIPT - output print option for time steps:
* = 0 or 1: print at all time steps
*     > 1: print every NSKIPT time steps
*
* NDXOPT - axial discretization options:
*     = 0: uniform axial zone length
*     = 1: user-specified axial zone length (Records 9.4 & 9.5)
*     = 2: CANDU bundles, uniform grid in the heated section (Record 9.4)
*     = 3: CANDU bundles, zone lengths in the heated section change by a
*          constant ratio, one zone for unheated bundle junction (Record 9.4)
*
* MFLOWX - axial momentum equation inertia term option:
*     = 0: include the inertia term (default)
*     = 1: neglect the inertia term for all nodes
*     > 1: neglect the inertia term for the first MFLOWX nodes
*
* MFLOWY - lateral momentum equation inertia term option:
*     = 0: include the inertia term (default)
*     = 1: neglect the inertia term (may improve convergence in some cases)
*
* REINIT - re-initialization option:
*     = 0: re-initialize the field variables (default)
*     = 1: keep the solution of the previous case as the starting point for
*          the solution of the current case (not applicable for first case)
*---------
*            NSKIPX=0: print at all axial nodes
*           /  NSKIPT=0: print at all time steps
*          /  /          N3, N4, N5: not used
*         /  /          /  NDXOPT=0: uniform axial zone length
*        /  /          /  /   N7: not used
*       /  /          /  /  /   MFLOWX=0: include axial inertia
*      /  /  __________/  /  /  /   MFLOWY=0: incl. lat. inertia
* NGROUP /  /  /  /  /  /  /  /  /  /  REINIT=0: reset field
*  /  /  /  /  /  /  /  /  /  /  /        variables
   9  0  0  0  0  0  0  0  0  0  0  0
*-------------------------------
* Record 9.1:  Iteration and convergence control          (4I5,5F10.0)
* IELMT - maximum number of energy (enthalpy) iterations
* MAXINR - maximum number of inner (mass) iterations
* NTRIES - maximum number of outer (flow) iterations
* IREBAL - inner iteration frequency for global pressure rebalancing
* HERROR - relative enthalpy change criteria (Dh/h)
* EERROR - relative flow change criteria (Df/f)
* MERROR - mass equation relative residual error criteria
* TERROR - lateral momentum residual error criteria (N)
* AXEROR - axial momentum residual error criteria (N)
*    IELMT   NTRIES
```

/ MAXINR / IREBAL HERROR EERROR MERROR TERROR AXEROR
/ / / / / / / /
35 120 1 1.E-5 1.E-4 1.E-5 1.E-5 1.E-5

---

ecord 9.2: Time step control (2(F10.0,I5))
SSDT: steady-state time step size (default: 10^15)
/ NSSS: number of steady-state time steps (default: 1)
/ / TTDT: transient time step size (default: 0)
/ / / NDT: number of transient time steps (default: 0)
/ / / /
E15 1 0.5 0

---

ecord 9.3: Relaxation parameters (4E5.0)
WRELAX: lateral momentum relaxation parameter (default: 1)
/ FRELAX: axial momentum relaxation parameter (default: 1)
/ / RRELAX: density relaxation parameter (default: 1)
/ / / HRELAX: enthalpy relaxation parameter (default: 1)
/ / /
1.0

---

ecord 9.4: Number of Axial Zones (*)
NHTDX: number of axial zones in each bundle heated length
/ - if there is an unheated length (i.e., BNDLHL < BNDLTL), then the
/ total number of zones is NDX = (1 + NBNDLS * (NHTDX + 1)),
- otherwise NDX = NBNDLS * NHTDX

---

ecord 9.5: Discretization by Axial Zones (include if NDXOPT=1)
)NODE(J), X(J), (J = 2, NDX+1) (6(I5,E5.2))
)NODE - type of zone (from X(J-1) to X(J))
= 0: normal heated zone (default)
= 1: unheated inlet-end zone
= 2: unheated bundle-junction zone
= 3: unheated exit-end zone
- axial location of end of zone (from inlet at X(1) = 0)
IDNODE IDNODE IDNODE IDNODE IDNODE IDNODE
X / X / X / X / X / X
/ / / / / / / / / / / /
5.237 0 10.47 0 20.44 0 25.68 0 28.43 0 29.88

627.6 0 628.4 0 629.8 0 632.5 0 637.4 0 646.8

---

roup 10: Mixing Models and Parameters

AMIX - thermal and momentum (TM) mixing and void diffusion (VD) options
= 0: no mixing
= 1: Carlucci TM and VD mixing (one equivalent coefficient is used for
VD mixing of both phases) (recommended)
= 2: Carlucci TM and VD mixing (separate VD coefficients for each phase)
= 3: Carlucci TM and Rowe VD mixing
= 4: Carlucci TM and modified Shoukri VD mixing
= 5: Rogers-Rosehart TM and modified Shoukri VD mixing

ALFAQ - equilibrium void (EV) fraction options
= 0: no equilibrium void
= 1: Rowe et al. EV model (new for V3R0) (recommended)
= 2: Lahey EV model (from V2R8)

SPMIX - homogeneous flow TM mixing flag for subchannel pairs (NAMIX=1 to 4)
=0: sets defaults for triangles ATR=0.0018, BTR=-0.4,
and for squares ASQ=0.005, BSQ=0.106 (default)
=1: user specifies parameter values (Record 10.1)

INMIX - incremental VD mixing multiplier flag (NAMIX=1 or 2)
=0: sets INMULT=3.0 (default)
=1: user specifies value for INMULT (Record 10.2)

OBFAC - TM mixing obstruction factor flag (NAMIX=1 to 4)
=0: sets OFA=3.3 and OFB=0.13 (default)
=1: user specifies values for OFA and OFB (Record 10.3)

GPFAC - TM mixing mass flux and pressure dependent factor flag (NAMIX>0)
=0: sets FGPA=11.0 and FGPB=0.9 (default)
=1: user specifies values for FGPA and FGPB (Record 10.4)

ROWVM - Rowe et al. VD coefficient flag (NAMIX=3)
=0: sets ROWVM=0.026 (default)
=1: user specifies value for ROWVM (Record 10.5)

ASSVM - modified Shoukri VD coefficients flag (NAMIX=4 or 5)
=0: sets VD coefficient ASSVM1=0.05 and

```
*        exponent of Ohkawa-Lahey VD multiplier ASSVM2=1.5 (default)
*     =1: user specifies values for ASSVM1 and ASSVM2 (Record 10.6)
*
* NRWEQV - Rowe et al. EV coefficient flag (NAMIX>0, NALFAQ=1)
*     =0: sets AEVR=0.85 (default)
*     =1: user specifies value for AEVR (Record 10.7)
*----------
*              NAMIX=5: Rogers and Rosehart
*             /typical CANDU pressures and mass fluxes: air,water
*            /   NALFAQ=1: default Rowe EV model
*           /   /0.1>P>0.4MPa, 0.45>G>0.09Mg/m^2s: air, water
*          /   /   NSPMIX=0: default homogeneous TM mixing parameters
*         /   /   /   NINMIX=0: default incremental VD multiplier
*        /   /   /   /   NOBFAC=0: default TM obstruction factor
*       /   /   /   /   /   NGPFAC=0: default G- and P-dependent TM
*      /   /   /   /   /   /   NROWVM=0: default Rowe VD
*     /   /   /   /   /   /   /   NASSVM=0: default Shoukri VD
* NGROUP /   /   /   /   /   /   /   /   NRWEQV=0: default Rowe EV
*   /   /   /   /   /   /   /   /   /   /
  10   5   1   0   0   0   0   0   0   0
*------------------------------
* Records 10.1 to 10.7:  Mixing options parameter values      (12E5.0)
*----------
* Record 10.1:  TM mixing parameters (include if NAMIX=1 to 4 and NSPMIX=1)
*      ATR: coefficient in mixing relation for triangular arrays
*     /   BTR: exponent in triangular array mixing relation
*    /   /   ASQ: coefficient in mixing relation for square arrays
*   /   /   /   BSQ: exponent in square array mixing relation
*  /   /   /   /
*.0018 -0.4 .005 .106
*------------------------------
* Record 10.2:  Incremental VD mixing (include if NAMIX=1 or 2 and NINMIX=1)
*    INMULT: multiplier for incremental void mixing coefficient
*   /
* 3.0
*------------------------------
* Record 10.3:  Obstruction factor (include if NAMIX=1 to 4 and NOBFAC=1)
*     OFA: coefficient of factor to enhance TM mixing near obstructions
*    /    OFB: obstruction factor exponent
*   /   /
* 3.3  0.13
*------------------------------
* Record 10.4:  Mass-flux and pressure factor (include if NAMIX>0 and NGPFAC=1)
*      FGPA: coefficient in factor accounting for mass-flux and pressure
*     /     effects on TM mixing (set FGPA=0 to turn off factor)
*    /   FGPB: 2nd coefficient in mass-flux and pressure factor
*   /   /
*11.0  0.9
*------------------------------
* Record 10.5:  Rowe VD mixing (include if NAMIX=3 and NROWVM=1)
*    ROWVM: coefficient for Rowe et al. void diffusion model
*   /
*.026
*------------------------------
* Record 10.6:  Shoukri VD mixing (include if NAMIX=4 or 5 and NASSVM=1)
*     ASSVM1: modified Shoukri turbulent VD mixing coefficient
*    /   ASSVM2: exponent of Ohkawa-Lahey VD mixing multiplier
*   /   /
*0.05  1.5
*------------------------------
* Record 10.7:  Rowe EV model (include if NAMIX>0, NALFAQ=1 and NRWEQV=1)
*    AEVR: coefficient for Rowe et al. equilibrium void model
*   /
*0.85
*
*=========================================================================================
* Group 11:  Operating Conditions and Transient Forcing Functions
*-----------------------------------------------------------------------------------------
* IH    - thermal inlet/outlet boundary condition options:
*     = 0: specify flow specific enthalpy THXIN or THXOUT (Record 11.1)
*     = 1: specify temperature THXIN or THXOUT (Record 11.1)
*     = 2: specify specific enthalpy for each subchannel (Record 11.2)
*     = 3: specify inlet temperature for each subchannel (Record 11.2)
*     = 4: specify flow equilibrium quality THXIN or THXOUT (Record 11.1)
*     = 5: specify inlet equilibrium quality for each subchannel (Record 11.2)
*
* IG    - flow inlet/outlet boundary condition (Records 11.1 & 11.2) options:
*     = 0: specify uniform INLET mass flux GIN (Record 11.1) (default)
*     = 1: average INLET mass flux GIN (11.1) is split to equalize the axial
*          friction and form-loss pressure drop in each subchannel
*     = 2: average INLET mass flux GIN (11.1) is split between subchannels by
*          specified flow fractions (Record 11.3)
```

=10: specify uniform EXIT mass flux GIN (11.1) to each subchannel
=12: average EXIT mass flux GIN (11.1) is split by flow fractions (11.3)

IP - transient forcing function for system pressure:
= 0: no pressure transient (default)
> 1: read NP pairs of time and pressure ratio (Record 11.4)

IH - transient forcing function for inlet thermal BC:
= 0: no inlet thermal BC transient (default)
> 1: read NH pairs of time and inlet thermal BC ratio (11.5)

IG - transient forcing function for boundary flow or pressure drop:
= 0: no flow or pressure drop transient (default)
> 1: read NG pairs of time and mass flow rate ratio (if DPS=0), or
time and pressure drop or pressure drop factor (Record 11.6)

IQ - transient forcing function for average heat flux:
= 0: no heat flux transient (default)
> 1: read NQ pairs of time and heat flux ratio (Record 11.7)

IDPTBL - transient pressure-drop boundary condition (NG) option:
= 0: Record 11.6 specifies pressure drop versus time (default)
= 1: Record 11.6 specifies pressure-drop ratio versus time

IHX - transient forcing function for exit enthalpy or temperature:
= 0: no exit enthalpy/temperature transient (default)
> 1: read NHX pairs of time and exit enthalpy/temperature ratio (11.8)

LENIN - inlet plenum option:
= 0: no inlet plenum; use standard inlet boundary conditions (default)
= 1: use inlet plenum; apply BCs to reference subchannel IREFIN (11.1)

LENEX - exit plenum option:
= 0: no exit plenum (default)
= 1: use exit plenum; apply BCs to reference subchannel IREFEX (11.1)

)DVMAX - onset-of-dryout iteration option:
= 0: no onset-of-dryout iterations (default)
> 1: iterate up to ODVMAX times for the onset of dryout (Record 11.9)

IH=1: THXIN is inlet temperature
/ IG=0: specify uniform INLET mass flux GIN (Record 11.1) (default
/ / / NH=0: no inlet enthalpy/temperature transient
/ / / / NG=0: no flow/DP transient
/ / / / / NQ=0: no heat flux transient
/ / / / / / NDPTBL=0: irrelevant with NG=0
/ / / / / / / NHX=0: no exit H/T transient
/ / / / / / / / PLENIN=1: inlet plenum
/ / / / / / / / / PLENEX=1: exit plenum
GROUP / / / / / / / / / / ODVMAX=0: no dryout
/ / / / / / / / / / / iterations
0 2 0 0 0 0 0 0 0 0 0

ecord 11.1: Operating/boundary conditions (6F10.0,2I5)
EXIT - exit (system) pressure (MPa)
HXIN - inlet temperature/enthalpy/quality (see IH) (degC, kJ/kg or 1)
IN - average inlet mass flux (Mg/ms2)
FLUX - average heat flux (MW/m2)
HXOUT - exit temperature/enthalpy/quality (see IH) (degC, kJ/kg or 1)
PS - channel pressure drop; any non-zero value forces a pressure-drop
boundary condition and over-rides the IG options (default: 0)
REFIN - reference subchannel connected to the inlet plenum (default: 1)
REFEX - reference subchannel connected to the exit plenum (default: 1)

PEXIT	THXIN	GIN	AFLUX	THXOUT	DPS	IREFIN	IREFEX
/	/	/	/	/	/	/	/
000	3400.00	0.0189939	0.0074148				

N and AFLUX indicate mdot=0.065kg/s, and Q=0.065MW

ecord 11.2: Inlet thermal BC distribution (include if IH=2, 3 or 5)
FLNLT(I) or TINLET(I) or XFLNLT(I), (I = 1, NCHANL) (6F10.0)
FLNLT - inlet flow specific enthalpy for each subchannel (kJ/kg)
INLET - inlet temperature for each subchannel (degC)
FLNLT - inlet flow equilibrium quality for each subchannel (1)
ample entry:
65.00 265.00 265.00 265.00 265.00 265.00

ecord 11.3: Inlet flow distribution (include if IG=2 or 12)
INLET(I), (I = 1, NCHANL) (12E5.0)
INLET - inlet flow fraction (F(I) / FTOTAL) for each subchannel

nominal PT case (using actual flow area fractions for nominal geometry)

```
.0084.0168.0168.0084.0257.0169.0514.0169.0514.0169.0257.0256
.0220.0282.0220.0512.0220.0282.0220.0512.0220.0282.0220.0256
.0242.0479.0467.0449.0427.0404.0383.0365.0354.0175
*------------------------------
*  Record 11.4:  Pressure transient table (include if NP>1)
*  YP(I), FP(I), (I = 1, NP)                          (12E5.0)
*  YP    - time to apply transient pressure factor FP
*  FP    - fraction of steady-state pressure PEXIT at time YP
* 0.0  1.0  0.4  1.1  0.5  1.0  1.0  1.0  1.0  0.0  2.0  0.0
*------------------------------
*  Record 11.5:  Thermal inlet BC transient table (include if NH>1)
*  YH(I), FH(I), (I = 1, NH)                          (12E5.0)
*  YH    - time to apply transient enthalpy/temperature factor FH
*  FH    - fraction of inlet enthalpy (IH=0 or 2), temperature (IH=1 or 3)
*          or quality (IH=4 or 5) at time YH
* 0.0  1.0  1.0  1.0
*------------------------------
*  Record 11.6:  Flow or pressure drop transient table (include if NG>1)
*  YG(I), FG(I), (I = 1, NG)                          (12E5.0)
*  YG    - time to apply flow/DP factor FG
*  FG    - fraction of steady-state mass flux GIN (DPS=0) or pressure drop DPS
*          (DPS>0, NDPTBL=1), or actual pressure drop (DPS>0, NDPTBL=0), at YG
* 0.0  1.0  0.2  0.8  0.4  0.4  0.6  0.3  0.8  0.3  1.0  0.4
*------------------------------
*  Record 11.7:  Heat flux transient table (include if NQ>1)
*  YQ(I), FQ(I), (I = 1, NQ)                          (12E5.0)
*  YQ    - time to apply transient heat flux factor FQ
*  FQ    - fraction of steady-state heat flux AFLUX at time YQ
* 0.0  1.0  0.4  1.2  1.0  1.2  1.1 0.14  2.0 0.06
*------------------------------
*  Record 11.8:  Thermal outlet BC transient table (include if NHX>1)
*  YHX(I), FHX(I), (I = 1, NHX)                       (12E5.0)
*  YHX   - time to apply transient enthalpy/temperature factor FHX
*  FHX   - fraction of exit enthalpy (IH=0 or 2), temperature (IH=1 or 3) or
*          quality (IH=4 or 5) at time YHX
* 0.0  1.0  1.0  1.0
*------------------------------
*  Record 11.9:  Onset-of-dryout iterations (include if ODVMAX>0)
*  ODVNMU, MCHFLO, MCHFUP, ODVTOL                     (*)
*  ODVNMU - name of BC parameter on which to iterate (case-insensitive string)
*     ='Heat Flux'      : iterate on average heat flux AFLUX
*     ='Inlet Temperature': iterate on inlet temperature THXIN
*     ='Inlet Mass Flux' : iterate on inlet mass flux GIN
*     ='Exit Pressure'   : iterate on exit pressure PEXIT
*  MCHFLO - lower bound for the target minimum CHF ratio (MCHFR)
*  MCHFUP - upper bound for the target minimum CHF ratio (MCHFR)
*  ODVTOL - relative convergence criteria for the iterate ODVNMU
*'Heat Flux'  0.9995  1.0005  5.E-4
*
*=============================================================================
*  Group 12:  Output Printout Options
*-----------------------------------------------------------------------------
*  NPAXL  - option to print subchannel data at any axial node(s):
*     = 0: print nodal subchannel data at the channel exit only
*     < 0: print subchannel data at all axial nodes
*     > 0: read NPAXL axial nodes to print (Record 12.1), plus the channel exit
*
*  NPCHAN - option for printing subchannel data:
*     = 0: print no subchannel data
*     < 0: print data for all subchannels
*     > 0: read NPCHAN subchannel numbers to print out (Record 12.2)
*
*  NPROD  - option for printing fuel rod heat fluxes and temperatures:
*     = 0: print no fuel rod data
*     < 0: print data for all rods
*     > 0: read NPROD rod numbers to print out (Record 12.3)
*
*  NPNODE - option for printing interior rod temperatures:
*     = 0: print rod centerline and rod cladding surface temperature
*  =3 to 7: NPNODE equally spaced interior rod temperatures are printed along
*           with the clad surface temperature
*
*  NPGAP  - option for printing cross-flows through gaps:
*     = 0: print no gap cross-flow data
*     < 0: print cross-flow data for all gaps
*     > 0: read NPGAP gap numbers at which to print cross-flows (Record 12.4)
*
*  NPLOT  - plotting option:
*     = 0: no plots
*     = 3: create REGIS files of section geometry at given nodes (Record 12.9)
*
*  NSCHF  - number of subsequent sorted CHF ratios to print:
```

= 0: print minimum CHF ratio data only
> 0: also print the next NSCHF smallest CHF ratios

NPAXL: number of axial nodes to print
/ NPCHAN: number of subchannels to print
/ / NPROD: number of rods to print
/ / / NPNODE: number of interior fuel nodes to print
/ / / / NPGAP: number of gap cross-flows to print
/ / / / / NPLOT=0: no cross-section plots
/ / / / / / NMARK, NPCRS: not used
NGROUP / / / / / / _____/ NSCHF: subsequent CHFRs
/ / / / / / / / / /
2 -1 -1 -1 0 -1 0 0 0 0

Record 12.1: Axial nodes to print (include if NPAXL>0)
PRINTX(I), (I = 1, NPAXL) (24I3)
PRINTX - axial node numbers at which to print subchannel data
2111117

Record 12.2: Subchannels to print (include if NPCHAN>0)
PRINTC(I), (I = 1, NPCHAN) (24I3)
PRINTC - subchannel numbers for which data is printed
31 62 63

Record 12.3: Rods to print (include if NPROD>0)
PRINTR(I), (I = 1, NPROD) (24I3)
PRINTR - rod numbers for which data is printed
30

Record 12.4: Gaps to print (include if NPGAP>0)
PRINTG(I), (I = 1, NPGAP) (24I3)
PRINTG - gap numbers for which data is printed
10 12 14 34 36 38 76 78

Record 12.5: Cross-sectional geometry plots (include if NPLOT=3) (*)
NPLOTS - number of nodes (Record 12.6) at which to plot geometry

---
Record 12.6: Cross-sections to plot (include if NPLOT=3 and NPLOTS>0)
PLTNOD(I), (I = 1, NPLOTS) (12I5)
PLTNOD - axial node numbers at which to plot cross-section geometry

---
GROUP 16: PRESSURE TUBE HEAT TRANSFER MODEL
---
geometry: pressure tube inner dia: 12.18 cm
pt thickness: 0.361 cm
ct thickness: 0.140 cm
PT and CT are both: Zr-2.5%Nb (material type 2)
Heat transfer coeff (gap+ht to moderator) = 1 kW/m2K
Moderator Temperature: 79.85 C

#laxiall # | rad | propsl#
wallslnodelsectl thetal templiterloutput
suming ballooned PT geometry
16 2 60 10 0 2 0 1 200
suming nominal PT geometry
6 3 60 10 0 2 0 1 200
n1 n2 n3 n4 n5 n6 n7 n8

TIDlMax error (PTID=press tube inner dia - [cm])
llooned
.18 .05
minal
84 .05
aterial Types (2=Zr-2.5Nb)
llooned
2 2
minal
5 2

mber of radial wall segments
llooned
3 3
minal
3 3

all thickness [cm]
llooned
61 .140
minal
.160 .140

```
*
*   Subchannel      moderator    htc
* Sect. |    angle temperature [kW/m2K]
* | |    |      |       |
*ballooned outer subchannel information
*   1  25   12.91   79.85     1.0
*   2  26   25.14   79.85     1.0
*   3  27   23.47   79.85     1.0
*   4  28   21.51   79.85     1.0
*   5  29   19.77   79.85     1.0
*   6  30   18.41   79.85     1.0
*   7  31   17.43   79.85     1.0
*   8  32   16.79   79.85     1.0
*   9  33   16.43   79.85     1.0
*  10  34    8.12   79.85     1.0
*-------------------------------------------------------
*nominal outer subchannel information
   1  25   10.15   79.85     1.0
   2  26   20.28   79.85     1.0
   3  27   20.22   79.85     1.0
   4  28   20.14   79.85     1.0
   5  29   20.05   79.85     1.0
   6  30   19.95   79.85     1.0
   7  31   19.85   79.85     1.0
   8  32   19.78   79.85     1.0
   9  33   19.73   79.85     1.0
  10  34    9.86   79.85     1.0
*-------------------------------------------------------
* End of Case
   0
* End of File
   0
```

Appendix B

New Model FORTRAN Code

```
      PROGRAM MAIN
      IMPLICIT NONE
      INTEGER ID,JD,KD,I,J,K,JB,JE,IB,IE,IDAT
      PARAMETER(ID=15,JD=16,KD=24,IDAT=10)
C-----------------------------------------------------------------
C ID is the array dimension for subchannel arrays, KD is the array
C dimension for gap arrays, and JD is the array dimension
C for all outer wall segment arrays, inclusive (PT, gap, and CT).
C therefore the following must be true:
C ID.GT.NUMV, JD.GT.(NUMW1+NUMW2+NUMW3+1), KD.GT.NUMG
C-----------------------------------------------------------------
      INTEGER NUMV,NUMG,IWG(KD),IEG(KD),IMAX,ITER,
     :        BUNDLE,BUNI,NUMITER,NUMW1,NUMW2,NUMW3,WALMAT(JD),
     :        GAPN(ID),GAPV(ID,ID),GN(ID,KD),OPCV(ID,ID),
     :        INNRIT,INNTOT,MIDIT,MIDTOT,IMODHL,PTFILM,SYMMET
      REAL HIN(ID),HINI,H(ID),HAVG(ID),HWRK(ID),HOCALC,HICALC,
     :     MTOT,MTOTO,FRAC(ID),MDOT(ID),PRESS,
     :     QLOSS(ID),QMODBUN,QMODCH,QCHAN,QBUN,QFUEL(ID),
     :     DHYD(ID),PWET(ID),AREAX(ID),DHYDTOT,PWETTOT,AREATOT,
     :     AREAG(KD),GAP(KD),LEN,RODDIA(KD),
     :     TW(JD,ID),HMOD,TMOD,ROT(JD),ANG(ID),WID(JD),
     :     BPO(ID),BP(ID),WPG(KD),DG(KD),APP(ID),DGN(ID,ID),BMASS(KD),
     :     ERR,HDIFFMAX,DIFF,X,XI,TCONST,PI,MULT,RELAX,
     :     HNMX(ID),HMIX(ID),HFCTR(ID),HFCTRI(ID),
C-----------------------------------------------------------------
C these are the unsaturated vapour and saturated (liquid or vapour)
C properties, respectively.the saturated liquid properties (if desired)
C may be used to calculate the heat transfer coefficient at the PT wall.
C This would be done if there were a liquid film predicted to exist at
C the PT wall. Otherwise the unsaturated vapour properties should be used
C-----------------------------------------------------------------
     :     TEMP(ID), TEMPK(ID), DEN(ID), CP(ID), VISC(ID), COND(ID),
     :     TEMPS(ID),TEMPKS(ID),DENS(ID),CPS(ID),VISCS(ID),CONDS(ID)
      CHARACTER HNAME*52
      PARAMETER(TCONST=273.15,PI=3.141593,RELAX=0.627,SYMMET=1)
C-----------------------------------------------------------------
C establishing a constant to account for the utilization
C of symmetry (to use symmetry, set SYMMET.eq.1)
C-----------------------------------------------------------------
      MULT=1.0
      IF (SYMMET.EQ.1) MULT=0.5
C-----------------------------------------------------------------
C open input and output files
C-----------------------------------------------------------------
      CALL FILE(IDAT)
C-----------------------------------------------------------------
C specifying the geometrical parameters
C-----------------------------------------------------------------
      CALL GEOM(DHYD,PWET,AREAG,AREAX,GAP,LEN,
     :     DHYDTOT,AREATOT,PWETTOT,MULT,GN,
     :     RODDIA,IWG,IEG,NUMG,NUMV,GAPN,GAPV,OPCV,ID,KD)
C-----------------------------------------------------------------
C reading in convergence parameter data, operating condition data,
C and moderator heat loss conditions and options
C-----------------------------------------------------------------
      CALL INITAL(HNAME,HIN,HNMX,H,HINI,BMASS, HICALC,QCHAN,
     :     MTOT,MTOTO,PRESS,XI,X,ERR,HDIFFMAX,HMOD,TMOD,MULT,
     :     IMODHL,PTFILM,IMAX,NUMITER,BUNI,NUMV,NUMG,
     :     NUMW1,NUMW2,NUMW3,IDAT,ID,KD)
C-----------------------------------------------------------------
C defining moderator heat loss geometry and parameters based on
C options read in from INITAL
C-----------------------------------------------------------------
      CALL GEOM2(WID,ROT,ANG,PI,MULT,
     :     NUMV,NUMW1,NUMW2,NUMW3,WALMAT,ID,JD,IB,IE,JB,JE)
C-----------------------------------------------------------------
C writing out the channel conditions to the enthalpy output files:
C-----------------------------------------------------------------
      CALL HEADER(MTOTO,QCHAN,PRESS,SYMMET)
C-----------------------------------------------------------------
C writing out the inlet enthalpy distribution for the first bundle:
C-----------------------------------------------------------------
      write(12,1001) x,(HIN(I),I=1,NUMV),HICALC
1001  FORMAT(F6.1,1X,15(F8.1))
C-----------------------------------------------------------------
C calculate the flow fractions of each CV based on correlations
C derived from ASSERT results
C-----------------------------------------------------------------
      CALL FLOWSPLIT(FRAC,MTOTO,PRESS,NUMV,ID)
C-----------------------------------------------------------------
C calculate CV mass flow
C-----------------------------------------------------------------
      DO 10 I=1,NUMV
         MDOT(I)=FRAC(I)*MTOT
```

```
 10   CONTINUE
C-----------------------------------------------------------------------
C  make initial guess at all wall temperatures, (helps convergence
C  for middle loop of the first bundle).  The initial guess for wall
C  temperature for subsequent bundles is equal to the wall temperature
C  solution from the previous bundle
C-----------------------------------------------------------------------
      CALL WTIG(TW,TCONST,NUMW1,NUMW2,NUMW3,JD,JB,JE,ID,IB,IE)
C-----------------------------------------------------------------------
C  calculate the constant that governs the degree of buoyancy induced
C  mixing.  The constant is a function of pressure, mass flow,
C  channel power and inlet enthalpy
C-----------------------------------------------------------------------
      CALL BUOYMASS(BMASS,PRESS,MTOT,QCHAN,HINI,MULT,NUMG,KD)
C-----------------------------------------------------------------------
C  the code contains 3 nested loops, the outer loop is the bundle loop,
C  the middle loop is the updating property and wall temperature loop,
C  and the inner loop is the Point Gauss Seidel (PGS) solver.
C-----------------------------------------------------------------------
C  starting the outer loop for calculations for each bundle
C-----------------------------------------------------------------------
      DO 200 BUNDLE=BUNI,12
C-----------------------------------------------------------------------
C  calculating the bundle fuel power, CV fuel power, total CV energy
C  source and properties for the fluid in each control volume:
C  (note that the energy source (BPO) does not yet include heat losses
C  to the moderator-these will be subtracted in an inner property loop)
C-----------------------------------------------------------------------
       CALL QBUNDLE(QBUN,QCHAN,BUNDLE)
       CALL QCV(QFUEL,QBUN,BUNDLE,NUMV,ID)
       CALL SOURCE(BPO, QFUEL,MDOT,HIN,NUMV,ID)
       CALL PROPCV(PRESS,HIN,
     :      TEMP, TEMPK, DEN, CP, VISC, COND,TCONST,
     :      TEMPS,TEMPKS,DENS,CPS,VISCS,CONDS,NUMV,ID)
C-----------------------------------------------------------------------
C  middle loop to check on effect of updating flow properties and
C  updating wall temperatures:(MIDIT is middle loop iteration counter)
C-----------------------------------------------------------------------
       MIDIT=0
       DO 100 ITER=1,NUMITER
        MIDIT=MIDIT+1
C-----------------------------------------------------------------------
C  setting the value for HWRK to H (used for relaxation)
C-----------------------------------------------------------------------
        CALL HRESET(HWRK,H,NUMV,ID)
C-----------------------------------------------------------------------
C  calculating the thermal mixing coefficient due to turbulence
C  for each gap:
C-----------------------------------------------------------------------
        CALL WTURB(WPG,MDOT,VISC,DHYD,AREAX,RODDIA,GAP,PWET,DEN,
     :       IEG,IWG,NUMG,ID,KD)
C-----------------------------------------------------------------------
C  calculating the diffusion coefficient for turbulent and
C  crossflow mixing
C-----------------------------------------------------------------------
        CALL DIFPHI(DG, WPG,AREAG,GAP,MDOT,LEN,IWG,IEG,NUMG,ID,KD)
C-----------------------------------------------------------------------
C  calculating the lateral mass flow rate for buoyancy cross flow and
C  adding it to lateral mass flow rate due to turbulent diffusion.
C-----------------------------------------------------------------------
        CALL BUOYFLO(DG,DGN,BMASS,QCHAN,MDOT,AREAX,DEN,LEN,
     :       NUMG,NUMV,GAPN,GAPV,OPCV,GN,IWG,IEG,ID,KD)
C-----------------------------------------------------------------------
C  calculating the APP (kg/s) coefficient:
C-----------------------------------------------------------------------
        CALL COEFF(APP, DGN,MDOT,IEG,IWG,NUMG,NUMV,GAPN,ID,KD)
C-----------------------------------------------------------------------
C  if IMODHL=1, then heat loss to moderator will be considered.
C  wall heat loss calculated assuming only radial conduction
C-----------------------------------------------------------------------
C  ASSERT uses unsaturated liquid properties to calculate
C  the heat loss to the moderator (to simulate a liquid film on
C  the PT wall).  The user may want to use unsaturated vapour
C  properties to calculate moderator heat loss.
C  To use saturated liquid properties,let   PTFILM=1,
C  to use unsaturated vapour properties,let PTFILM=0 (or anything else).
C  the value CON=1.44 is used only for ASSERT comparison reasons;
C  it seems to compensate for the difference between using unsaturated
C  liquid properties (as ASSERT does), and saturated liquid properties.
C-----------------------------------------------------------------------
        IF (IMODHL.EQ.1) THEN
         IF (PTFILM.EQ.1) THEN
          CALL WALLHT(QLOSS,TMOD,HMOD,TW,TCONST,TEMPKS,MDOT,
     :         CONDS,CPS,VISCS,ROT,LEN,AREAX,DHYD,ANG,WID,
```

```
     :          1.44,WALMAT,ID,JD,KD,JB,JE,IB,IE)
          ELSE
           CALL WALLHT(QLOSS,TMOD,HMOD,TW,TCONST,TEMPK,MDOT,
     :          COND,CP,VISC,ROT,LEN,AREAX,DHYD,ANG,WID,
     :          1.00,WALMAT,ID,JD,KD,JB,JE,IB,IE)
          END IF
         END IF
-----------------------------------------------------------------
 calculating new heat source terms (BP in kW) that include
 the wall heat losses (QLOSS in W)
-----------------------------------------------------------------
         DO I=1,NUMV
          BP(I)=BPO(I)-0.001*QLOSS(I)
         ENDDO
-----------------------------------------------------------------
 inner loop - running PGS algorithm to get enthalpy convergence
-----------------------------------------------------------------
         CALL PGS(H,ERR,APP,BP,DGN,OPCV,
     :        IWG,IEG,NUMV,NUMG,IMAX,GAPV,GAPN,INNRIT,INNTOT,ID,KD)
-----------------------------------------------------------------
 applying relaxation--good value seems to be RELAX=0.63
-----------------------------------------------------------------
         DO 20 K=1,NUMV
          HAVG(K)=RELAX*H(K)+(1-RELAX)*HWRK(K)
         CONTINUE
-----------------------------------------------------------------
 updating fluid properties
-----------------------------------------------------------------
         CALL PROPCV(PRESS,HAVG,
     :        TEMP, TEMPK, DEN, CP, VISC, COND,TCONST,
     :        TEMPS,TEMPKS,DENS,CPS,VISCS,CONDS,NUMV,ID)
-----------------------------------------------------------------
 calculating the average enthalpy entering and leaving the bundle:
-----------------------------------------------------------------
         CALL AVG_H(HOCALC, H,MDOT,MTOT,NUMV,ID)
         CALL AVG_H(HICALC, HIN,MDOT,MTOT,NUMV,ID)
-----------------------------------------------------------------
 calculating the difference in the enthalpies
 between the current iteration and the previous:
-----------------------------------------------------------------
         CALL HDIFF(DIFF, H,HWRK,NUMV,ID)
-----------------------------------------------------------------
 exiting the middle loop if the property loop has converged:
-----------------------------------------------------------------
         IF (DIFF.LT.HDIFFMAX) GOTO 110
 0     CONTINUE
 0     CONTINUE
       MIDTOT=MIDTOT+MIDIT
       IF (ITER.GE.NUMITER) WRITE(12,*) 'PROPERTY LOOP CONV. PROBLEM'
       IF (ITER.GE.NUMITER) WRITE(6,*)  'PROPERTY LOOP CONV. PROBLEM'
-----------------------------------------------------------------
 alc the enthalpy using FACTAR Partial Mixing methodology, and also
 or mixing models using no mixing and mixing at endplates:
-----------------------------------------------------------------
       CALL FACTAR(HFCTR,HFCTRI, MDOT,QFUEL,HIN,BUNDLE,BUNI,ID)
       CALL ANALYTICAL(HNMX,HMIX, HICALC,MDOT,QFUEL,NUMV,ID)
-----------------------------------------------------------------
 etting the enthalpy calculated for this bundle to the inlet
 nthalpy for the next bundle, also calculating new axial position:
-----------------------------------------------------------------
       CALL HRESET(HIN, H,NUMV,ID)
       X=X+LEN*100.0
-----------------------------------------------------------------
 alculations for current bundle are complete. Writing out output:
-----------------------------------------------------------------
       CALL OUTDAT(APP,BP,MDOT,QFUEL,WPG,
     :      DIFF,AREAG,DG,X,H,HMIX,HNMX,HOCALC,HFCTR,
     :      BUNDLE,ITER,NUMG,NUMV,ID,KD)
-----------------------------------------------------------------
 alc total mod heat losses and exit the outer loop (the bundle loop)
-----------------------------------------------------------------
       CALL MODHL(QMODBUN,QMODCH,QLOSS,IB,IE,ID)
      CONTINUE
-----------------------------------------------------------------
 ite out total heat loss to moderator over entire channel length.
 so write total iterations needed for convergence
-----------------------------------------------------------------
      WRITE(12,30) QMODCH
      FORMAT(' QMODCH= ',F12.6)
      WRITE(12,*)'INNTOT=',INNTOT
      WRITE(12,*)'MIDTOT=',MIDTOT
-----------------------------------------------------------------
      STOP
      END
```

```
      SUBROUTINE GEOM(DHYD,PWET,AREAG,AREAX,GAP,LEN,
     :               DHYDTOT,AREATOT,PWETTOT,MULT,GN,
     :               RODDIA,IWG,IEG,NUMG,NUMV,GAPN,GAPV,OPCV,ID,KD)
      IMPLICIT NONE
      INTEGER ID,KD,I,K,IWG(KD),IEG(KD),IW,IE,NUMG,NUMV,
     :        GAPN(ID),GAPV(ID,ID),OPCV(ID,ID),GN(ID,KD)
      REAL DHYD(ID),AREAG(KD),AREAX(ID),GAP(KD),PWET(ID),LEN,
     :     DHYDTOT,AREATOT,PWETTOT,RODDIA(KD),MULT
-------------------------------------------------------------------
  This subroutine sets up the geometry for the 'FACTAR'
  control volumes

   DHYD(I) = hydraulic diameter for cv I [m]
   PWET(I) = wetted perimeter for cv I [m]
   AREAG(I) = flow area for gap I [m2]
   AREAX(I) = cross-sectional flow area for control volume I [m2]
   GAP(I) = width of a *single* ASSERT gap for gap number I [m]
   LEN = length of FACTAR control volume = 1 bundle length [m]
   AREATOT = bundle cross-sectional flow area [m2]
   PWETTOT = bundle wetted perimeter [m]
   DHYDTOT = bundle hydraulic diameter [m]
   IWG(I) = control volume number west of gap I
   IEG(I) = control volume number east of gap I
   NUMG = total number of gaps (16)
   NUMV = total number of control volumes (11)
   ID = array dimension (>16)
   RODDIA(K)=average diameter of fuel rod at gap K (m)
   GAPN(I)=the number of gaps surrounding CV I.
   GN(I,J)=the number of the gap between CV I and J.
   GAPV(I,J)=the value of the gap CV I at gap number J
   OPCV(I,J)=the CV opposite to CV I at gap number J
-------------------------------------------------------------------
defining the number of gaps, control volumes and the cv length:
-------------------------------------------------------------------
   NUMG=23
   NUMV=14
   LEN=0.4953
-------------------------------------------------------------------
assigning the wetted perimiter for each control volume:
-------------------------------------------------------------------
   PWET(1)=6.18E-02*MULT
   PWET(2)=6.18E-02*MULT
   PWET(3)=1.65E-01*MULT
   PWET(4)=1.24E-01*MULT
   PWET(5)=8.24E-02*MULT
   PWET(6)=2.89E-01*MULT
   PWET(7)=4.12E-02*MULT
   PWET(8)=1.24E-01*MULT
   PWET(9)=8.24E-02*MULT
   PWET(10)=8.24E-02*MULT
   PWET(11)=3.70E-01*MULT
   PWET(12)=1.63E-01*MULT
   PWET(13)=1.63E-01*MULT
   PWET(14)=4.07E-02*MULT
-------------------------------------------------------------------
assigning the flow area for each control volume:
-------------------------------------------------------------------
   AREAX(1)=  8.56200E-05*MULT
   AREAX(2)=  8.56200E-05*MULT
   AREAX(3)=  3.18700E-04*MULT
   AREAX(4)=  2.31600E-04*MULT
   AREAX(5)=  1.44500E-04*MULT
   AREAX(6)=  5.79740E-04*MULT
   AREAX(7)=  9.56000E-05*MULT
   AREAX(8)=  2.48140E-04*MULT
   AREAX(9)=  1.70220E-04*MULT
   AREAX(10)= 1.61380E-04*MULT
   AREAX(11)= 7.00460E-04*MULT
   AREAX(12)= 2.66960E-04*MULT
   AREAX(13)= 2.43920E-04*MULT
   AREAX(14)= 5.93600E-05*MULT
-------------------------------------------------------------------
assigning the hydraulic diameter for each control volume:
-------------------------------------------------------------------
   DO I=1,NUMV
    DHYD(I)=4.0*AREAX(I)/PWET(I)
   ENDDO
-------------------------------------------------------------------
calculating the total bundle cross-sectional area, wetted
perimeter and hydraulic diameter
-------------------------------------------------------------------
   AREATOT=0.0
   PWETTOT=0.0
   DO 10 I=1,NUMV
```

```
        AREATOT=AREATOT+AREAX(I)
        PWETTOT=PWETTOT+PWET(I)
 10    CONTINUE
       DHYDTOT=(4.0*AREATOT/PWETTOT)
C-----------------------------------------------------------------------
C  assigning the width of individual ASSERT gaps:
C-----------------------------------------------------------------------
       DO K=1,3
        GAP(K)=0.00178
       ENDDO
       DO K=4,8
        GAP(K)=0.001788
       ENDDO
       DO K=9,13
        GAP(K)=0.001918
       ENDDO
       GAP(14)=0.00178
       GAP(15)=0.001795
       GAP(16)=0.001795
       GAP(17)=0.004062
       GAP(18)=0.004062
       GAP(19)=0.001704
       GAP(20)=0.004062
       GAP(21)=0.001826
       GAP(22)=0.001423
       GAP(23)=0.001209
C-----------------------------------------------------------------------
C  define average rod diameter for each gap
C-----------------------------------------------------------------------
       DO K=1,20
        RODDIA(K)=0.01312
       ENDDO
C-----------------------------------------------------------------------
C  for gaps next to the PT wall, the average rod diamter is half of
C  the single rod it is adjacent to.
C-----------------------------------------------------------------------
       DO K=21,23
        RODDIA(K)=0.5*0.01312
       ENDDO
C-----------------------------------------------------------------------
C  assigning the gap areas for FACTAR gaps:
C     AREAG = (number of ASSERT gaps)*(ASSERT gap width)*(length)
C-----------------------------------------------------------------------
       AREAG(1)= 3.0*GAP(1)*LEN*MULT
       AREAG(2)= 2.0*GAP(2)*LEN*MULT
       AREAG(3)= 1.0*GAP(3)*LEN*MULT
       AREAG(4)= 5.0*GAP(4)*LEN*MULT
       AREAG(5)= 2.0*GAP(5)*LEN*MULT
       AREAG(6)= 2.0*GAP(6)*LEN*MULT
       AREAG(7)= 2.0*GAP(7)*LEN*MULT
       AREAG(8)= 1.0*GAP(8)*LEN*MULT
       AREAG(9)= 9.0*GAP(9)*LEN*MULT
       AREAG(10)=4.0*GAP(10)*LEN*MULT
       AREAG(11)=2.0*GAP(11)*LEN*MULT
       AREAG(12)=2.0*GAP(12)*LEN*MULT
       AREAG(13)=1.0*GAP(13)*LEN*MULT
       DO K=14,NUMG
        AREAG(K)=2.0*GAP(K)*LEN*MULT
       ENDDO
C-----------------------------------------------------------------------
C  assigning pointers for the gap #s:  IWG(GAP) and IEG(GAP) point
C  to the two CVs on either side of 'GAP'
C  note that the west CVs are always vertically higher than the east
C  this is useful for buoyancy calculations later in the code.
C-----------------------------------------------------------------------
       IWG(1)=1
       IWG(2)=2
       IWG(3)=2
       IWG(4)=3
       IWG(5)=4
       IWG(6)=4
       IWG(7)=5
       IWG(8)=5
       IWG(9)=6
       IWG(10)=8
       IWG(11)=9
       IWG(12)=10
       IWG(13)=10
       IWG(14)=1
       IWG(15)=3
       IWG(16)=4
       IWG(17)=6
       IWG(18)=7
       IWG(19)=8
```

```
      IWG(20)=9
      IWG(21)=11
      IWG(22)=12
      IWG(23)=13

      IEG(1)=3
      IEG(2)=4
      IEG(3)=5
      IEG(4)=6
      IEG(5)=7
      IEG(6)=8
      IEG(7)=9
      IEG(8)=10
      IEG(9)=11
      IEG(10)=12
      IEG(11)=13
      IEG(12)=13
      IEG(13)=14
      IEG(14)=2
      IEG(15)=4
      IEG(16)=5
      IEG(17)=7
      IEG(18)=8
      IEG(19)=9
      IEG(20)=10
      IEG(21)=12
      IEG(22)=13
      IEG(23)=14
C---------------------------------------------------------------------
C now assigning pointers in the other direction.
C For instance: GAPV(SC,GAPN) points to a gap adjacent to subchannel
C 'SC', while GAPN defines how many gaps are adjacent to 'SC'.
C For this geometry, GAPN is always less than or equal to 5 because
C no subchannel other than 4 has as many as 5 adjacent gaps.
C GN(I,J) points to the number (not the value) of the gap that is
C adjacent to CV I, at GAP J
C---------------------------------------------------------------------
C first initialize to zero
C---------------------------------------------------------------------
      DO I=1,NUMV
       GAPN(I)=0
       DO K=1,NUMV
        GAPV(I,K)=0
       ENDDO
       DO K=1,NUMG
        GN(I,K)=0
       ENDDO
      ENDDO

      DO I=1,NUMV
       DO K=1,NUMG
        IF ((IWG(K).EQ.I) .OR. (IEG(K).EQ.I)) THEN
         GAPN(I)=GAPN(I)+1
         GN(I,K)=GAPN(I)
         GAPV(I,GAPN(I))=K
        END IF
       ENDDO
      ENDDO
C---------------------------------------------------------------------
C now assigning a pointer for opposite CVs, for instance, OPCV(I,K)
C points to the CV that is opposite CV I, at GAP K
C---------------------------------------------------------------------
C first initialize to zero
C---------------------------------------------------------------------
      DO I=1,NUMV
       DO K=1,NUMV
        OPCV(I,K)=0
       ENDDO
      ENDDO

      DO I=1,NUMV
       DO K=1,GAPN(I)
        IF (IWG(GAPV(I,K)).EQ.I) THEN
         OPCV(I,K)=IEG(GAPV(I,K))
        ELSE IF (IEG(GAPV(I,K)).EQ.I) THEN
         OPCV(I,K)=IWG(GAPV(I,K))
        END IF
       ENDDO
      ENDDO

      RETURN
      END

C---------------------------------------------------------------------
```

```
      SUBROUTINE GEOM2(WID,ROT,ANG,PI,MULT,
     :           NUMV,NUMW1,NUMW2,NUMW3,WALMAT,ID,JD,IB,IE,JB,JE)
      IMPLICIT NONE
      INTEGER I,J,JB,JE,IB,IE,ID,JD
      INTEGER NUMV,NRW,NUMW1,NUMW2,NUMW3,WALMAT(JD)
      REAL WID(JD),RWALLI,RWALLO,ROT(JD),
     :     ANG(ID),SECT(ID),PI,PTWID,GAPWID,CTWID,MULT
C-----------------------------------------------------------------------
C  this subroutine calculates the geometry for the moderator heat
C  loss routine
C  NRW = the total number of wall segments being modelled
C  JB-1 = the face of the inside wall, JE+1 is the face of the outside wall
C  JB = the first normal wall segement, JE is the last normal wall segment
C  IB = first outer subchannel, IE is the last outer subchannel
C     RWALLI=inner radius of PT (m)
C     PTWID=width of pressure tube (m)
C     GAPWID=width of CO2 gap (m)
C     CTWID=width of CT (m)
C     RWALLO=outer radius of CT (m)
C     WID(J)=width of segment J (m)
C     WALMAT(J)=type of wall material at segment J.
C     ROT(J)=outer radius of segment J (m)
C     SECT(I)=angle (in degrees) subtended by subchannel I on the PT
C     ANG(I)=angle (in radians) subtended by subchannel I on the PT
C-----------------------------------------------------------------------
      NRW=NUMW1+NUMW2+NUMW3
      JB=2
      JE=NRW+JB-1
      IB=11
      IE=NUMV
C-----------------------------------------------------------------------
C  wall segment width info
C-----------------------------------------------------------------------
      RWALLI=0.0517
      PTWID =0.00361
      GAPWID=0.0016
      CTWID =0.0014
      RWALLO=RWALLI+PTWID+GAPWID+CTWID
      WID(JB-1)=0.0
      WID(JE+1)=0.0
      DO J=JB,JE
       IF (J.GE.JB)              WID(J)=PTWID/(NUMW1)
       IF (J.GE.JB+NUMW1)        WID(J)=GAPWID/(NUMW2)
       IF (J.GE.JB+NUMW1+NUMW2) WID(J)=CTWID/(NUMW3)
      ENDDO
C-----------------------------------------------------------------------
C  define wall material types
C  WALMAT=1 indicates Zirc + 2.5%Nb
C  WALMAT=2 indicates CO2
C  WALMAT=3 indicates a wall of zero thickness
C-----------------------------------------------------------------------
      WALMAT(JB-1)=3
      WALMAT(JE+1)=3
      DO J=JB,JE
       IF (J.GE.JB)              WALMAT(J)=1
       IF (J.GE.JB+NUMW1)        WALMAT(J)=2
       IF (J.GE.JB+NUMW1+NUMW2) WALMAT(J)=1
      ENDDO
C-----------------------------------------------------------------------
C  define outer wall radii
C-----------------------------------------------------------------------
      ROT(JB-1)=RWALLI
      DO J=JB,JE+1
       ROT(J)=ROT(J-1)+WID(J)
      ENDDO
C-----------------------------------------------------------------------
C  these are the angles that the outer subchannels
C  subtend on the pressure tube
C-----------------------------------------------------------------------
      SECT(11)=181.656*MULT
      SECT(12)= 79.596*MULT
      SECT(13)= 79.020*MULT
      SECT(14)= 19.728*MULT
      DO I=IB,IE
       ANG(I)=SECT(I)*PI/180.0
      ENDDO
C
      RETURN
      END
```

```
      SUBROUTINE FILE(IDAT)
      INTEGER IDAT
C-----------------------------------------------------------------------
C opening the files for i/o:
C-----------------------------------------------------------------------
      OPEN(IDAT,FILE='input.dat')
      OPEN(11,FILE='output.out')
      OPEN(14,FILE='hmix.out')
      OPEN(15,FILE='h_nomix.out')
      OPEN(16,FILE='h_factar.out')
      OPEN(18,FILE='a.dat')

      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE HEADER(MTOT,QCHAN,PRESS,SYMMET)
      IMPLICIT NONE
      INTEGER SYMMET
      REAL MTOT,QCHAN
      REAL PRESS
C-----------------------------------------------------------------------
      This subroutine writes out the header for the enthalpy
      output file.The header contains information on the channel
      conditions:  Header is written to unit 12.

             MTOT = total mass flow [kg/s]
             QCHAN = total channel power [MW]
             PRESS = outlet pressure [MPa]
C-----------------------------------------------------------------------
      write(12,2000) MTOT,QCHAN,PRESS
      write(12,2001)
      write(14,2000) MTOT,QCHAN,PRESS
      write(14,2001)
      write(15,2000) MTOT,QCHAN,PRESS
      write(15,2001)

2000 FORMAT('# Predicted Enthalpies [kJ/kg]: mdot=',F8.4,
     :  '[kg/s]; Q=',F8.2,'[MW]; P=',F8.4,' [MPa]')
2001 FORMAT('# x[cm]  h-cv1   h-cv2   h-cv3   h-cv4   h-cv5   h-cv6
     :h-cv7   h-cv8   h-cv9   h-cv10  h-cv11  h-cv12  h-cv13  h-cv14
     :bndl avg')

      write(16,*) ' x[cm]    h-inner     h-outer  [kJ/kg]'
      RETURN
      END
```

```
      SUBROUTINE INITAL(HNAME,HIN,HNMX,H,HINI,BMASS, HICALC,QCHAN,
     :           MTOT,MTOTO,PRESS,XI,X,ERR,HDIFFMAX,HMOD,TMOD,MULT,
     :           IMODHL,PTFILM,IMAX,NUMITER,BUNI,NUMV,NUMG,
     :           NUMW1,NUMW2,NUMW3,IDAT,ID,KD)
      IMPLICIT NONE
      INTEGER ID,KD,I,K,IDAT,NUMV,NUMG,IMAX,BUNI,NUMITER,
     :        NUMW1,NUMW2,NUMW3,UNICON,IMODHL,PTFILM
      REAL HICALC,QCHAN,MTOT,XI,HUNI,HVARI(ID),HVAVG,
     :     X,ERR,HNMX(ID),H(ID),HDIFFMAX,HMOD,TMOD,MULT,HIN(ID)
      REAL PRESS,BMASS(KD),HINI,MTOTO
      CHARACTER PCH*4,HCH*6,MCH*5,QCH*6,FNAME*52,HNAME*52
C-----------------------------------------------------------------------
      ERR=maximum error for PGS loop
      IMAX=maximum number of iterations for PGS loop
      HDIFFMAX=maximum error for property loop
      NUMITER=maximum number of iterations for property loop
      BUNI=1st bundle to be modelled
      PRESS=channel pressure (MPa)
      HUNI=uniform inlet enthalpy (kJ/kg)
      HINI=average inlet enthalpy (kJ/kg)
      XI=axial distance of first bundle(m)
      HVARI(I)=(non-uniform) inlet enthalpy of CV I (kK/kg)
      MTOTO=channel mass flow rate (kg/s)
      MTOT=channel mass flow following symmetry calcs.
      QCHAN=channel power(MW)
      UNICON=option for uniform/non-uniform inlet enthalpy
      IMODHL=option for using moderator heat loss
      PTFILM=option for using saturated liquid properties or
          non-saturated vapour properties for moderator heat
          loss calcs.
      HMOD=convection coeeficient of moderator (W/m^2K)
      TMOD=temperature of moderator (K)
      NUMW1=number of wall segments for PT
      NUMW2=number of wall segments for CO2 gap.
      NUMW2=number of wall segments for CT
C-----------------------------------------------------------------------
C reading in the data from input.dat.  this data is assumed to be
C for a full bundle (not a half bundle).  NOTE: HUNI and HVARI
C both specify the inlet enthalpy, which one is used is dependant
C on the value that UNICON is assigned.  If UNICON=0, then HVARI is
C used, otherwise, HUNI is used.
C-----------------------------------------------------------------------
      READ(18,4) (BMASS(K),K=1,NUMG)
      FORMAT(E12.5)
      READ(IDAT,5) ERR,IMAX,HDIFFMAX,NUMITER,BUNI
      READ(IDAT,*) PRESS
      READ(IDAT,*) HUNI
      READ(IDAT,*) XI,(HVARI(I),I=1,NUMV),HVAVG
      READ(IDAT,*) MTOTO
      READ(IDAT,*) QCHAN
      READ(IDAT,6) UNICON
      READ(IDAT,7) IMODHL,PTFILM
      READ(IDAT,*) HMOD,TMOD
      READ(IDAT,*) NUMW1,NUMW2,NUMW3
      FORMAT(/E7.1,I5,1X,E7.1,I5,I2/)
      FORMAT(I4/)
      FORMAT(2(I2))
C-----------------------------------------------------------------------
C recalculates channel conditions based on the half bundle
C (symmetry) option.
C-----------------------------------------------------------------------
      QCHAN=QCHAN*MULT*1.0432443
      QCHAN=QCHAN*MULT
      MTOT =MTOTO*MULT
C-----------------------------------------------------------------------
C closing the data file so that it may be reopened and read with
C the operating conditions as character type data.  The operating
C conditions are then converted to file names for the output data.
C-----------------------------------------------------------------------
      CLOSE(IDAT)
      OPEN(IDAT,FILE='input.dat')
      READ(IDAT,8) PCH,HCH,MCH,QCH
      FORMAT(///A4/A6//A5/A6)
C-----------------------------------------------------------------------
C FNAME is for the value check file name, HNAME is for
C the enthalpy output file name.
C-----------------------------------------------------------------------
      FNAME=PCH//'MPa_'//HCH//'kJpkg_'//
     :      MCH//'kgps_'//QCH//'MW_'//'check.out'
      HNAME=PCH//'MPa_'//HCH//'kJpkg_'//
     :      MCH//'kgps_'//QCH//'MW_'//'enthalpy.out'
      OPEN(17,FILE=FNAME)
      OPEN(12,FILE=HNAME)
C-----------------------------------------------------------------------
```

```
C  the following allows the user to choose between specifying a uniform
C  channel inlet enthalpy distribution or a variable channel inlet
C  enthalpy distribution. if UNICON=1: uniform inlet enthalpy dist'n
C-----------------------------------------------------------------------
      IF (UNICON.EQ.0) THEN
        DO I=1,NUMV
          HIN(I)=HVARI(I)
        ENDDO
        HICALC=HVAVG
        X=XI
      ELSE
        DO I=1,NUMV
          HIN(I)=HUNI
        ENDDO
        HICALC=HUNI
        X=0.0
      END IF
      HINI=HICALC
C-----------------------------------------------------------------------
C  setting the array HNMX(I) equal to the inlet enthalpy:
C-----------------------------------------------------------------------
      DO I=1,NUMV
       HNMX(I)=HIN(I)
      ENDDO
C-----------------------------------------------------------------------
C  guessing a value for H(I) (required for buoyancy calculation):
C-----------------------------------------------------------------------
      DO I=1,NUMV
       H(I)=HIN(I)
      ENDDO
C
      RETURN
      END
```

```
      SUBROUTINE FLOWSPLIT(FRAC, MTOT,PRESS,NUMV,ID)
      IMPLICIT NONE
      INTEGER ID,NUMV
      REAL FRAC(15),MTOT
      REAL PRESS
C-----------------------------------------------------------------------
C     This subroutine calculates the fraction of the mass flow
C     through each control volume.  It is based on the correlations
C     developed from a multitude of ASSERT-PV simulations at a
C     range of pressures, mass flows, fuel powers, and inlet
C     enthalpy.
C
C          MTOT = total mass flow through channel [kg/s]
C          PRESS = channel pressure [MPa]
C          FRAC(I) = fraction of total mass flow in cv I [-]
C-----------------------------------------------------------------------
      CALL CV1(FRAC(1),MTOT,PRESS)
      CALL CV2(FRAC(2),MTOT,PRESS)
      CALL CV3(FRAC(3),MTOT,PRESS)
      CALL CV4(FRAC(4),MTOT,PRESS)
      CALL CV5(FRAC(5),MTOT,PRESS)
      CALL CV6(FRAC(6),MTOT,PRESS)
      CALL CV7(FRAC(7),MTOT,PRESS)
      CALL CV8(FRAC(8),MTOT,PRESS)
      CALL CV9(FRAC(9),MTOT,PRESS)
      CALL CV10(FRAC(10),MTOT,PRESS)
      CALL CV12(FRAC(12),MTOT,PRESS)
      CALL CV13(FRAC(13),MTOT,PRESS)
      CALL CV14(FRAC(14),MTOT,PRESS)
      CALL SUMCV(FRAC(11),11,FRAC)
      RETURN
      END

C=======================================================================
      SUBROUTINE CV1(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C-setting array values (determined by least squares fit to ASSERT
C output)
C-----------------------------------------------------------------------
      A(1)=  -.0301
      A(2)=   .0414
      A(3)=   .4599
      A(4)=   .6278
      A(5)=   .5056
      A(6)=   .2677
      B(1)=   .0361
      B(2)=   .0401
      B(3)=  -.0006
      B(4)=  -.0180
      B(5)=  -.0036
      B(6)=   .0219
      C(1)=   .0131
      C(2)=   .0131
      C(3)=   .0143
      C(4)=   .0151
      C(5)=   .0152
      C(6)=   .0151
      D=   .0001
      E=   .0202
      MINT(1)=  .24129
      MINT(2)=  .14930
      MINT(3)=  .11180
      MINT(4)=  .10360
      MINT(5)=  .10076
      MINT(6)=  .10004
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
      RETURN
      END

C=======================================================================
      SUBROUTINE CV2(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--setting array values (determined by least squares fit to ASSERT
C output)
C-----------------------------------------------------------------------
```

```
      A(1)=  -.0473
      A(2)=   .1124
      A(3)=   .1415
      A(4)=   .0969
      A(5)=  -.0503
      A(6)=   .0240
      B(1)=   .0458
      B(2)=   .0284
      B(3)=   .0414
      B(4)=   .0484
      B(5)=   .0644
      B(6)=   .0491
      C(1)=   .0132
      C(2)=   .0147
      C(3)=   .0151
      C(4)=   .0157
      C(5)=   .0158
      C(6)=   .0168
      D=    .0005
      E=    .0221
      MINT(1)=  .23618
      MINT(2)=  .14419
      MINT(3)=  .10449
      MINT(4)=  .09092
      MINT(5)=  .08414
      MINT(6)=  .07905
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
      RETURN
      END
C

C=======================================================================
      SUBROUTINE CV3(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*mtot^2 + B*mtot + C
C
C--setting array values (determined by least squares fit to ASSERT
C  output)
C-----------------------------------------------------------------------
      A(1)=  -.1748
      A(2)=   .1597
      A(3)=   .3897
      A(4)=   .4522
      A(5)=   .4067
      A(6)=   .2997
      B(1)=  -.0141
      B(2)=  -.0470
      B(3)=  -.0748
      B(4)=  -.0817
      B(5)=  -.0723
      B(6)=  -.0503
      C(1)=   .1007
      C(2)=   .0980
      C(3)=   .0969
      C(4)=   .0962
      C(5)=   .0954
      C(6)=   .0942
      D=   -.0004
      E=    .0939
      MINT(1)=  .14991
      MINT(2)=  .16834
      MINT(3)=  .15455
      MINT(4)=  .15892
      MINT(5)=  .16621
      MINT(6)=  .17539
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CV4(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL D2(10),E2(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*mtot^2 + B*mtot + C
C
```

```
-setting array values (determined by least squares fit to ASSERT
 output)
------------------------------------------------------------------
    D2(1)=  -.0021
    D2(2)=  -.0021
    D2(3)=  -.0021
    D2(4)=  -.0021
    D2(5)=  -.0021
    D2(6)=  -.0021
    E2(1)=   .0604
    E2(2)=   .0604
    E2(3)=   .0604
    E2(4)=   .0604
    E2(5)=   .0604
    E2(6)=   .0604
    D=  -.0005
    E=   .0641
    MINT(1)=  .09905
    MINT(2)=  .09905
    MINT(3)=  .09905
    MINT(4)=  .09905
    MINT(5)=  .09905
    MINT(6)=  .09905
------------------------------------------------------------------
    CALL PRESSBND(IBIG,ISMALL,PRESS)
    CALL CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,D2,E2,D,E,IBIG,ISMALL)
    RETURN
    END

==================================================================
    SUBROUTINE CV5(SPLIT,MTOT,PRESS)
    IMPLICIT NONE
    INTEGER IBIG,ISMALL
    REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
------------------------------------------------------------------
-form of correlation:   split = A*mtot^2 + B*mtot + C

-setting array values (determined by least squares fit to ASSERT
 output)
------------------------------------------------------------------
    A(1)=   .0165
    A(2)=   .1145
    A(3)=   .2530
    A(4)=   .2144
    A(5)=   .2067
    A(6)=   .2218
    B(1)=   .0066
    B(2)=  -.0138
    B(3)=  -.0456
    B(4)=  -.0423
    B(5)=  -.0443
    B(6)=  -.0496
    C(1)=   .0400
    C(2)=   .0414
    C(3)=   .0433
    C(4)=   .0437
    C(5)=   .0442
    C(6)=   .0448
    D= .6000E-04
    E= .4220E-01
    MINT(1)=  .20948
    MINT(2)=  .15853
    MINT(3)=  .14775
    MINT(4)=  .14553
    MINT(5)=  .14243
    MINT(6)=  .12674
------------------------------------------------------------------
    CALL PRESSBND(IBIG,ISMALL,PRESS)
    CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
    RETURN
    END

==================================================================
    SUBROUTINE CV6(SPLIT,MTOT,PRESS)
    IMPLICIT NONE
    INTEGER IBIG,ISMALL
    REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
------------------------------------------------------------------
-form of correlation:   split = A*mtot^2 + B*mtot + C

-setting array values (determined by least squares fit to ASSERT
 output)
------------------------------------------------------------------
    A(1)=  -.9778
```

```
      A(2)=  -.5221
      A(3)=   .1791
      A(4)=   .5990
      A(5)=   .6432
      A(6)=   .6312
      B(1)=   .0519
      B(2)=  -.0082
      B(3)=  -.1160
      B(4)=  -.1820
      B(5)=  -.1819
      B(6)=  -.1730
      C(1)=   .1933
      C(2)=   .1892
      C(3)=   .1892
      C(4)=   .1894
      C(5)=   .1880
      C(6)=   .1866
      D=-.1200E-02
      E= .1744E+00
      MINT(1)=  .15990
      MINT(2)=  .14709
      MINT(3)=  .13505
      MINT(4)=  .18904
      MINT(5)=  .18602
      MINT(6)=  .18831
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CV7(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL D2(10),E2(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*mtot^2 + B*mtot + C
C
C--setting array values (determined by least squares fit to ASSERT
C  output)
C-----------------------------------------------------------------------
      D2(1)=  -.0031
      D2(2)=  -.0031
      D2(3)=  -.0031
      D2(4)=  -.0031
      D2(5)=  -.0031
      D2(6)=  -.0031
      E2(1)=   .0245
      E2(2)=   .0245
      E2(3)=   .0245
      E2(4)=   .0245
      E2(5)=   .0245
      E2(6)=   .0245
      D=  -.0003
      E=   .0295
      MINT(1)=  .16769
      MINT(2)=  .16769
      MINT(3)=  .16769
      MINT(4)=  .16769
      MINT(5)=  .16769
      MINT(6)=  .16769
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,D2,E2,D,E,IBIG,ISMALL)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CV8(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL D2(10),E2(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*(mtot)**B
C
C--setting array values (determined by least squares fit to ASSERT
C  output)
C-----------------------------------------------------------------------
      D2(1)=  -.0030
      D2(2)=  -.0030
      D2(3)=  -.0030
      D2(4)=  -.0030
      D2(5)=  -.0030
```

```
   D2(6)=  -.0030
   E2(1)=   .0684
   E2(2)=   .0684
   E2(3)=   .0684
   E2(4)=   .0684
   E2(5)=   .0684
   E2(6)=   .0684
   D=  -.0006
   E=   .0727
   MINT(1)=  .16673
   MINT(2)=  .16673
   MINT(3)=  .16673
   MINT(4)=  .16673
   MINT(5)=  .16673
   MINT(6)=  .16673
-----------------------------------------------------------------
   CALL PRESSBND(IBIG,ISMALL,PRESS)
   CALL CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,D2,E2,D,E,IBIG,ISMALL)
   RETURN
   END

=================================================================
   SUBROUTINE CV9(SPLIT,MTOT,PRESS)
   IMPLICIT NONE
   INTEGER IBIG,ISMALL
   REAL D2(10),E2(10),D,E,MINT(10),SPLIT,PRESS,MTOT
-----------------------------------------------------------------
-form of correlation:   split = A*(mtot)**B

-setting array values (determined by least squares fit to ASSERT
 output)
-----------------------------------------------------------------
   D2(1)=  -.0033
   D2(2)=  -.0033
   D2(3)=  -.0033
   D2(4)=  -.0033
   D2(5)=  -.0033
   D2(6)=  -.0033
   E2(1)=   .0445
   E2(2)=   .0445
   E2(3)=   .0445
   E2(4)=   .0445
   E2(5)=   .0445
   E2(6)=   .0445
   D=  -.0004
   E=   .0507
   MINT(1)=  .11792
   MINT(2)=  .11792
   MINT(3)=  .11792
   MINT(4)=  .11792
   MINT(5)=  .11792
   MINT(6)=  .11792
-----------------------------------------------------------------
   CALL PRESSBND(IBIG,ISMALL,PRESS)
   CALL CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,D2,E2,D,E,IBIG,ISMALL)
   RETURN
   END

=================================================================
   SUBROUTINE CV10(SPLIT,MTOT,PRESS)
   IMPLICIT NONE
   INTEGER IBIG,ISMALL
   REAL D2(10),E2(10),D,E,MINT(10),SPLIT,PRESS,MTOT
-----------------------------------------------------------------
-form of correlation:   split = A*(mtot)**B

-setting array values (determined by least squares fit to ASSERT
 output)
-----------------------------------------------------------------
   D2(1)=  -.001598
   D2(2)=  -.001598
   D2(3)=  -.001598
   D2(4)=  -.001299
   D2(5)=  -.001299
   D2(6)=  -.001025
   E2(1)=   .043874
   E2(2)=   .043874
   E2(3)=   .043874
   E2(4)=   .044354
   E2(5)=   .044354
   E2(6)=   .044940
   D=  -.000253
   E=   .046905
   MINT(1)=  .10418
```

```
      MINT(2)=  .10418
      MINT(3)=  .10418
      MINT(4)=  .08730
      MINT(5)=  .08730
      MINT(6)=  .07849
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,D2,E2,D,E,IBIG,ISMALL)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CV12(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*(mtot)**B
C
C--setting array values (determined by least squares fit to ASSERT
C  output)
C-----------------------------------------------------------------------
      A(1)=   .7018
      A(2)=  -.2678
      A(3)= -1.1766
      A(4)= -1.1766
      A(5)= -1.1766
      A(6)= -1.1766
      B(1)=  -.0189
      B(2)=   .1276
      B(3)=   .2642
      B(4)=   .2642
      B(5)=   .2642
      B(6)=   .2642
      C(1)=   .0628
      C(2)=   .0631
      C(3)=   .0633
      C(4)=   .0633
      C(5)=   .0633
      C(6)=   .0633
      D= .5000E-03
      E= .7760E-01
      MINT(1)=  .15468
      MINT(2)=  .16063
      MINT(3)=  .14783
      MINT(4)=  .14783
      MINT(5)=  .14783
      MINT(6)=  .14783
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CV13(SPLIT,MTOT,PRESS)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
C-----------------------------------------------------------------------
C--form of correlation:   split = A*(mtot)**B
C
C--setting array values (determined by least squares fit to ASSERT
C  output)
C-----------------------------------------------------------------------
      A(1)=   .8952
      A(2)=  1.9653
      A(3)=  1.4039
      A(4)=   .5625
      A(5)=   .0842
      A(6)=  -.2708
      B(1)=  -.0348
      B(2)=  -.0996
      B(3)=  -.0027
      B(4)=   .0890
      B(5)=   .1392
      B(6)=   .1734
      C(1)=   .0469
      C(2)=   .0508
      C(3)=   .0506
      C(4)=   .0505
      C(5)=   .0508
      C(6)=   .0514
      D= .1000E-02
```

```
   E= .6640E-01
   MINT(1)=  .16129
   MINT(2)=  .11174
   MINT(3)=  .09898
   MINT(4)=  .09508
   MINT(5)=  .08988
   MINT(6)=  .08287
-----------------------------------------------------------------------
   CALL PRESSBND(IBIG,ISMALL,PRESS)
   CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
   RETURN
   END

=======================================================================
   SUBROUTINE CV14(SPLIT,MTOT,PRESS)
   IMPLICIT NONE
   INTEGER IBIG,ISMALL
   REAL A(10),B(10),C(10),D,E,MINT(10),SPLIT,PRESS,MTOT
-----------------------------------------------------------------------
-form of correlation:   split = A*(mtot)**B

-setting array values (determined by least squares fit to ASSERT
 output)
-----------------------------------------------------------------------
   A(1)=   .0350
   A(2)=   .0552
   A(3)=  -.0419
   A(4)=  -.0826
   A(5)=  -.0706
   A(6)=  -.0705
   B(1)=   .0255
   B(2)=   .0195
   B(3)=   .0315
   B(4)=   .0340
   B(5)=   .0278
   B(6)=   .0239
   C(1)=   .0100
   C(2)=   .0114
   C(3)=   .0120
   C(4)=   .0126
   C(5)=   .0133
   C(6)=   .0140
   D= .2000E-03
   E= .1610E-01
   MINT(1)=  .18090
   MINT(2)=  .15439
   MINT(3)=  .14653
   MINT(4)=  .13649
   MINT(5)=  .12647
   MINT(6)=  .22579
-----------------------------------------------------------------------
   CALL PRESSBND(IBIG,ISMALL,PRESS)
   CALL CALCSPLIT(SPLIT,PRESS,MTOT,MINT,A,B,C,D,E,IBIG,ISMALL)
   RETURN
   END

=======================================================================
   SUBROUTINE PRESSBND(IBIG,ISMALL,PRESS)
   IMPLICIT NONE
   INTEGER IBIG,ISMALL
   REAL P(10),PRESS
-----------------------------------------------------------------------
   P(1)=0.25
   P(2)=1.0
   P(3)=2.0
   P(4)=3.0
   P(5)=4.0
   P(6)=5.0
-----------------------------------------------------------------------
-determining bounding pressures:

-----low pressure:
-----------------------------------------------------------------------
   IF (PRESS.LT.P(1)) THEN
      IBIG=2
      ISMALL=1
      GOTO 100
   ENDIF
-----------------------------------------------------------------------
-----intermediate pressures:
-----------------------------------------------------------------------
     IF ((PRESS.GE.P(1)).AND.(PRESS.LT.P(2))) THEN
       IBIG=2
       ISMALL=1
```

```
          GOTO 100
        ENDIF
C
        IF ((PRESS.GE.P(2)).AND.(PRESS.LT.P(3))) THEN
          IBIG=3
          ISMALL=2
          GOTO 100
        ENDIF
C
        IF ((PRESS.GE.P(3)).AND.(PRESS.LT.P(4))) THEN
          IBIG=4
          ISMALL=3
          GOTO 100
        ENDIF
C
        IF ((PRESS.GE.P(4)).AND.(PRESS.LT.P(5))) THEN
          IBIG=5
          ISMALL=4
          GOTO 100
        ENDIF
C
        IF ((PRESS.GE.P(5)).AND.(PRESS.LT.P(6))) THEN
          IBIG=6
          ISMALL=5
          GOTO 100
        ENDIF
C-----------------------------------------------------------------------
C------high pressure:
C-----------------------------------------------------------------------
      IF (PRESS.GE.P(6)) THEN
          IBIG=6
          ISMALL=5
          GOTO 100
      ENDIF
C-----------------------------------------------------------------------
C------- ending determination of bounding pressures:
C-----------------------------------------------------------------------
 100  CONTINUE
C
      RETURN
      END
C
C=======================================================================
      SUBROUTINE CALCSPLIT(SPLIT,PRESS,MTOT,MINT,
     :                     A,B,C,D,E,IBIG,ISMALL)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL SPLIT,PRESS,MTOT,MINT(10),A(10),B(10),C(10),D,E,P(10),
     :     SPLITSMALL,SPLITBIG
C-----------------------------------------------------------------------
      P(1)=0.25
      P(2)=1.0
      P(3)=2.0
      P(4)=3.0
      P(5)=4.0
      P(6)=5.0
C-----------------------------------------------------------------------
C---calculate the flow splits, starting with lower pressure value:
C-----------------------------------------------------------------------
      IF (MTOT.LE.MINT(ISMALL)) THEN
        SPLITSMALL=A(ISMALL)*MTOT**2.0+B(ISMALL)*MTOT+C(ISMALL)
      ELSE
        SPLITSMALL=D*LOG(MTOT)+E
      ENDIF
C-----------------------------------------------------------------------
C---calculate the flow split for higher pressure value:
C-----------------------------------------------------------------------
      IF (MTOT.LE.MINT(IBIG)) THEN
        SPLITBIG=A(IBIG)*MTOT**2.0+B(IBIG)*MTOT+C(IBIG)
      ELSE
        SPLITBIG=D*LOG(MTOT)+E
      ENDIF
C-----------------------------------------------------------------------
C---interpolate flowsplit to correct pressure:
C-----------------------------------------------------------------------
      SPLIT=SPLITSMALL + ((PRESS-P(ISMALL))/(P(IBIG)-P(ISMALL)))*
     :                 (SPLITBIG-SPLITSMALL)
C
      RETURN
      END
C
C
C=======================================================================
      SUBROUTINE CALCSPLIT2(SPLIT,PRESS,MTOT,MINT,
```

```
     :                    D2,E2,D,E,IBIG,ISMALL)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL
      REAL SPLIT,PRESS,MTOT,MINT(10),D2(10),E2(10),D,E,P(10),
     :     SPLITSMALL,SPLITBIG
C-----------------------------------------------------------------------
      P(1)=0.25
      P(2)=1.0
      P(3)=2.0
      P(4)=3.0
      P(5)=4.0
      P(6)=5.0
C-----------------------------------------------------------------------
C-calculate the flow splits, starting with lower pressure value:
C-----------------------------------------------------------------------
      IF (MTOT.LE.MINT(ISMALL)) THEN
        SPLITSMALL=D2(ISMALL)*LOG(MTOT)+E2(ISMALL)
      ELSE
        SPLITSMALL=D*LOG(MTOT)+E
      ENDIF
C-----------------------------------------------------------------------
C-calculate the flow split for higher pressure value:
C-----------------------------------------------------------------------
      IF (MTOT.LE.MINT(IBIG)) THEN
        SPLITBIG=D2(IBIG)*LOG(MTOT)+E2(IBIG)
      ELSE
        SPLITBIG=D*LOG(MTOT)+E
      ENDIF
C-----------------------------------------------------------------------
C--interpolate flowsplit to correct pressure:
C-----------------------------------------------------------------------
      SPLIT=SPLITSMALL + ((PRESS-P(ISMALL))/(P(IBIG)-P(ISMALL)))*
     :                 (SPLITBIG-SPLITSMALL)

      RETURN
      END

C=======================================================================
      SUBROUTINE SUMCV(SPLIT,CVI,SPLITARY)
      IMPLICIT NONE
      INTEGER I,CVI
      REAL SPLIT,SPLITARY(15),SUM
C-----------------------------------------------------------------------
C--This subroutine ensures that the sum of the mass splits is unity
C-----------------------------------------------------------------------
      SPLITARY(CVI)=0.0
      SUM=0.0
C-----------------------------------------------------------------------
C--summing up the flow splits for the control volumes:
C-----------------------------------------------------------------------
      DO 10 I=1,14
       SUM=SUM+SPLITARY(I)
10    CONTINUE
C-----------------------------------------------------------------------
C--setting the value of SPLIT at CVI equal to unity minus the
C  sum of the other flowsplits
C-----------------------------------------------------------------------
      SPLIT=1.0-SUM

      RETURN
      END
```

```
      SUBROUTINE QBUNDLE(QBUN,QCHAN,BUNDLE)
      IMPLICIT NONE
      INTEGER BUNDLE
      REAL QBUN,QCHAN
C-----------------------------------------------------------------------
C     BUNDLE = bundle number for current calculation [-]
C     QBUN = fuel power for current bundle [MJ/s]
C     QCHAN = total channel power [MW]
C
C This subroutine calculates the total bundle power.  The fractional
C power comes from the axial flux distribution used in ASSERT-PV.
C this is a cosine function max/average=1.6
C-----------------------------------------------------------------------
      IF (BUNDLE.EQ.1)  QBUN=0.0170*QCHAN
      IF (BUNDLE.EQ.2)  QBUN=0.0500*QCHAN
      IF (BUNDLE.EQ.3)  QBUN=0.0795*QCHAN
      IF (BUNDLE.EQ.4)  QBUN=0.1036*QCHAN
      IF (BUNDLE.EQ.5)  QBUN=0.1206*QCHAN
      IF (BUNDLE.EQ.6)  QBUN=0.1294*QCHAN
      IF (BUNDLE.EQ.7)  QBUN=0.1294*QCHAN
      IF (BUNDLE.EQ.8)  QBUN=0.1206*QCHAN
      IF (BUNDLE.EQ.9)  QBUN=0.1036*QCHAN
      IF (BUNDLE.EQ.10) QBUN=0.0795*QCHAN
      IF (BUNDLE.EQ.11) QBUN=0.0500*QCHAN
      IF (BUNDLE.EQ.12) QBUN=0.0170*QCHAN

      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE QCV(QFUEL,QBUN,BUNDLE,NUMV,ID)
      IMPLICIT NONE
      INTEGER ID,I,NUMV,BUNDLE
      REAL QFUEL(ID),QBUN,CHECK
C-----------------------------------------------------------------------
C     This subroutine calculates the rate of energy input from
C     the fuel to the control volume.  The constants used come
C     from the radial flux distribution specified in ASSERT
C     and the fuel area in contact with each FACTAR control volume.

C     QFUEL(I) = fuel power into control volume I [MJ/s]
C     QBUN = total fuel power for current bundle [MJ/s]
C     CHECK = sum of QFUEL(I) - checking to make sure that the
C             total energy distributed to each cv is correct
C     BUNDLE = current bundle number
C-----------------------------------------------------------------------
      QFUEL(1)= 0.032626*QBUN
      QFUEL(2)= 0.032626*QBUN
      QFUEL(3)= 0.094081*QBUN
      QFUEL(4)= 0.070676*QBUN
      QFUEL(5)= 0.047271*QBUN
      QFUEL(6)= 0.195585*QBUN
      QFUEL(7)= 0.026420*QBUN
      QFUEL(8)= 0.084029*QBUN
      QFUEL(9)= 0.055168*QBUN
      QFUEL(10)=0.056388*QBUN
      QFUEL(11)=0.152580*QBUN
      QFUEL(12)=0.067813*QBUN
      QFUEL(13)=0.067813*QBUN
      QFUEL(14)=0.016953*QBUN
C-----------------------------------------------------------------------
C doing a check to make sure the sum of QFUEL(i) = QBUN
C-----------------------------------------------------------------------
      CHECK=0.0
      DO 10 I=1,NUMV
         CHECK=CHECK+QFUEL(I)
10    CONTINUE
C-----------------------------------------------------------------------
C writing out if the check fails:
C-----------------------------------------------------------------------
      IF (ABS((CHECK-QBUN)/QBUN).GT.1.0E-4) THEN
       WRITE(6,*) 'Message from fuel.f: Energy distribution incorrect'
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE PROPCV(PRESS,HINN,
     :           TEMP, TEMPK, DEN, CP, VISC, COND,TCONST,
     :           TEMPS,TEMPKS,DENS,CPS,VISCS,CONDS,NUMV,ID)
      IMPLICIT NONE
      INTEGER ID,IERRV,I,NUMV
      REAL TCONST, HINN(ID),PRESS
      REAL TEMP(ID), DEN(ID), CP(ID), VISC(ID), COND(ID), TEMPK(ID)
      REAL TEMPS(ID),DENS(ID),CPS(ID),VISCS(ID),CONDS(ID),TEMPKS(ID)
C-----------------------------------------------------------------
C        PRESS - PRESSURE (MPa)
C        HINN - SPECIFIC ENTHALPY (J/kg)
C        TEMP  - TEMPERATURE (C)
C        DEN  - DENSITY (kg/m3)
C        CP - SPECIFIC HEAT AT CONST. PRESSURE (J/kg-K)
C        VISC - DYNAMIC VISCOSITY (N-s/m2)
C        COND - THERMAL CONDUCTIVITY (W/m-K)
C
C        TEMPS  - saturation TEMPERATURE (C)
C        DENS  - saturation DENSITY (kg/m3)
C        CPS - saturation SPECIFIC HEAT AT CONST. PRESSURE (J/kg-K)
C        VISCS - saturation DYNAMIC VISCOSITY (N-s/m2)
C        CONDS - saturation THERMAL CONDUCTIVITY (W/m-K)
C-----------------------------------------------------------------
      DO 10 I=1,NUMV
       CALL PROP(PRESS*1.0E06,HINN(I)*1.0E03,
     :      TEMP(I), DEN(I), CP(I), VISC(I), COND(I),
     :      TEMPS(I),DENS(I),CPS(I),VISCS(I),CONDS(I),IERRV)

       TEMPK(I) =TEMP(I)+TCONST
       TEMPKS(I)=TEMPS(I)+TCONST
   10 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE WTURB(WPG,MDOT,VISC,DHYD,AREAX,RODDIA,GAP,PWET,DEN,
     :           IEG,IWG,NUMG,ID,KD)
      IMPLICIT NONE
      INTEGER ID,KD,IEG(KD),IWG(KD),NUMG,I,J,K
      REAL WPG(KD),MDOT(ID),DHYD(ID),AREAX(ID),RODDIA(KD),GAP(KD),
     :     PWET(ID),DEN(ID),VISC(ID),REI,REJ,SMI,SMJ,LIJ,GI,GJ
C-----------------------------------------------------------------------
      This subroutine uses Petrunik's vapour thermal mixing
      correlation to calculate the mixing coefficient w'. Note
      that this quantity is associated with a GAP not a control
      volume.

      The Reynolds number which appears in the correlation is
      based on the total hydraulic diameter and total flow
      through the two adjacent control volumes.

      WPG(IG) - effective mass flow through gap IG due to turb
                per unit length of control volume and for a single
                ASSERT gap [kg/ms]
      MDOT(I) - mass flow through cv I [kg/s]
      VISC(I) - viscosity of fluid in cv I [kg/ms]
      DHYD(I) - hydraulic diameter of control volume I [m]
      AREAX(I) - cross-sectional area of cv I [m2]
      PWET(I) - wetted perimeter of cv I [m]
      DEN(I) - density of fluid in cv I [kg/m3]
      REG(IG) - average Reynolds number of adjacent cv to gap IG
      REI,REJ=Reynolds number for CVs I and J
      SMI,SMJ laminar mixing reduction factors
      LIJ=Rogers and Rosehart gap size/rod size factor for mixing
      GI,GJ=mass fluxes for I and J (kg/m^2s)
      WPG=turbulent diffusion flow rate (kg/ms)
C-----------------------------------------------------------------------
      DO 10 K=1,NUMG
        I=IWG(K)
        J=IEG(K)
        REI=(MDOT(I)/AREAX(I))*DHYD(I)/VISC(I)
        REJ=(MDOT(J)/AREAX(J))*DHYD(J)/VISC(J)
        SMI=(1.0+(DHYD(J)/DHYD(I))**1.5)*
     :       (DHYD(I)/RODDIA(K))/(2.0*REI**0.1)
        SMJ=(1.0+(DHYD(I)/DHYD(J))**1.5)*
     :       (DHYD(J)/RODDIA(K))/(2.0*REJ**0.1)
        LIJ=0.0058*(GAP(K)/RODDIA(K))**(-1.46)
        GI=MDOT(I)/AREAX(I)
        GJ=MDOT(J)/AREAX(J)
        WPG(K)=0.5*LIJ*(SMI*GI+SMJ*GJ)*GAP(K)
0     CONTINUE
      CALL OUT1D(WPG,'WPG      ',17,1,NUMG,1,KD)

      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE DIFPHI(DG, WPG,AREAG,GAP,MDOT,LEN,IWG,IEG,NUMG,ID,KD)
      IMPLICIT NONE
      INTEGER NUMG,ID,KD,IG,I,K,IWG(KD),IEG(KD)
      REAL DG(KD),WPG(KD),AREAG(KD),GAP(KD),MDOT(ID),MC
      REAL CRAD(2),LEN
C-----------------------------------------------------------------------
      This subroutine calculates the 'diffusion' coefficient
      due to turbulent diffusion and diversion flow mixing

            DG(IG) = diffusion coefficient [kg/s]
            WPG(IG) = thermal mixing coeff [kg/ms]
            AREAG(IG) = area of control volume gap IG [m2]
            GAP(IG) = width of ASSERT gap for gap IG [m]
            MDOT(I) = mass flow through cv I [kg/s]
            NUMG = number of FACTAR gaps
            ID = array dimension
C-----------------------------------------------------------------------
 first, calculating the contribution from turbulence:
C-----------------------------------------------------------------------
       DO 5 IG=1,NUMG
          DG(IG)=AREAG(IG)*WPG(IG)/GAP(IG)
       CONTINUE
      CALL OUT1D(DG,'DG       ',17,1,NUMG,1,KD)
C-----------------------------------------------------------------------
 second, calculating the contribution to DG(IG) from cross-flow mixing:
C-----------------------------------------------------------------------
      CRAD(1)=0.02
      DO K=1,13
        DG(K)=DG(K)+CRAD(1)*(MDOT(IWG(K))+MDOT(IEG(K)))
      ENDDO
      CALL OUT1D(DG,'DG       ',17,1,NUMG,1,KD)
C-----------------------------------------------------------------------
```

```
C  gaps 14 through 23 are in the circumferential direction and convection
C  will not be accounted for so there are no changes to (DG(i),i=14,...,23)
C-----------------------------------------------------------------------
      RETURN
      END
```

```
      SUBROUTINE BUOYMASS(BMASS,PRESS,MTOT,QCHAN,HINI,MULT,NUMG,KD)
      IMPLICIT NONE
      INTEGER IBIG,ISMALL,K,KD,NUMG
      REAL C(6),D(6),E(6),P(6),X,BMASS(KD),B(KD),BCON,BCONSMALL,BCONBIG,
     :     PRESS,MTOT,QCHAN,HINI,PFRAC,MULT,CON
C-----------------------------------------------------------------------
C this subroutine calculates the appropriate degree of mixing
C due to buoyancy based on the channel pressure, channel power, and
C channel mass flow.
C   P   =channel pressure(MPa)
C   C,D,E=coefficients for proportionality factor BCON
C   X   =x value in the BCON correlation
C   BCON=proportionality factor
C   B(K)=symbolic mass flow at gap K.
C-----------------------------------------------------------------------
C define pressure ranges
C-----------------------------------------------------------------------
      P(1)=0.25
      P(2)=1.0
      P(3)=2.0
      P(4)=3.0
      P(5)=4.0
      P(6)=5.0
C-----------------------------------------------------------------------
C form of correlation:   BCON = C*x^2 + D*x + E
C where x = (QCHAN/MTOT)^3+(0.001H)^3)
C-----------------------------------------------------------------------
      C(1)=   0.00020125
      C(2)=   0.00051625
      C(3)=   0.00119875
      C(4)=   0.001617
      C(5)=   0.001931125
      C(6)=   0.002184
      D(1)=   -0.02538375
      D(2)=   -0.0685125
      D(3)=   -0.15199625
      D(4)=   -0.21501375
      D(5)=   -0.258335
      D(6)=   -0.29372
      E(1)=   0.9792475
      E(2)=   3.105795
      E(3)=   6.41823875
      E(4)=   9.39826125
      E(5)=   11.65476375
      E(6)=   13.581785
C-----------------------------------------------------------------------
C establish which pressure range the channel pressure falls under
C-----------------------------------------------------------------------
      CALL PRESSBND(IBIG,ISMALL,PRESS)
      WRITE(17,*)'IBIG,ISMALL',IBIG,ISMALL
C-----------------------------------------------------------------------
C calculate the upper and lower bounds of BCON based on
C the pressure range
C-----------------------------------------------------------------------
      X=(QCHAN/MTOT)**3.0+(0.001*HINI)**3.0
      BCONSMALL=C(ISMALL)*(X**2.0)+D(ISMALL)*X+E(ISMALL)
      BCONBIG  =C(IBIG)  *(X**2.0)+D(IBIG)  *X+E(IBIG)
      WRITE(17,*)'BCONSMALL,BCONBIG',BCONSMALL,BCONBIG
C-----------------------------------------------------------------------
C calculate BCON using interpolation
C-----------------------------------------------------------------------
      PFRAC=(PRESS-P(ISMALL))/(P(IBIG)-P(ISMALL))
      BCON=BCONSMALL+PFRAC*(BCONBIG-BCONSMALL)
C-----------------------------------------------------------------------
C these are the gap mass flows predicted to occur under buoyancy
C dominant conditions.  These are only proportional to the actual
C mass flows, they must still be multiplied by BCON to get the true
C mass flow
C-----------------------------------------------------------------------
      B(1)=  0.000137915
      B(2)= -9.56161E-05
      B(3)= -4.36942E-05
      B(4)=  0.000250811
      B(5)= -1.28908E-05
      B(6)= -9.9217E-05
      B(7)= -8.39496E-05
      B(8)= -5.78781E-05
      B(9)=  0.000271447
      B(10)=-0.000110188
      B(11)=-7.71355E-05
      B(12)=-0.000162829
      B(13)= 7.87057E-05
      B(14)=-0.000137915
      B(15)=-0.000112897
```

```
      B(16)=-9.95293E-05
      B(17)=-2.53339E-05
      B(18)=-3.04494E-05
      B(19)=-2.04204E-05
      B(20)=-2.6245E-05
      B(21)= 0.000271447
      B(22)= 0.000161259
      B(23)=-7.87057E-05
C
      DO K=1,NUMG
       BMASS(K)=B(K)*BCON*MULT
      ENDDO
C
      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE BUOYFLO(DG,DGN,BMASS,QCHAN,MDOT,AREAX,DEN,LEN,
     :           NUMG,NUMV,GAPN,GAPV,OPCV,GN,IWG,IEG,ID,KD)
      IMPLICIT NONE
      INTEGER I,J,K,IG,ID,KD,NUMG,NUMV,IWG(KD),IEG(KD),
     :        GAPN(ID),GAPV(ID,ID),IW,IE,OPCV(ID,ID),GN(ID,KD)
      REAL DG(KD),BMASS(KD),DGN(ID,ID),ALP(ID,ID),MDOT(ID),VAVG,
     :     QCHAN,AREAX(ID),DEN(ID),AREASUM,RHOAVG,GRAV,CONST,LEN,VOL
C-----------------------------------------------------------------------
C  This subroutine calculates the total mass flow at any one gap
C     ALP(I,J)=the flow direction factor for the buoyancy cross flow
C              for CV I adjacent to CV J.
C     DG(K)   =thetotal diffusion flow at gap K
C     BMASS(K)=the total buoyancy croos flow at gap K.
C-----------------------------------------------------------------------
C  ALP is equivalent to ALPHA, a value which is generated from the
C  Peclet number of the advective flow.
C-----------------------------------------------------------------------
      DO I=1,NUMV
       DO J=1,NUMV
        ALP(I,J)=0.0
       ENDDO
      ENDDO
C-----------------------------------------------------------------------
C  the value of ALP is dependant on the direction of B
C  if B is leaving the CV, then ALP=0.0,
C  if it is entering the CV, then ALP=+1.0
C-----------------------------------------------------------------------
      DO K=1,NUMG
       IW=IWG(K)
       IE=IEG(K)
       IF (BMASS(K).GT.0.0) THEN
        ALP(IW,GN(IW,K))=0.0
        ALP(IE,GN(IE,K))=+1.0
       ELSE
        ALP(IW,GN(IW,K))=+1.0
        ALP(IE,GN(IE,K))=0.0
       END IF
C-----------------------------------------------------------------------
C  DGN(I,K) is the total mass flow entering CV I at gap K, due to
C  both turbulent diffusion and buoyant convection.
C  the turbulent diffusion component is DG(K), the buoyant convection
C  component is ALP(I,K)*abs(BMASS(K))
C-----------------------------------------------------------------------
       DGN(IW,GN(IW,K))=DG(K)+ALP(IW,GN(IW,K))*ABS(BMASS(K))
       DGN(IE,GN(IE,K))=DG(K)+ALP(IE,GN(IE,K))*ABS(BMASS(K))
      ENDDO
C
      RETURN
      END
```

```
   SUBROUTINE COEFF(APP, DGN,MDOT,IEG,IWG,NUMG,NUMV,GAPN,ID,KD)
   IMPLICIT NONE
   INTEGER ID,KD,IG,I,J,NUMG,NUMV,IWG(KD),IEG(KD),IW,IE,GAPN(ID)
   REAL APP(ID),MDOT(ID),DGN(ID,ID)
-------------------------------------------------------------------
   This subroutine calculates the APP coefficient for enthalpy.

       APP(I) = ap coefficient for cv I [kg/s]
       MDOT(I) = mass flow through cv I [kg/s]
       DGN(IG) = total mass flow for gap IG [kg/s]
       NUMG = number of gaps
       NUMV = number of control volumes
       ID = array dimension
-------------------------------------------------------------------
next calculate the contribution from the 'flow' through the gaps
initiallize APP to equal the axial massflow through the CV, and then
add on all of the lateral flow (through the gaps) contributions
-------------------------------------------------------------------
   DO I=1,NUMV
    APP(I)=MDOT(I)
    DO J=1,GAPN(I)
     APP(I)=APP(I)+DGN(I,J)
    ENDDO
   ENDDO

   RETURN
   END

-------------------------------------------------------------------
   SUBROUTINE SOURCE(BP, QFUEL,MDOT,HIN,NUMV,ID)
   IMPLICIT NONE
   INTEGER ID,I,NUMV
   REAL BP(ID),QFUEL(ID),MDOT(ID)
   REAL HIN(ID)
-------------------------------------------------------------------
    BP(I) = source term for discrete equation [kJ/s]
    QFUEL(I) = fuel power into control volume I [MJ/s]
    MDOT(I) = mass flow through control volume I [kg/s]
    HIN(I) = inlet enthalpy for cv I [kJ/kg]
    NUMV = total number of control volumes
    ID = array dimension

    This subroutine calculates the source term for the discrete
    equation for coolant enthalpy.

    The source is from the energy convected into the cv plus
    the energy from the fuel.
-------------------------------------------------------------------
   DO 10 I=1,NUMV
      BP(I)=MDOT(I)*HIN(I) + 1000.0*QFUEL(I)
10 CONTINUE

   RETURN
   END
```

```
      SUBROUTINE WALLHT(QLOSS,TMOD,HMOD,TW,TCONST,TEMPK,MDOT,
     :                  COND,CP,VISC,ROT,LEN,AREAX,DHYD,ANG,WID,
     :                  CON,WALMAT,ID,JD,KD,JB,JE,IB,IE)
      IMPLICIT NONE
      INTEGER ID,JD,KD,I,J,K,JB,JE,IB,IE,WALMAT(JD),PTFILM
      REAL ROT(JD),AREAX(ID),DHYD(ID),LEN,ANG(ID),WID(JD),
     :     RE(ID),PR(ID),NULAM,NUTURB(ID),NU(ID),
     :     KW(JD,ID),HCOOL(ID),HMOD,
     :     TW(JD,ID),TMOD,TCONST,
     :     R(JD,ID),RTOT(ID),QLOSS(ID),MDOT(ID)
      REAL TEMPK(ID),COND(ID),CP(ID),VISC(ID),CON
C-----------------------------------------------------------------------
      this subroutine calculates the heat lost to the moderator based
      on the present subchannel enthalpies, properties and wall temps
      HCOOL(I)=the PT wall convection coefficient of subchannel I.(W/m2K)
      HMOD=moderator CT wall convection coefficient (W/m2K)
      TW(J,I)=temperature of wall segment J closest to subchannel I (K)
      KW(J,I)=conduction coefficient of wall segment J closest to
              subchannel I (W/mK)
      R(J,I)=resistance to heat transfer of wall segment J closest
              to subchannel I (K/W)
      RTOT(I)=total resistance to heat transfer of walls closest
              to subchannel I. (K/W)
      QLOSS(I)=amount of heat lost from CV I to moderator (W)
C-----------------------------------------------------------------------
using laminar (7.86) and turbulent (Dittus Boelter) Nusselt
numbers, calculate coolant convection coefficient HCOOL (W/m^2K)
C-----------------------------------------------------------------------
Aug 21,2002: discovered ASSERT only uses turbulent Nu # for wall heat
loss. consequently, below code was modified to ignore laminar Nu #
C-----------------------------------------------------------------------
      NULAM=7.86
      DO I=IB,IE
       RE(I)=MDOT(I)*DHYD(I)/(VISC(I)*AREAX(I))
       PR(I)=CP(I)*VISC(I)/COND(I)
       NUTURB(I)=0.023*(RE(I)**0.8)*(PR(I)**0.4)
        NU(I)=MAX(NUTURB(I),NULAM)
       NU(I)=NUTURB(I)
       HCOOL(I)=NU(I)*COND(I)/DHYD(I)
      ENDDO
C-----------------------------------------------------------------------
 calculate conduction coefficients of each wall
 segment, for each subchannel, based on current temperatures.
 so far, only Zirc + 2.5%Nb, and CO2 are allowed as wall materials
C-----------------------------------------------------------------------
      DO 301 I=IB,IE
       DO 301 J=JB,JE
        CALL GTKW(TW(J,I),KW(J,I),TCONST,WALMAT(J))
01    CONTINUE
C-----------------------------------------------------------------------
 calculate all wall heat resistances R (K/W)
 The constant 1.44 is needed only for ASSERT comparison reasons.
 ASSERT incorrectly uses the saturated vapout properties to calculate
 the mod heat loss.
 When the unsaturated vapour properties are (correctly) used
 instead of the saturated liquid properties (as ASSERT uses),
 then this constant isn't needed.
C-----------------------------------------------------------------------
      DO I=IB,IE
       R(JB-1,I)=1/(CON*HCOOL(I)*ANG(I)*ROT(JB-1)*LEN)
       R(JE+1,I)=1/(CON*HMOD    *ANG(I)*ROT(JE+1)*LEN)
       DO J=JB,JE
        R(J,I)=LOG(ROT(J)/ROT(J-1))/(CON*ANG(I)*KW(J,I)*LEN)
       ENDDO
      ENDDO
C-----------------------------------------------------------------------
 set total resistance initially to zero and then sum all R(J,I)(K/W)
C-----------------------------------------------------------------------
      DO I=IB,IE
       RTOT(I)=0.0
      ENDDO
      DO I=IB,IE
       DO J=JB-1,JE+1
        RTOT(I)=RTOT(I)+R(J,I)
       ENDDO
      ENDDO
C-----------------------------------------------------------------------
 calculate heat loss (W) based on total wall heat resistance,
 mod temp, and coolant temp.  Also update wall temps.
C-----------------------------------------------------------------------
      DO I=1,IE
       QLOSS(I)=0.0
      ENDDO
      DO I=IB,IE
```

```
        QLOSS(I)=(TEMPK(I)-TMOD)/RTOT(I)
        TW(JB-1,I)=  TEMPK(I)-QLOSS(I)*R(JB-1,I)
        TW(JE+1,I)=      TMOD+QLOSS(I)*R(JE+1,I)
        DO J=JB,JE
         TW(J,I)=   TW(J-1,I)-QLOSS(I)*R(J,I)
        ENDDO
       ENDDO
C
       RETURN
       END


C-----------------------------------------------------------------------
       SUBROUTINE GTKW(TWZ,KWZ,TCONST,MATTYP)
       IMPLICIT NONE
       INTEGER MATTYP
       REAL TWZ,KWZ,TCONST
C-----------------------------------------------------------------------
       IF (MATTYP.EQ.1) THEN
C-----------------------------------------------------------------------
C  Pressure tube wall heat resistance (Zirc + 2.5% Nb)
C-----------------------------------------------------------------------
        IF ((TWZ-TCONST) .LE. 200.0) THEN
          KWZ = 20.2 + 0.008 * (TWZ - 150.0-TCONST)
        ELSE IF ((TWZ-TCONST) .LE. 250.0) THEN
C-----------------------------------------------------------------------
C  this (**...**) is ASSERT's correlation from subroutine 'COND.f'
C  (I think it is wrong because it creates a discontinuity
C  in the temperature-conduction curve)
C       **   KWZ = 17.7 + 0.004 * (TWZ - 200.0-TCONST)    **
C  the following line is the correlation that makes more sense
C  to me (it is continuous)
C-----------------------------------------------------------------------
          KWZ = 20.6 + 0.004 * (TWZ - 200.0-TCONST)
        ELSE IF ((TWZ-TCONST) .LE. 300.0) THEN
          KWZ = 20.8 + 0.010 * (TWZ - 250.0-TCONST)
        ELSE
          KWZ = 21.3 + 0.008 * (TWZ - 300.0-TCONST)
        END IF
C-----------------------------------------------------------------------
C  Gap heat resistance (Carbon Dioxide)-Same as ASSERT correlation
C-----------------------------------------------------------------------
       ELSE IF (MATTYP.EQ.2) THEN
        IF ((TWZ-TCONST) .LE. 537.7778) THEN
         KWZ = 1.4510D-2 + (TWZ-TCONST) * 0.76324D-4
        ELSE IF ((TWZ-TCONST) .LE. 815.5556) THEN
         KWZ = 2.0860D-2 + (TWZ-TCONST) * 0.63551D-4
        ELSE
         KWZ = 3.2039D-2 + (TWZ-TCONST) * 0.49844D-4
        END IF
       ELSE IF (MATTYP.EQ.3) THEN
        KWZ=0.0
       ELSE
       write(17,201)MATTYP
 201   format('no such material for WALMAT= ',i4)
       END IF
C
       RETURN
       END

C-----------------------------------------------------------------------
       SUBROUTINE WTIG(TW,TCONST,NUMW1,NUMW2,NUMW3,
      :           JD,JB,JE,ID,IB,IE)
       IMPLICIT NONE
       INTEGER NUMW1,NUMW2,NUMW3,J,I,JD,JB,JE,ID,IB,IE,M
       REAL TW(JD,ID),TCONST,T1,T2,T3,T4,INC2
C-----------------------------------------------------------------------
C  wall temp initial guesses for BUNDLE=1 (all temperatures in Kelvins)
C  these guesses are only rough approximations of the final wall temps,
C  and are only used to improve the rate of convergence of the middle
C  loop for the first bundle.
C-----------------------------------------------------------------------
C  T4 is the temp at the wall closest to the moderator
C-----------------------------------------------------------------------
       T4=83.5+TCONST
C-----------------------------------------------------------------------
C  T3 and T2 are the temperatures of the outside and inside segments
C  of the CO2 gap
C-----------------------------------------------------------------------
       T3=105.5+TCONST
       T2=216.5+TCONST
C-----------------------------------------------------------------------
C  T1 is the temperature of the wall closest to the outer subchannels
C-----------------------------------------------------------------------
```

```
   T1=233.5+TCONST
-----------------------------------------------------------------
INC2 is the amount by which the temperature drops in each successive
segment of the CO2 gap
-----------------------------------------------------------------
   IF (NUMW2.EQ.1) THEN
    INC2=0.0
   ELSE
    INC2=(T2-T3)/(NUMW2-1)
   END IF
-----------------------------------------------------------------
now assigning the temperatures to TW(J,K) based on the number of
wall segments requested in input.dat
-----------------------------------------------------------------
   DO 40 I=IB,IE
    DO 40 J=JB-1,JE+1
     M=J-NUMW1-JB
     IF (J.GE.JB-1) TW(J,I)=T1
     IF (J.GE.JB+NUMW1) THEN
       IF (NUMW2.EQ.1) THEN
        TW(J,I)=(T2+T3)*0.5
       ELSE
        TW(J,I)=T2-M*INC2
       END IF
     END IF
     IF (J.GE.JB+NUMW1+NUMW2) TW(J,I)=T4
   CONTINUE

   RETURN
   END

-----------------------------------------------------------------
   SUBROUTINE MODHL(QMODBUN,QMODCH,QLOSS,IB,IE,ID)
   IMPLICIT NONE
   INTEGER IB,IE,ID,I
   REAL QMODBUN,QMODCH,QLOSS(ID)
-----------------------------------------------------------------
calculating heat loss for each bundle and
heat loss for all the bundles (in MW)
-----------------------------------------------------------------
   QMODBUN=0.0
   DO I=IB,IE
    QMODBUN=QMODBUN+QLOSS(I)*1.0E-06
   ENDDO
   QMODCH=QMODCH+QMODBUN

   RETURN
   END
```

```
      SUBROUTINE PGS(H,ERR, APP,BP,DGN,OPCV,IWG,IEG,NUMV,NUMG,
     :               IMAX,GAPV,GAPN,INNRIT,INNTOT,ID,KD)
      IMPLICIT NONE
      INTEGER ID,KD,ITER,I,J,IEG(KD),IWG(KD),NUMV,NUMG,IMAX,
     :        GAPV(ID,ID),GAPN(ID),INNRIT,INNTOT,OPCV(ID,ID)
      REAL H(ID),APP(ID),BP(ID),DGN(ID,ID),RES,ERR,SUMDGH(ID)
C-----------------------------------------------------------------------
      This subroutine solves the discrete equations for
      the control volume enthalpies.

      Need to include both circumferential and radial diffusion

      H(I) = enthalpy of fluid at cv I [kJ/kg]
      RES = average residual at end of the solution [kJ/s]
      RESI = initial residual before solver [kJ/s]
      IEND = maximum number of solver iterations
      ERR = maximum error at end
      WRK(I) = work array for solver
      APP(I) = discrete equation coefficient for cv I[kg/s]
      BP(I) = discrete equation source term for cv I [kW]
      DGN(I,J) = diffusion coeffient for CV I from CV J. [kg/s]
      SUMDGH(I)=total energy entering CV I from surrounding CVs.
      INNRIT=number of inner iterations
C-----------------------------------------------------------------------
      DO I=1,NUMV
       SUMDGH(I)=0.0
      ENDDO
C-----------------------------------------------------------------------
 start iterations
C-----------------------------------------------------------------------
      RES=1.0
      INNRIT=0
      DO 20 ITER=1,IMAX
       INNRIT=INNRIT+1
C-----------------------------------------------------------------------
 sweeping forwards along subchannels
C-----------------------------------------------------------------------
 SUMDGH(I) is the sum of all energy entering a CV from neighboring CVs
C-----------------------------------------------------------------------
       DO I=1,NUMV
        DO J=1,GAPN(I)
          SUMDGH(I)=SUMDGH(I)+DGN(I,J)*H(OPCV(I,J))
        ENDDO
        H(I)=(SUMDGH(I)+BP(I))/APP(I)
        SUMDGH(I)=0.0
       ENDDO
C-----------------------------------------------------------------------
 sweeping backwards along subchannels
C-----------------------------------------------------------------------
       DO I=NUMV,1,-1
        DO J=1,GAPN(I)
          SUMDGH(I)=SUMDGH(I)+DGN(I,J)*H(OPCV(I,J))
        ENDDO
        H(I)=(SUMDGH(I)+BP(I))/APP(I)
        SUMDGH(I)=0.0
       ENDDO
C-----------------------------------------------------------------------
 calculate the residual, and exit the loop if it is
 sufficiently small
C-----------------------------------------------------------------------
       CALL RESID(RES, H,APP,BP,DGN,OPCV,
     :      NUMV,IWG,IEG,GAPN,GAPV,ID,KD)
       IF (RES.LT.ERR) THEN
        GOTO 30
       ENDIF
C-----------------------------------------------------------------------
 writing out flag for solver problems:
C-----------------------------------------------------------------------
       IF (ITER.GE.IMAX) THEN
           WRITE(6,*) 'PGS CONVERGENCE PROBLEM'
           WRITE(6,*) 'RES=',RES
           WRITE(17,*)'PGS CONVERGENCE PROBLEM'
           WRITE(17,*)'RES=',RES
       ENDIF

20    CONTINUE
30    CONTINUE
      WRITE(17,*)'INNRIT= ',INNRIT
      INNTOT=INNTOT+INNRIT

      RETURN
      END

C-----------------------------------------------------------------------
```

```
      SUBROUTINE RESID(RES, H,APP,BP,DGN,
     :           OPCV,NUMV,IWG,IEG,GAPN,GAPV,ID,KD)
      IMPLICIT NONE
      INTEGER NUMV,ID,KD,I,J,GAPN(ID),GAPV(ID,ID),
     :        IEG(KD),IWG(KD),OPCV(ID,ID)
      REAL RES,WRK(ID),H(ID),APP(ID),BP(ID),DGN(ID,ID)
      REAL SUMDGH(ID)
C-----------------------------------------------------------------------
C     This subroutine calculates the average residual in kJ/s
C     from the PGS solver.
C-----------------------------------------------------------------------
      DO I=1,NUMV
       SUMDGH(I)=0.0
      ENDDO
C
      DO I=1,NUMV
       DO J=1,GAPN(I)
        SUMDGH(I)=SUMDGH(I)+DGN(I,J)*H(OPCV(I,J))
       ENDDO
       WRK(I)=H(I)-(SUMDGH(I)+BP(I))/APP(I)
       SUMDGH(I)=0.0
      ENDDO
C-----------------------------------------------------------------------
C  averaging the errors:
C-----------------------------------------------------------------------
      RES=0.0
      DO 10 I=1,NUMV
       RES=RES+ABS(WRK(I))
 10   CONTINUE
      RES=RES/NUMV
C
      RETURN
      END
```

```
      SUBROUTINE HDIFF(DIFF, H,HWRK,NUMV,ID)
      IMPLICIT NONE
      INTEGER ID,I,NUMV
      REAL DIFF,H(ID),HWRK(ID)
C-----------------------------------------------------------------------
C  This subroutine calculates the average difference between
C  H calculated at current property loop iteration and the values
C  calculated at the previous iteration
C
C       DIFF = average difference in enthalpies [kJ/kg]
C       H(I) = enthalpy for control volume I  [kJ/kg]
C       HWRK(I) = enthalpy for cv I from previous iteration [kJ/kg]
C       NUMV = number of control volumes
C       ID = array dimension
C-----------------------------------------------------------------------
C calculating the total difference in enthalpies:
C-----------------------------------------------------------------------
      DIFF=0.0
      DO 10 I=1,NUMV
       DIFF = DIFF + ABS(H(I)-HWRK(I))
      CONTINUE
C-----------------------------------------------------------------------
C calculating the average difference:
C-----------------------------------------------------------------------
      DIFF=DIFF/NUMV

      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE AVG_H(HOCALC, H,MDOT,MTOT,NUMV,ID)
      IMPLICIT NONE
      INTEGER I,ID,NUMV
      REAL H(ID),MDOT(ID),MTOT,HOCALC
C-----------------------------------------------------------------------
      This subroutine checks to see what the average enthalpy
      is at the outlet.

      HOCALC = average enthalpy at bundle exit [kJ/kg]
      H(I) = enthalpy for cv I [kJ/kg]
      MDOT(I) = mass flow in cv I [kg/s]
      MTOT = total channel mass flow [kg/s]
      NUMV = number of control volumes [-]
      ID = array dimension
C-----------------------------------------------------------------------
      HOCALC=0.0
      DO 10 I=1,NUMV
       HOCALC=HOCALC+MDOT(I)*H(I)
10    CONTINUE

      HOCALC=HOCALC/MTOT
      RETURN
      END
```

```
      SUBROUTINE FACTAR(HFCTR,HFCTRI, MDOT,QFUEL,HIN,BUNDLE,BUNI,ID)
      IMPLICIT NONE
      INTEGER I,ID,BUNDLE,BUNI
      REAL HFCTR(ID),HFCTRI(ID),MDOT(ID),QFUEL(ID),HIN(ID),
     :     MDOT1,MDOT2,Q1,Q2
C-----------------------------------------------------------------------
      This subroutine calculates the enthalpy at the bundle exit
      using 'FACTAR PARTIAL MIXING' methodology.
      HFCTR(I)  = enthalpy at bundle outlet using FACTAR Partial Mixing
                  (two control volumes only) [kJ/kg]
      HFCTRI(I) = enthalpy of fluid entering bundle (for FACTAR
                  control volumes) [kJ/kg]
      MDOT(I)   = mass flow in control volume I [kg/s]
      QFUEL(I)  = heat transfer from fuel for control volume I [kJ/s]
      HIN(I)    = enthalpy of fluid entering bundle using new mixing
                  model [kJ/kg]
      MDOT1     = mass flow in inner portion of bundle [kg/s]
      MDOT2     = mass flow in outer flow annulus [kg/s]
      Q1        = fuel power in inner portion of bundle [kJ/s]
      Q2        = fuel power in outer portion of bundle [kJ/s]
C-----------------------------------------------------------------------
C mass flows for inner CVs
C-----------------------------------------------------------------------
      MDOT1=0.0
      Q1=0.0
      DO 10 I=1,10
        MDOT1=MDOT1+MDOT(I)
        Q1=Q1+QFUEL(I)
10    CONTINUE
C-----------------------------------------------------------------------
C mass flows for outer CVs
C-----------------------------------------------------------------------
      MDOT2=0.0
      Q2=0.0
      DO 20 I=11,14
        MDOT2=MDOT2+MDOT(I)
        Q2=Q2+QFUEL(I)
20    CONTINUE
C-----------------------------------------------------------------------
C for first bundle
C-----------------------------------------------------------------------
      IF (BUNDLE.EQ.BUNI) THEN
C-----------------------------------------------------------------------
C inlet enthalpy for FACTAR cv 1: (inner)
C-----------------------------------------------------------------------
        HFCTRI(1)=0.0
        DO 30 I=1,10
          HFCTRI(1)=HFCTRI(1)+MDOT(I)*HIN(I)
30      CONTINUE
        HFCTRI(1)=HFCTRI(1)/MDOT1
C-----------------------------------------------------------------------
C inlet enthalpy for FACTAR cv 2: (outer)
C-----------------------------------------------------------------------
        HFCTRI(2)=0.0
        DO 40 I=11,14
          HFCTRI(2)=HFCTRI(2)+MDOT(I)*HIN(I)
40      CONTINUE
        HFCTRI(2)=HFCTRI(2)/MDOT2
      ENDIF
C-----------------------------------------------------------------------
C calculate the outlet enthalpy for each FACTAR control volume:
C-----------------------------------------------------------------------
      HFCTR(1)=HFCTRI(1)+1000.0*Q1/MDOT1
      HFCTR(2)=HFCTRI(2)+1000.0*Q2/MDOT2
C-----------------------------------------------------------------------
C  reset the inlet enthalpy to oulet enthalpy from last bundle
C-----------------------------------------------------------------------
      HFCTRI(1)=HFCTR(1)
      HFCTRI(2)=HFCTR(2)

      RETURN
      END

C-----------------------------------------------------------------------
      SUBROUTINE ANALYTICAL(HNMX,HMIX, HICALC,MDOT,QFUEL,NUMV,ID)
      IMPLICIT NONE
      INTEGER I,ID,NUMV
      REAL HNMX(ID),HMIX(ID),HICALC,MDOT(ID),QFUEL(ID),DEL
C-----------------------------------------------------------------------

C  This subroutine calculates the enthalpy at the bundle outlet
C  using two different analytical approaches.  These are:

C           1) Mixing at endplates (HMIX(I))
```

```
C              2) No mixing at all (HNMIX(I))
C
C     HNMX(I) = calculated enthalpy if there were no mixing at all [kJ/kg]
C     HMIX(I) = calculated enthalpy for complete mixing at endplates [kJ/kg]
C     HICALC = average enthalpy at inlet [kJ/kg]
C     MDOT(I) = mass flow through control volume I [kg/s]
C     QFUEL(I) = fuel power to cv I [kJ/s]
C     NUMV = number of control volumes
C     ID = array dimension
C
C-----------------------------------------------------------------------
       DO 10 I=1,NUMV
         DEL=QFUEL(I)/MDOT(I)
         HMIX(I)=HICALC + 1000.0*DEL
         HNMX(I)=HNMX(I) + 1000.0*DEL
 10    CONTINUE
C
       RETURN
       END
```

```
   SUBROUTINE OUTDAT(APP,BP,MDOT,QFUEL,WPG,
  :           DIFF,AREAG,DG,X,H,HMIX,HNMX,HOCALC,HFCTR,
  :           BUNDLE,ITER,NUMG,NUMV,ID,KD)
   IMPLICIT NONE
   INTEGER I,K,NUMV,ID,NUMG,KD,BUNDLE,ITER
   REAL APP(ID),BP(ID),MDOT(ID),QFUEL(ID),WPG(KD)
   REAL AREAG(KD),DG(KD),X,H(ID),HOCALC,HMIX(ID)
   REAL HNMX(ID),HFCTR(ID),QLOSS(ID),DIFF
-------------------------------------------------------------------
convergence data
-------------------------------------------------------------------
   write(6,49)  BUNDLE,ITER,DIFF
   write(11,49) BUNDLE,ITER,DIFF
   FORMAT('BUNDLE= ',I2,2X,'ITER= ',I4,2X,'DIFF= ',E10.4)
-------------------------------------------------------------------
other data
-------------------------------------------------------------------
   CALL OUT1D(APP,   'APP      ',11,1,NUMV,1,ID)
   CALL OUT1D(BP,    'BP       ',11,1,NUMV,1,ID)
   CALL OUT1D(MDOT,  'MDOT     ',11,1,NUMV,1,ID)
   CALL OUT1D(QFUEL,'QFUEL    ',11,1,NUMV,1,ID)
   CALL OUT1D(WPG,   'WPG      ',11,1,NUMV,1,KD)
   CALL OUT1D(DG,    'DG       ',11,1,NUMV,1,KD)
        write(12,1001) x,(H(I),I=1,NUMV),HOCALC
        write(14,1001) x,(HMIX(I),I=1,NUMV),HOCALC
        write(15,1001) x,(HNMX(I),I=1,NUMV),HOCALC
        write(16,1002) x,(HFCTR(I),I=1,2)
001 FORMAT(F6.1,1X,15(F8.1))
002 FORMAT(F6.1,1X,2(F12.3))

   RETURN
   END
```

Appendix C

E-mail Correspondence Concerning

ASSERT Limitations Under Low Flow