Solute Mixing in a Fracture Junction Under Equal and Unequal Flow Conditions

J. Sidney Wise

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Hydrology

Department of Earth and Environmental Science
New Mexico Institute of Mining and Technology
Socorro, New Mexico, USA

May, 1999

Abstract

I conducted a series of experiments using a physical model of a fracture junction with simple geometry. The flow conditions in the experiments included equal flow, where the flow rates in all four fracture branches were identical; unequal flow, where the flow rates in the two inflow fractures were different but the flow rates in adjacent inflow and outflow branches were identical; and forced mixing (a type of unequal flow), where water from one of the inflow branches crossed to the opposite side of the junction and mixed with water from the other inflow branch. Forced mixing occurred due to the fact that the flow rate in one inflow branch was greater than the flow rate in the adjacent outflow branch, resulting in overflow to the opposite outflow branch. The goal of the experiments was to determine how the mixing behavior of a solute would be effected by the Peclet number at the junction in each of these flow conditions.

In the case of equal flow, the present work verifies the findings of Li (1995), showing that partial mixing occurs at Peclet numbers between approximately 1 and 200. Complete mixing occurs below this range, and streamline routing occurs above this range. Photomicrographs of the equal flow case illustrate the three types of mixing behavior and show that upstream diffusion occurs at low Peclet numbers. In the case of unequal flow without forced mixing, where the water from each inflow branch exits through the adjacent outflow branch, the transition between mixing states occurs at a lower range of Peclet numbers than in equal flow conditions, so that complete mixing was not observed at a Peclet number of 1. In the case of forced mixing, the transition occurs at still lower Peclet numbers, too low to be observed in the present work.

Acknowledgements

I would like first of all to thank my advisor, John L. Wilson, who is an inexhaustible source of ideas and encouragement. I would also like to thank Rob Bowman and Steve Schery for serving as members of my thesis committee. Rob's down-to-earth clarity provides a necessary complement to John's and my intellectual ramblings.

Bob Reedy got this project started. He and John Wilson did the original design work, giving me a head start.

I'm grateful to Mike Whitworth, who was my research advisor on a previous project, for introducing me to the world of graduate school and scientific research.

The departmental staff, especially Edith Montoya, Liz Whitworth, Patci Mills, and Felix Berreras, have gone out of their way to help me on numerous occassions. The secretaries regularly deal with things that are completely incomprehensible to me, and Felix has provided essential advice and equipment in the lab.

Mary Watson-Finley and Dave Johnson have often taken time to explain the policies that not even professors understand. Without their patience, it would be difficult to find my way through graduate school.

I'd also like to thank the other students, especially Roseanna Nuepauer, Steve Roy, and Alison Peck, who have given me support -- both moral and academic -- every day that I've been here.

This work was funded by a grant from the National Science Foundation, Hydrological Sciences Program, project EAR-9405-837.

Contents

Chapter 1: Introduction and Background

An important issue in modeling contaminant transport in fractured rock is the question of how solutes travel through fracture junctions under laminar flow conditions. Numerous writers have described the nature and history of the problem, including Berkowitz et al. (1994), Robinson and Gale (1990), Li (1995), and Sahimi (1995). Figure 1.1 shows the idealized two-dimensional fracture geometry that typically appears in fracture junction studies. Fluid enters through two branches and exits through the other two. The present work considers only "continuous" fracture junctions, in which the two inflow branches are adjacent.

The basic question is how a solute entering the junction through one of the inflow branches will be distributed in the outflow branches. Three possible answers to this question are contained in competing models: the complete mixing model, the streamline routing model, and the partial mixing model. Previous experimental and theoretical

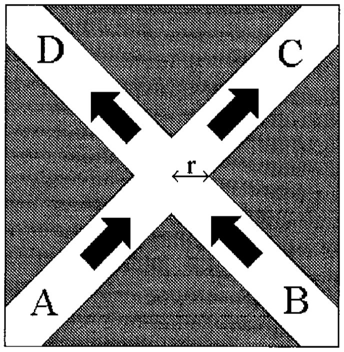


Figure 1.1: The fracture geometry, showing the junction radius *r*, the directions of flow, and the labelling conventions (A, B, C, and D) for the four branches.

work, to be described below, has supported each of the three models.

According to the complete mixing model, diffusion dominates the transport of a solute. Due to the random motion of molecules, the solute is evenly distributed throughout the junction. As a result, the concentrations of the solute in the two outflow branches are identical. This behavior can be described by a simple mixing equation

$$C_{out} = \frac{C_A Q_A + C_B Q_B}{Q_A + Q_B} \tag{1}$$

where C_{out} is the outflow concentration, C_A and C_B are the inflow concentrations, and Q_A and Q_B are the volumetric flow rates in the the inflow branches. The subscripts A and B refer to the labels in Figure 1.1. If we replace Q_A and Q_B with volumes instead of flow rates, then we see that the model is analogous to mixing known volumes of fluid with different solute concentrations. For example, adding one can of orange juice concentrate to three cans of water produces juice with one fourth of the original concentration.

According to the streamline routing model, the behavior of a solute molecule depends only on its initial position. In other words, there is no random motion. A streamline separates the flow entering from branches A and B. In the special case of equal flow in all four branches, the discharge from Branch A goes to Branch D, while discharge from Branch B goes to Branch C. Since there is no mixing across the dividing streamline in the streamline routing model, $C_D = C_A$ and $C_C = C_B$. The same can be true when Q_A and Q_B are different, as long as $Q_A = Q_D$ and $Q_B = Q_C$. Where this symmetry does not occur, Hull and Koslow (1986) reasoned that the outflow concentrations could still be determined using the assumption that solute does not cross streamlines. Then the fluid in each outflow branch may be viewed as a mixture of fluid (and solute) from the two inflow branches, as determined by the streamlines. This is referred to as "forced mixing." Modifying the equation above, we find that

$$C_C = \frac{C_A Q_{AC} + C_B Q_{BC}}{Q_{AC} + Q_{BC}} \qquad (2)$$

where Q_{AC} is the contribution from Branch A to flow in Branch C, and Q_{BC} is the contribution of Branch B to the flow in Branch C. Note that Q_{AC} is zero if $Q_B \geq Q_C$. In other words, a given outflow branch always receives a contribution from the adjacent inflow branch, but it may or may not receive a contribution from the opposite branch. Similarly,

$$C_D = \frac{C_A Q_{AD} + C_B Q_{BD}}{Q_{AD} + Q_{BD}} \qquad (3)$$

and $Q_{BD} = 0$ if $Q_A \geq Q_D$.

Hull and Koslow (1986) described forced mixing (with streamline routing) in terms of two pairs of equations for different flow conditions. If $Q_B \geq Q_C$, then

$$C_C = C_B \qquad (4)$$

and

$$C_D = \frac{C_A Q_A + C_B (Q_B - Q_C)}{Q_D} \qquad (5)$$

If $Q_A \geq Q_D$, then equation (3) is rewritten as

$$C_D = C_A \qquad (6)$$

and

$$C_C = \frac{C_B Q_B + C_A (Q_A - Q_D)}{Q_C} \qquad (7)$$

Note that if $Q_A = Q_D$ and $Q_B = Q_C$, then either pair of equations reduces to $C_C = C_B$ and $C_D = C_A$.

The third model is the partial mixing model, in which the first two models both apply, but under different limiting conditions. Complete mixing occurs when the influence of diffusion is large, and streamline routing occurs when the influence of advection is large. Partial mixing occurs as a transition between complete mixing and streamline routing. The partial mixing model hinges on the idea that the amount of diffusive mixing depends on the "junction residence time," or the time it takes for a molecule of solute to pass through the junction. The longer the solute residence time in the junction, the more time it will have to diffuse laterally, across the streamlines.

The Peclet number, which has various forms in different applications, is a dimensionless number describing the ratio between advective and diffusive influences. It applies conveniently to the partial mixing model in order to describe the flow conditions under which the three mixing phenomena will occur. Berkowitz et al. (1994) proposed a fracture junction Peclet number defined as

$$P_e = \frac{vr}{D} \tag{8}$$

where v is the average fluid velocity in the fractures, r is the fracture junction radius (Figure 1.1), and D is the diffusion coefficient of the solute. A low Pe indicates a long junction residence time (relative to the size of the junction and the diffusion coefficient), while a high Pe indicates a short junction residence time. According to the partial mixing model, complete mixing will occur at low values of P_e, while streamline routing will occur at high values of P_e. Partial mixing will occur in the middle range.

Authors have disagreed over which of the three models is most appropriate. The earliest experimental work (Krizek et al., 1972; Castillo, 1972) indicated that complete mixing was the appropriate model, but this work is generally considered to have been in error (see, for example, Robinson and Gale 1990). Subsequent experiments (Wilson and Witherspoon 1976, Hull and Koslow 1986, Robinson and Gale 1990) repeatedly demonstrated streamline routing. More recent experimental and numerical studies (Hull et al. 1987, Berkowitz et al. 1994, Li 1995, Stockman et al. 1996) have indicated that the partial mixing model is applicable.

The proponents of the partial mixing model have disagreed over how the model should be applied. Analytical models by Hull et al. (1987) implied that complete mixing occurs at Peclet numbers below 1, streamline routing occurs at Peclet numbers above 235, and a transition zone of partial mixing exists between those values. Numerical results by Berkowitz et al. (1994) indicated that streamline routing occurs above $P_e = 1$, partial mixing occurs below $P_e = 1$, and complete mixing does not occur, even at very low values of P_e. Li (1995) and Stockman et al. (1996) stated that the transition from complete mixing to streamline routing occurs between $P_e = 1$ and $P_e = 200$.

It is interesting to compare these ranges to the experiments reported in the literature. According to all three of the ranges quoted above, the experiments of Wilson and Witherspoon (1976), Hull and Koslow (1986), and Robinson and Gale (1990) were all conducted at Peclet numbers well above the transition zone, which explains why only streamline routing occured in all three studies.

Hull et al. (1987) conducted an experiment in which partial mixing (in the transition zone of P_e values) occurred. Their fracture network included many junctions, so a single Peclet number cannot be given for the experiment. However, a calculation based on the average flow rate indicates a mean P_e of 223 in the ambient flow field and 366 during sample injection. Because the exact range of P_e in the experiment is unknown, it is difficult to draw specific conclusions regarding the nature of the transition zone in their experiments.

Li (1995) and Stockman et al. (1996) conducted both experimental and numerical studies that supported their hypothesis of a transition range from Pe = 1 to Pe = 200. Their work effectively supported the similar range suggested by Hull et al. (1987).

The majority of the previous experiments were conducted with equal flow rates in the four branches of the fracture junction. This includes Li's (1995) experiments, which to date have been the only single-junction, low-P_e work. Li's proof of the applicability of the partial mixing model to equal flow rate conditions leaves one wondering if the same model would apply to conditions of unequal flow rates. In addition, it is useful to repeat and verify Li's equal-flow experiments, especially if this can be accomplished under more controlled conditions than in his original study.

The purpose of the present work is first to test Li's hypothesis using improved techniques, then to apply the same techniques to conditions of unequal flow rates. The working hypothesis is that under both equal and unequal flow conditions, the complete mixing rule will apply at $P_e = 1$ and below, the streamline routing rule will apply at $P_e = 200$ and above, and the partial mixing phenomenon will provide a smooth transition between these states.

Chapter 2: Experimental Methods

The experiments consisted of controlling (and varying) the flow velocities in a fracture junction and observing the behavior of solutes as they passed through it. To observe the solutes, I used two methods: microscopy and chemical analysis. Unfortunately, I could not use both methods simultaneously. The experiments are therefore divided into two categories: qualitative experiments, using microscopic visualization, and quantitative experiments, using chemical analysis. In all of the experiments, the fracture junction was "continuous." In other words, the two inflow branches were always adjacent to each other. In addition, the flow was always laminar.

Figure 2.1 shows the physical model of a fracture junction. It was custom built by Coronado Machine of Albuquerque, New Mexico. The model is made of transparent acrylic and measures 15cm (6 inches) square. The fracture junction (Figure 2.2), which is at the center of the model shown in Figure 2.1, consists of two inflow branches and two outflow branches, intersecting at right angles. Each fracture is 0.5mm wide and 10mm deep, and the fracture walls are smooth and impermeable.

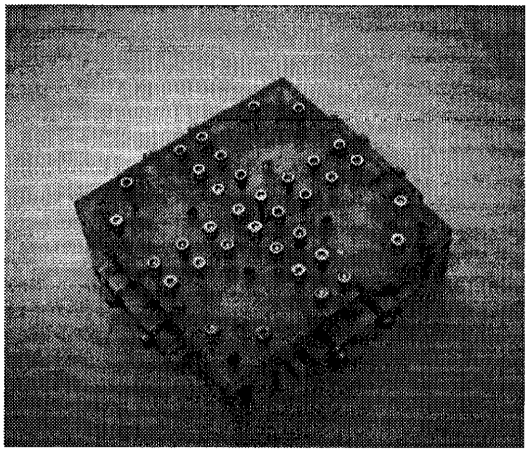


Figure 2.1: The fracture junction model. The model is made of transparent acrylic and is 15cm (6 inches) on a side.

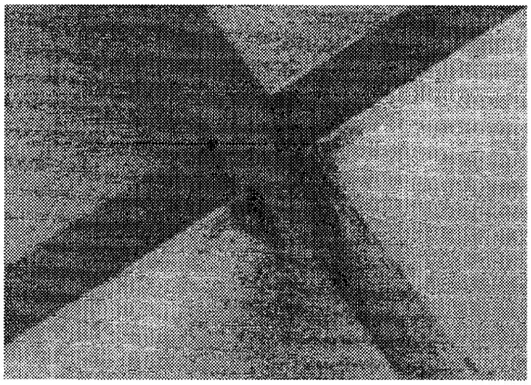


Figure 2.2: Photomicrograph of the fracture junction, which is within the model shown in figure 2.1. Each fracture is 0.5mm wide.

The machinist assembled the model from four separate blocks. In other words, each branch of the fracture junction is essentially the gap between two blocks of acrylic. This allowed precise control over the fracture geometry during construction.

After construction, I checked the geometry of the junction under a microscope to ensure that the branches were straight and perpendicular, and I measured the apertures of the fracture branches at the junction (see appendices for the method of measurement). Two of the branches, on opposite sides, had apertures of 0.50mm ±0.01mm. The other two branches measured 0.53mm ±0.01mm. When calculating the Peclet number, I assumed that the branches had an aperture of 0.52mm, with a junction radius r = 0.37mm.

I mounted the model vertically, with the water flowing upward, to prevent stratification of solute concentrations. The orientation of the model matched the orientation shown in Figures 2.3 and 2.4.

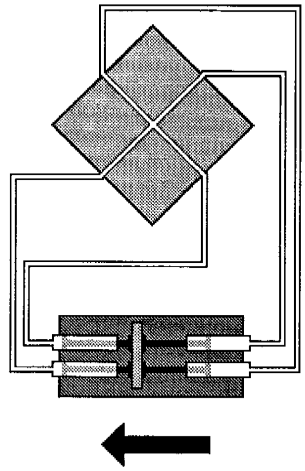


Figure 2.3: A simplified diagram of the four-syringe pump and the directions of flow.

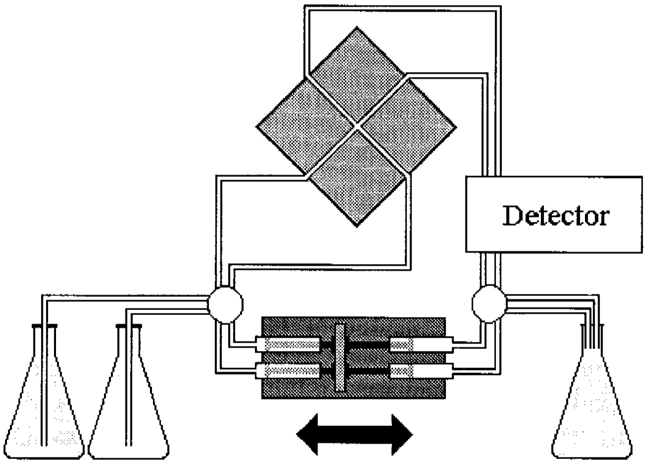


Figure 2.4: A complete diagram, showing the supply reservoirs (left), waste reservoir (right), the valves (circles), and the detector (a UV / Visible spectrophotometer).

A small, push-pull syringe pump, manufactured by KD Scientific (no model number), controlled flow in the model (Figure 2.3). The pump held four syringes, with two syringes injecting and two syringes refilling simultaneously. Each of the four syringes controlled the flow in one of the branches of the model. I tested the pump for two flow rates, 2.0 μl/min ($P_e = 1$) and 200 μl/min ($P_e = 100$), using a stopwatch and four graduated cylinders. The flow rate in all four syringes was accurate to 0.005%.

In order to refill the pump, the setup needed to be slightly more complicated than that shown in Figure 2.3. In the actual setup, as shown in Figure 2.4, the pump was connected through two valves (Hamilton, Serial MVP) to fluid reservoirs (Erlenmeyer flasks). The valves alternately connected the pump to the model or the reservoirs.

An external computer, a PC, controlled the pump and valves through serial communication lines. The computer set the pumping rate and refilled the pump automatically. This allowed experiments to run with only minimal human supervision. The same computer also monitored the temperature in the experimental apparatus.

The two supply reservoirs, shown on the left in Figure 2.4, sat on air-powered stirring plates. The flasks were sealed and connected to a vacuum line. Each time I added a new solution to the supply reservoirs, I degassed the solutions by stirring them under a vacuum. I then let them return to normal pressure to avoid evaporation.

Figure 2.4 also shows the location of the detector that monitored solute concentrations in the outflow branches. This detector was a Hewlett-Packard HP 8453 UV/visible spectrophotometer. The spectrophotometer held two flow-through cuvets (one for each outflow branch; Hellma, type 178.711-QS), along with five cuvets of standard solutions for calibration. The instrument recalibrated itself with the standards before each measurement of outflow concentrations.

A second computer, also a PC, operated the detector through an HPIB communication line. The computer also calculated solute concentrations and recorded data. The two computers communicated through a serial interface, in order to coordinate the measurements with the activity of the pump and valves. The first computer (the one controlling the pump and valves) signalled the second computer (the one controlling the detector) to make a measurement of the concentrations. The timing of this signal depended on the rate of flow, so that a certain volume of fluid would pass through the detector between measurements. This volume depended on the syringe size, but it was constant for each experimental run. The first computer also informed the second computer of the flow rate and the temperature, which were necessary for calculating the Peclet number.

The computer that operated the pump and valves used the LabVIEW software package, while the computer that operated the detector and analyzed the data ran HPChem and Microsoft Excel. Each of the three software packages allows macro programming (or "VI" programming in the case of LabVIEW), which was necessary for automating the experiments. The HPChem and LabVIEW code that I wrote for these experiments is explained in the appendices.

I used iodide as the chemical tracer in the quantitative experiments. Iodide is easily detected by ultraviolet absorbance at low concentrations, and its relatively high diffusion coefficient is helpful in achieving low Peclet numbers. In each experiment, one syringe injected a "mobile phase" solution of 10mM K_2HPO_4, while the other syringe injected a solution of both 10mM K_2HPO_4 and 0.1mM KI. Because the concentration of the buffer was much higher than the concentration of the tracer, the concentration of K^+ was essentially uniform. This allowed I^- to diffuse freely, without dependence on its counterion.

In order to ensure that the temperature was stable and known, the model, the pump, and the mobile phase reservoirs all resided inside an insulated box. This box and the detector were both inside an insulated cabinet. Two thermistors sat directly on the fracture model, and the first computer continuously logged data from these and other thermistors. I used the observed temperature to calculate the diffusion coefficient before calculating the Peclet number.

For the visualization experiments, I attached a light source to back of the model and mounted a microscope in front of it. The light source was a Schott KL-1500, which used fiber optics to channel light from a remote source. This allowed the hot bulb to be far from the model, although the light itself undoubtedly caused some warming. The Zeiss microscope, shown in Figures 2.5 and 2.6, was mounted on a radial arm. I mounted a video camera (Panasonic Digital 5000) on the microscope for casual observations. Looking at a video monitor was much more convenient than looking through the eyepieces, especially for adjusting the position of the microscope. To record permanent images during experiments, I mounted a 35mm still camera (Zeiss M35) on the microscope in place of the video camera. For the visualization experiments, the inflow branches contained solutions of yellow and green food dye, which was clearly visible to both the video and still cameras.

For several reasons, the visualization experiments were not suitable for specific quantification of the mixing behavior. The first reason is that in order to be visible, the concentration of the food dye had to be well above the detection limits of the spectrophotometer. In other words, I could either see it in high concentrations or measure

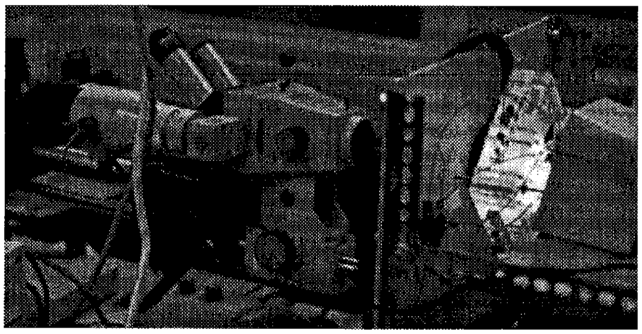


Figure 2.5: The microscope (center) views the model (lighted). The still camera, which is faintly visible at the left edge of the photo, is mounted on the microscope. This picture was taken before the apparatus was installed in the insulated box.

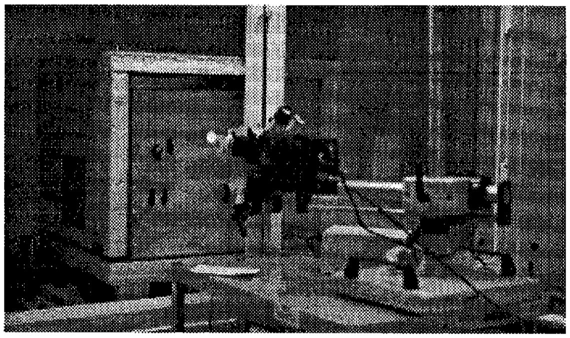


Figure 2.6: The insulated box sits inside the insulated cabinet. The glass door of the cabinet is open, and the front panel of the box has been removed. The microscope views the model through a hole in the plywood. The video camera, rather than the still camera, is mounted on the microscope. The boom that holds the microscope is in the foreground, at right.

it in low concentrations, but I couldn't see it and measure it simultaneously. Another difficulty arose from the fact that the composition of the food dye is unknown, and it probably consists of multiple components. Thus, no diffusion coefficients could be specified for the dyes. Finally, temperature control was difficult during the visualization experiments, because they necessitated exposing the model at the ambient air temperature and using intense light from the fiber optic source.

In all of the visualization experiments, the flow rates were equal in the four branches, duplicating the conditions used by Li (1995). The first set of quantitative experiments also used equal flow rates. Subsequent quantitative experiments used unequal flow rates.

To run equal flow rates, I mounted four 10ml syringes (Hamilton Gastight #1010) on the pump. For unequal flow rates, I used two of the 10ml syringes and two 2.5ml syringes (Hamilton Gastight #1002). Since the diameters of the two types of syringes differed, using the same plunger velocity in both types provided two different flow rates. Thus, the flow rate in a fracture branch attached to a 10ml syringe was four times the flow rate in a branch attached to a 2.5ml syringe.

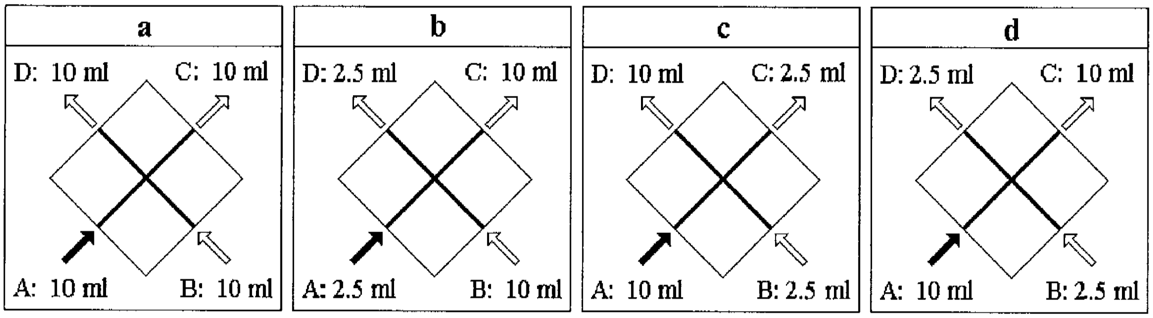


Figure 2.7: The syringe sizes used for the four branches in different experimental setups. The black arrow indicates the branch from which potassium iodide entered the junction. The flow velocity is proportional to the syringe size.

The various syringe setups are shown in Figure 2.7. Note that the sum of the syringe sizes on the two inflow branches (A and B) is always equal to the sum of the syringe sizes on the two outflow branches (C and D), satisfying mass conservation. An additional configuration, which is not shown, used the same syringe setup as in Figure 2.7-I but with the iodide injected from Branch B, rather than Branch A.

In Figure 2.7a, the flow rate is equal in the four branches. This is the flow regime used in the qualitative experiments and in Li's (1995) previous work, in addition to the first set of quantitative experiments described here. Figures 2.7b and 2.7c show cases in which the flow rates in the two inflow branches are unequal, but the flow rates in adjacent inflow and outflow branches (i.e. Branch A and Branch D) are equal. Figure 2.7d represents cross-flow, in which some of the water from Branch A exits through Branch C.

It is important to note that the appendices to this thesis provide much more detailed descriptions of the equipment and software that I used in these experiments. Although this chapter covers the most important aspects of the experiments, the reader who wishes to conduct similar experiments will find much more information in the appendices.

Chapter 3: Results and Discussion

The results are divided into two categories: equal flow experiments and unequal flow experiments. Equal flow experiments are those in which the flow rates in all four fracture branches are identical. This category, which is essentially an expansion of previous work, includes both qualitative and quantitative experiments. The unequal flow experiments, all of which are quantitative, represent a previously unexplored area.

Equal Flow: Qualitative Experiments

The qualitative experiments showed visually, for the first time, that streamline routing, partial mixing, and complete mixing can all occur in a fracture junction. These results are shown in Figure 3.1. Each pair of photographs (e.g., a and b) show the same scene at different magnifications. In all of these experiments, green dye entered through Branch A (lower left in the photographs), while yellow dye entered through Branch B (lower right). The fractures, as described in the methods chapter, are 0.5mm wide. For clarity, the photographs are presented here as black-and-white images. These images show only the red color channel of the original color images, as the red channel showed the greatest contrast between dye colors. In the image shown, white represents the yellow dye, and black represents the green dye. As explained in the methods chapter, the actual Peclet numbers could not be measured during experiments that used food dye.

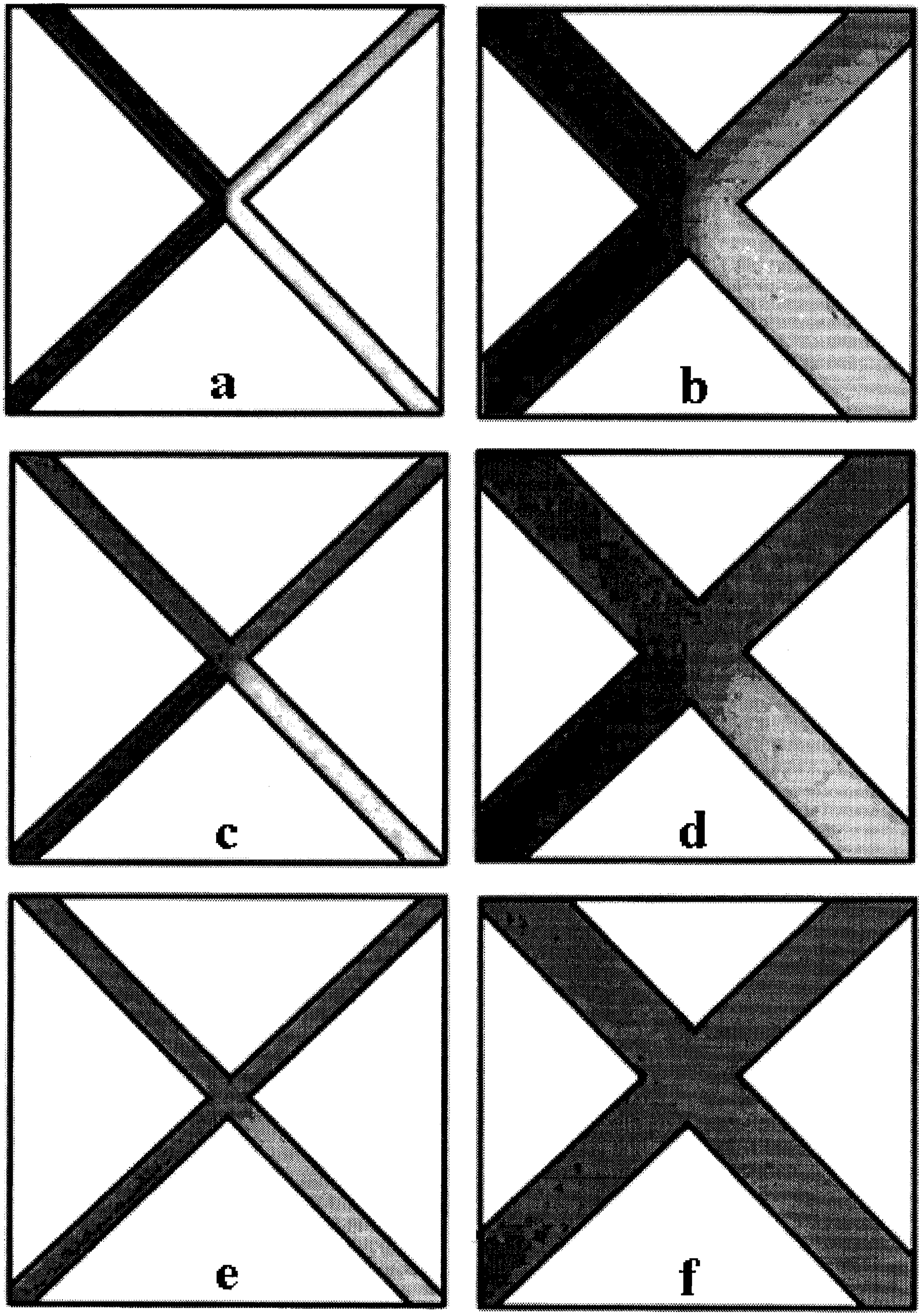


Figure 3.1: Results of the qualitative experiments, processed for clarity. In all photos, the fracture width is 0.5mm. The three photos on the right are closeups of the same situations shown on the left. a & b: high P_e. c & d: middle P_e. e & f: low P_e.

The first two photographs (Figure 3.1a,b) show the junction at a high Peclet number. The two outflow branches (at the top) are distinctly different shades, and a dividing line between black and white is visible in the junction. A small amount of mixing is occurring, but the dye is largely following the path of the water. Although the observation of slight mixing does not exactly support the hypothesis that streamline routing would occur, the observation is consistent with previous work by Li (1995) and Stockman et al. (1997). Those workers observed an asymptotic approach to streamline routing at high Peclet numbers, and they considered the small amount of mixing to be negligible.

The next two photographs (Figure 3.1c,d) show the junction at a lower Peclet number than in the previous images. Here, the two outflow branches are more similar in shade, though not yet identical. This indicates that partial mixing is occurring in the junction, which is consistent with the hypothesis.

Figure 3.1e,f show the junction at a very low Peclet number, at which complete mixing is occurring. The two outflow branches are now the same shade of grey. In the original color photos, the outflow branches were yellow-green. This observation is consistent with the hypothesis that complete mixing would occur at low Peclet numbers.

The water velocity in Figure 3.1e,f is slow enough that a significant amount of dye is diffusing upstream. Note that the black and white inflow branches both gradually turn grey as they approach the junction. In the close-up (Figure 3.1f), all four branches are grey. This upstream diffusion has an interesting implication with regard to previous work by Berkowitz et al. (1994) and Stockman et al. (1997). Those two papers, which presented the results of computer simulations, were in disagreement regarding the degree of mixing that would occur at low Peclet numbers. Stockman et al. (1997) suggested that the discrepency was due to a difference in boundary conditions, specifically the lengths of the fracture branches outside of the junction in comparison to their widths. Stockman et al. asserted that long inflow branches were necessary to produce a realistic model at low Peclet numbers. Because the present results show a great deal of upstream diffusion, it appears that Stockman et al. were correct in using relatively long branches in their simulations.

Equal Flow: Quantitative Experiments

The equal flow quantitative experiments demonstrated the same behavior as the qualitative experiments. Figure 3.2 shows 810 measured concentrations of iodide in the

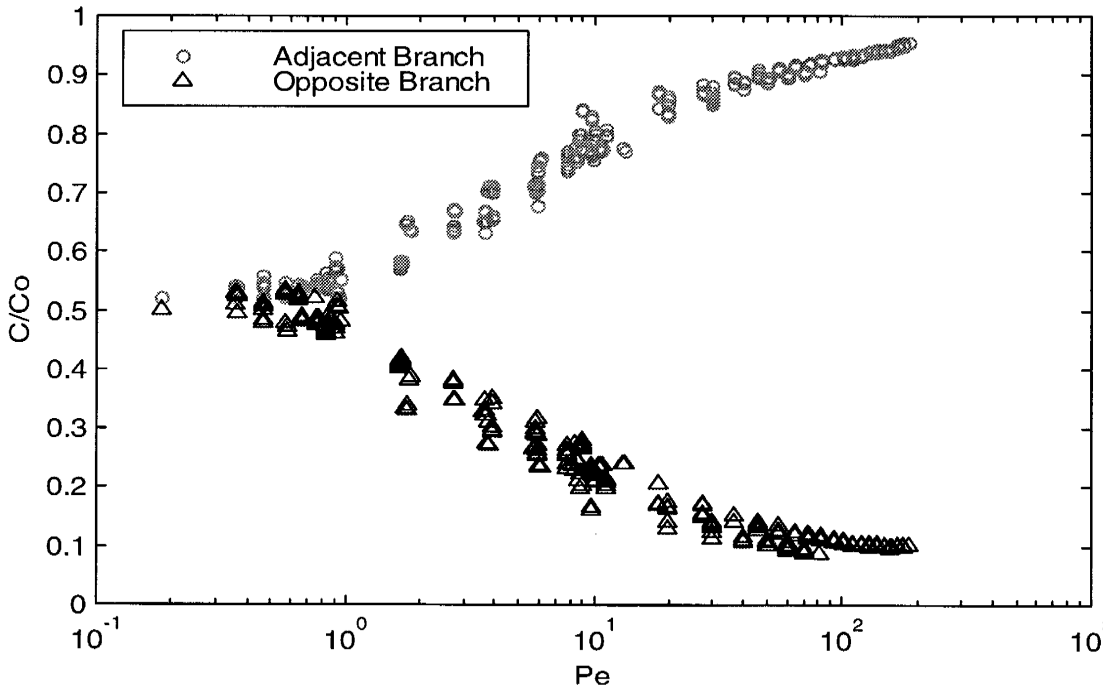


Figure 3.2: The iodide concentration (dimensionless) in the two outflow branches as a function of the junction Peclet number, under equal flow rate conditions.

two outflow branches. The concentrations are in a dimensionless form, as the ratio of outflow concentration to inflow concentration (C / C_0). At Peclet numbers below one, the solute is divided equally between the two branches, which means that complete mixing is occurring. As the Peclet number increases, iodide tends to exit through the branch adjacent to where it entered, and less iodide crosses the junction to exit through the opposite branch. In other words, the iodide has more of a tendency to follow streamlines and less of a tendency to diffuse laterally. When the Peclet number is in the neighborhood of 100, the concentrations asymptotically approach conditions approximating streamline routing. Like the qualitative experiments, the quantitative experiments show that a small amount of mixing still occurs, even at the highest observed Peclet number ($P_e = 200$). The concentration in the adjacent branch is approximately 0.9 (dimensionless), while the concentration in the opposite branch is approximately 0.1.

In order to verify that the fracture junction was symmetrical, I ran some experiments with the iodide entering through Branch A and some with the iodide entering through Branch B. Both sets of data are shown in Figure 3.2, so the "adjacent branch" is Branch C in some cases and Branch D in others, depending on whether the iodide is entering through Branch B or Branch A. I found that changing the iodide input branch did not affect the results, which indicates that the model was indeed symmetrical.

The dimensionless form of the reported concentrations is helpful in showing measurement error, because the sum of the dimensionless concentrations in the two outflow branches should always equal one. Figure 3.2 shows 405 pairs of data points for which the sum is less than 1.1 (24 pairs of data points, 6% of the total, exceeded this limit and are not shown). For the points shown, the mean sum of the pairs is 1.02 with a standard deviation of 0.03, indicating no significant bias in the measurements. Since changing the iodide's inflow location between Branch A and Branch B does not affect the results, it appears that the measurement error is not due to a flaw in the physical model.

Figure 3.3 shows a comparison of the "opposite branch" data (the same data that are shown as triangles in Figure 3.2) to Li's (1995) experimental and lattice gas automata (computer modeling) results. For clarity, the figure does not show the individual data points from the present work. Each open circle represents the mean value of multiple points, while the error bars represent two standard deviations above and below the mean. The number of data points included in each mean is printed on the graph.

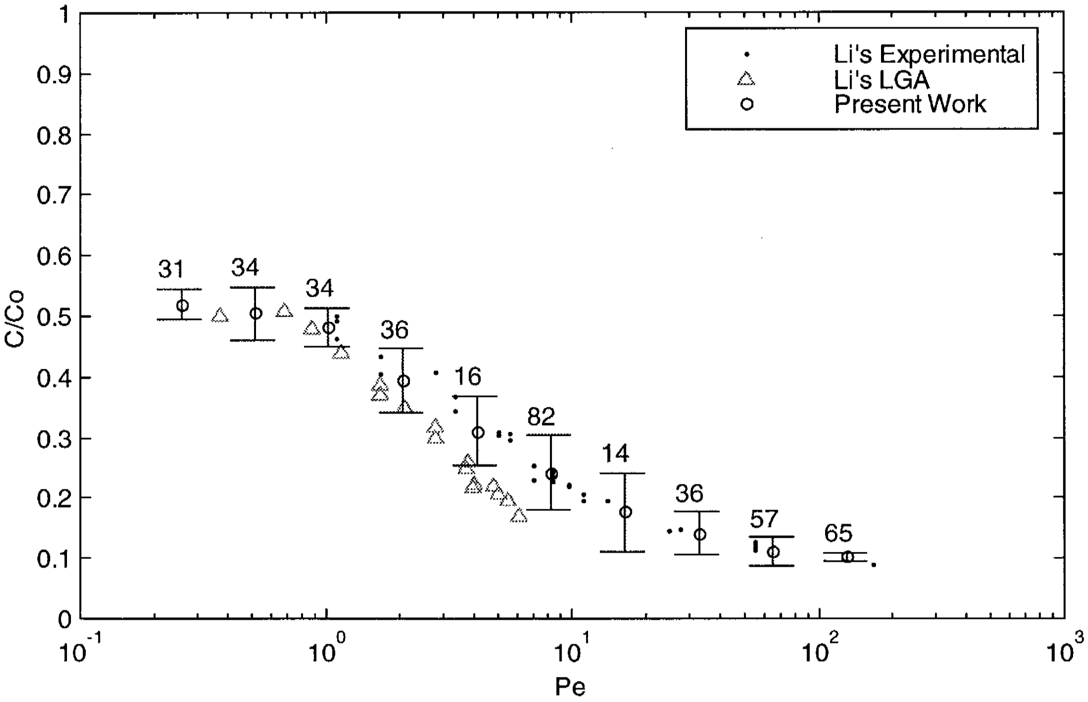


Figure 3.3: Comparison of the "opposite branch" data to Li's (1995) results. The numbers on the error bars indicate the number of measurements represented by each point.

It is clear that the present experimental results agree with Li's experimental results, although neither data set agrees with Li's lattice gas model. Because of an imperfection in Li's physical model, he assumed that the discrepency between model and experiment was due to flawed experimental data. However, the present work indicates that Li's experimental data were correct, since it is unlikely that both experiments would produce the same results otherwise.

Unequal Flow

Figure 3.4 is a repeat of Figure 1.1, to remind the reader of the syringe configurations in the various experiments. I will refer to the syringe configuration shown in Figure 3.4a as the equal flow configuration, while the configurations shown in 3.4b and 3.4c are the first and second unequal flow configurations. Figure 3.4d shows a third unequal flow configuration, which I will refer to as the forced mixing configuration. In all three of the unequal flow configurations, the tracer (iodide) always entered the junction from Branch A.

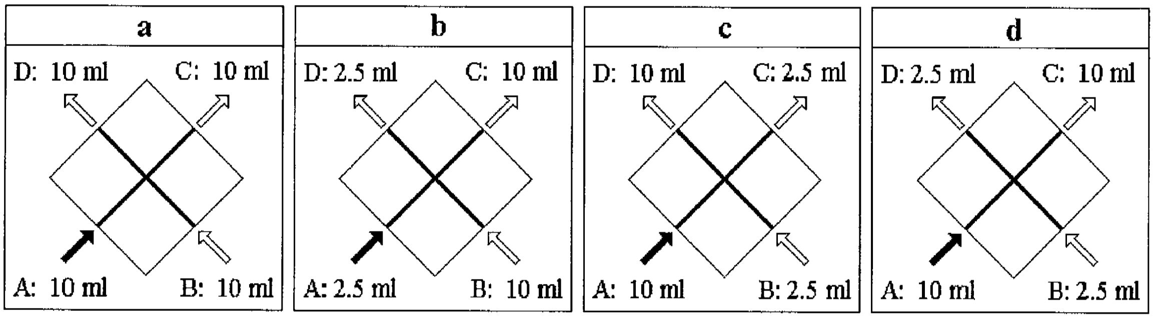


Figure 3.4: Repeat of Figure 2.7, showing the syringe configurations for various experiments. Configurations b, c, and d are the unequal flow configurations.

In the first unequal flow configuration (Figure 3.4b), the flow rates in branches A and D are equal, as are the flow rates in branches B and C. According to the streamline routing model, there would be no mixing in this situation, so the concentration in Branch D would be one, while the concentration in Branch C would be zero. According to the complete mixing model, as described in Chapter 1,

$$C_{out} = \frac{C_A Q_A + C_B Q_B}{Q_A + Q_B} = \frac{Q_A}{Q_A + Q_B} = \frac{1}{5} \tag{3.1}$$

where $C_A = 1$, $C_B = 0$, and $Q_B = 4Q_A$. The complete mixing model predicts an output concentration of 1/5 = 0.2 in both branches. Thus, the present hypothesis predicts that the concentration in Branch C will vary from 0.2 (at low P_e) to 0 (at high P_e), while the concentration in Branch D will vary from 0.2 to 1.

Figure 3.5 shows the experimental results from the first unequal flow configuration (Figure 3.4b). Note that in all the unequal flow data, the "adjacent branch" is Branch D, and the "opposite branch" is Branch C. The junction Peclet number is calculated using the mean flow velocity from the four branches. We see that at high Peclet numbers, the concentrations start to level off at 0.9 and 0.1, as they did in the equal flow experiments. At low Peclet numbers, a trend toward complete mixing is apparent. Complete mixing did not occur, however, even at the minimum Peclet number of 0.6. It appears that this configuration requires a lower Peclet number than the equal flow configuration in order to achieve complete mixing (recall that the Peclet number is based on average velocity). It should also noted that the smooth transition in concentration values over the observed range of Peclet numbers is consistent with the hypothesis stated in Chapter 1.

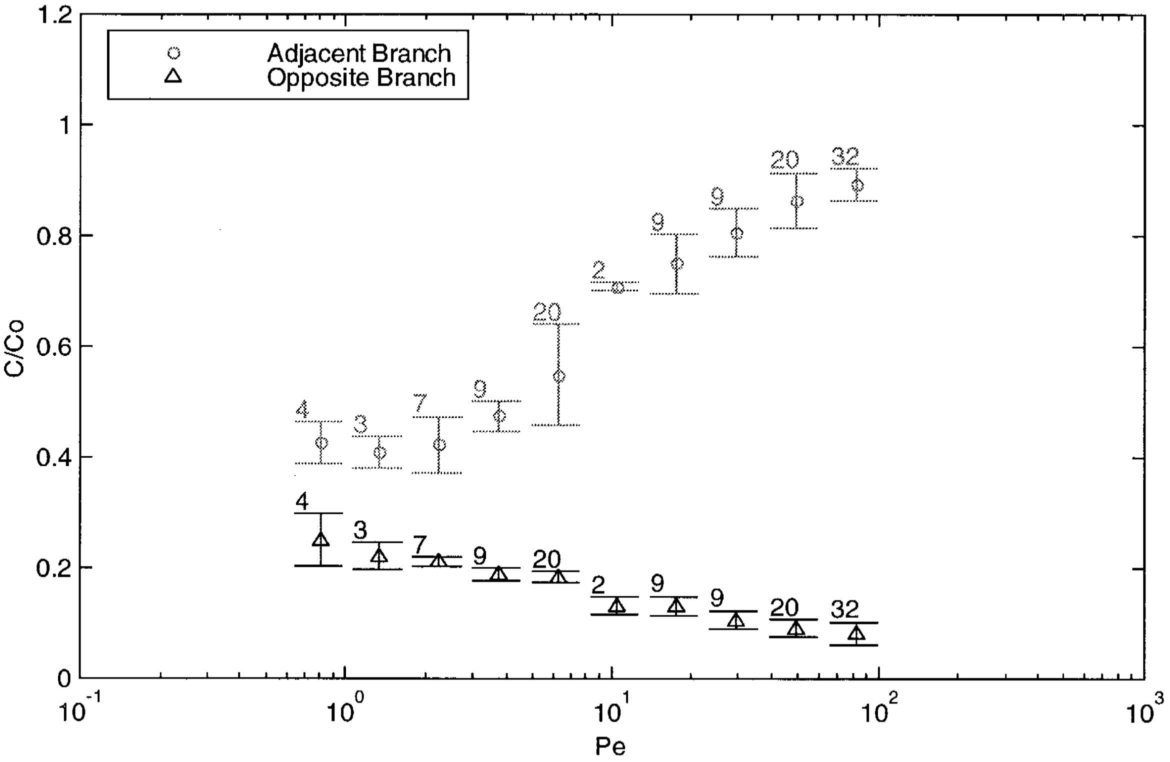


Figure 3.5: Data from configuration b. "Adjacent Branch" is Branch D; "Opposite Branch" is Branch C.

For the second unequal flow configuration (Figure 3.4c), the streamline routing model would again predict that $C_D = 1$ and $C_C = 0$. The complete mixing model would predict that the concentration would be 4/5 = 0.8 in both output branches. As shown in Figure 3.6, there is a trend toward complete mixing at low Peclet numbers and streamline routing at high Peclet numbers. Again, there is a smooth transition between concentration values. As in the data from the first unequal flow configuration, however, the data do not quite converge on the values predicted by either model. Also, high Peclet number concentrations are slightly above one for the adjacent branch, while the concentrations in the opposite branch are approximately 0.2. It is likely that a systematic error in measurement caused all the values to be shifted upward, and that the values should be 0.9 and 0.1, as with the previous configurations. I was unable to find the

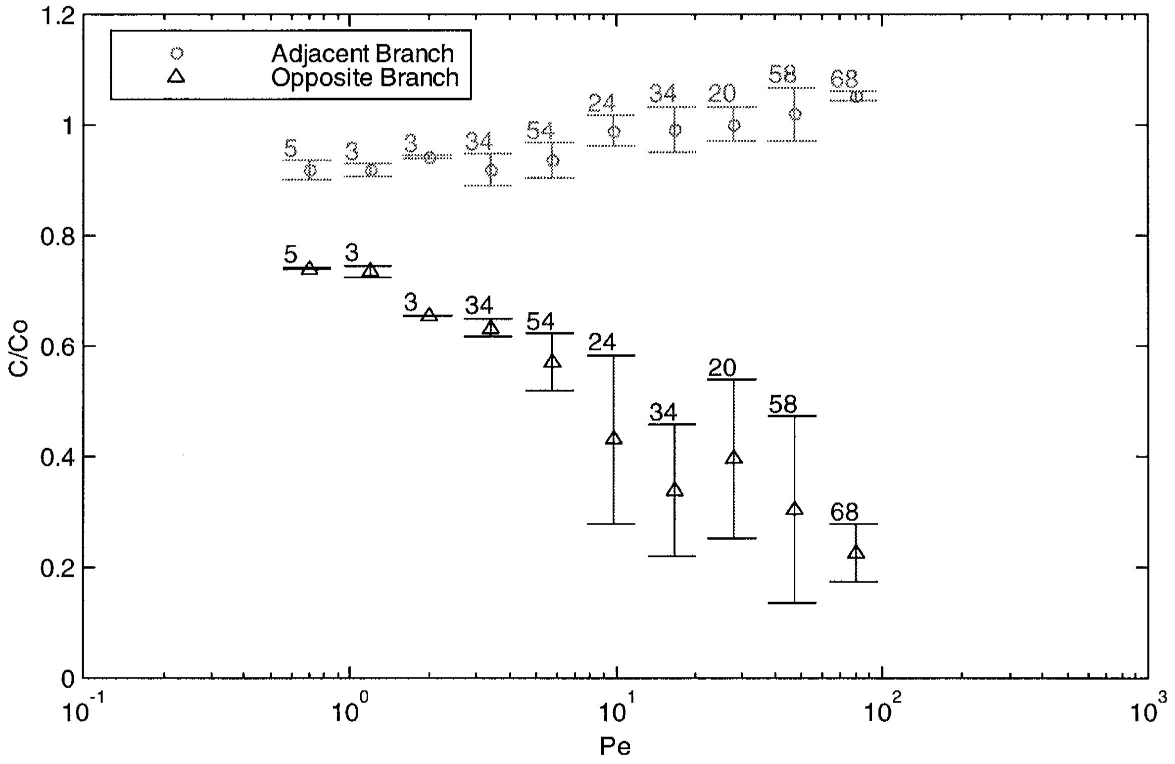


Figure 3.6: Data from configuration c.

source of this apparent error, which has no effect on the trends in the concentration data. The low Peclet number behavior is similar to that in the first unequal flow configuration -- the concentration values approach 0.8, the hypothetical concentration for complete mixing, but they do not reach it. In this case, the lowest Peclet number was 0.5.

The forced mixing configuration is unique in that a portion of the water from Branch A crosses the junction and exits through Branch C (Figure 3.4d). The term "forced mixing" means that water from branches A and B is mixed in Branch C, and the tracer must exit through both outflow branches even under streamline routing conditions. As described in Chapter 1, the streamline routing model predicts that

$$C_D = C_A = 1 \tag{3.2}$$

and

$$C_C = \frac{C_B Q_B + C_A (Q_A - Q_D)}{Q_C} = \frac{(Q_A - Q_D)}{Q_C} = \frac{3}{4} \tag{3.3}$$

where $C_A = 1$, $C_B = 0$, $Q_A = 4Q_B$, $Q_C = 4Q_B$, and $Q_D = Q_B$. The streamline routing model predicts that $C_D = 1$ and $C_C = 0.75$. The complete mixing model, using Equation 1.1, predicts that $C_{out} = 4/5 = 0.8$, the same as the second unequal flow configuration. The present hypothesis predicts that C_D will range from 0.8 at low Pe to 1 at high Pe, and C_C will range from 0.8 to 0.75.

The experimental data for the forced mixing configuration appear in Figure 3.7. Unfortunately, the variation in measurements in Branch C (the opposite branch) makes it difficult to distinguish between the hypothetical end member concentrations of 0.8 and 0.75. In Branch D, the values are generally greater than one, which again indicates a slight measurement error. However, it is clear that the concentrations in the two outflow

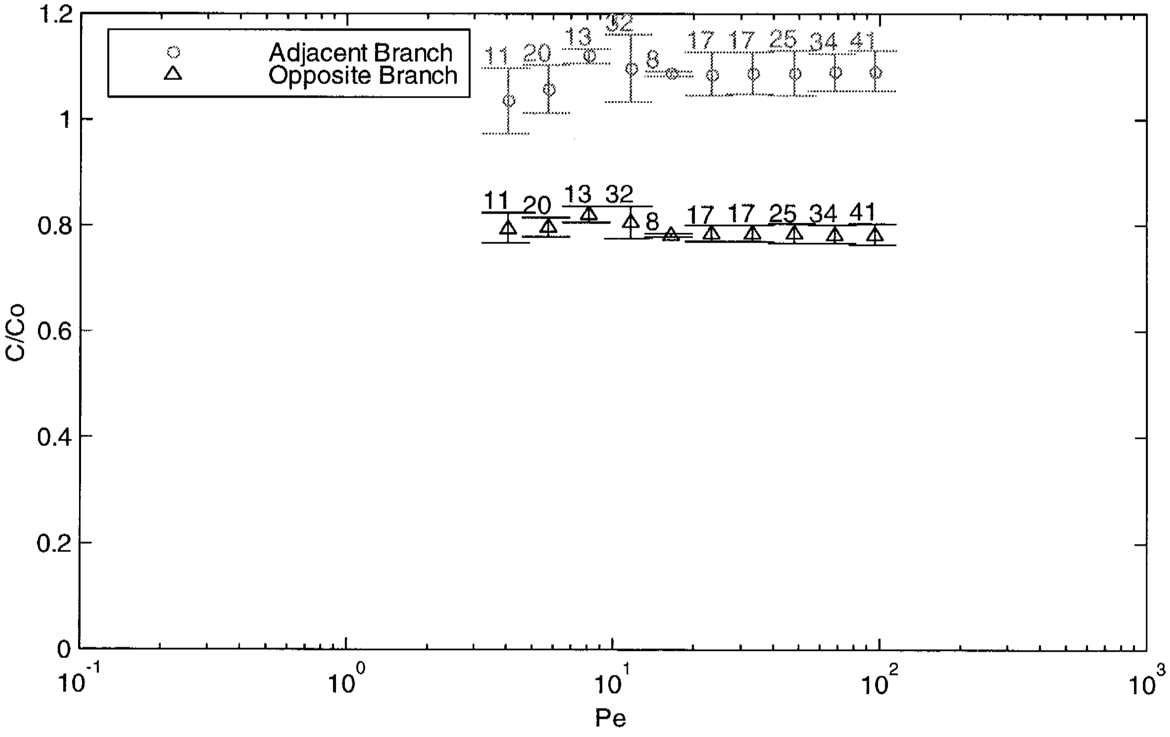


Figure 3.7: Data from the forced mixing configuration.

branches remain distinct, just as they did in the first and second unequal flow configurations. In this case, the lowest Peclet number is 3. The concentrations in the forced mixing configuration are fairly constant over the observed range of Peclet numbers, with the concentrations in the two outflow branches only slightly more similar at low Peclet numbers than at high Peclet numbers. It appears that in this range of P_e, we are seeing mostly streamline routing, with the transition to partial mixing occurring at the lowest observed Peclet numbers. It may be that since tracer molecules from Branch B have a relatively large distance to travel before reaching Branch D, the streamline routing model is valid at much lower Peclet numbers here than in other situations. Since forced mixing is probably the rule in nature, this dominance of streamline routing may be quite significant.

Chapter 4: Conclusions and Recommendations

I observed tracer mixing in a symmetrical fracture junction in four different situations, described here as different syringe configurations. In the equal flow configuration, the flow rate was equal in all four fracture branches. In the unequal flow configurations, the flow rates were different in the two inflow branches, but the flow rate in each outflow branch was equal to that in the adjacent inflow branch. In the forced mixing configuration, a portion of the contribution from Branch A crossed the junction and exited through the opposite branch, resulting in forced mixing of the tracer.

In the equal flow configuration, complete mixing occurred at low Peclet numbers ($P_e < 1$). Partial mixing occurred in the range from approximately $P_e = 1$ to approximately $P_e = 100$. In the range from $P_e = 100$ to $P_e = 200$, the highest Peclet number observed, the dimensionless concentrations were approximately 0.9 in the outflow branch adjacent to where the tracer was introduced and 0.1 in the opposite branch. These observations agree with Li's (1995) experimental data but are slightly different from Li's lattice gas automata simulation. In photomicrographs of equal flow, all three types of behavior are visible. There is considerable upstream diffusion at low Peclet numbers, which supports the assertion of Stockman et al. (1997) that computer simulations of low Peclet number mixing require long inflow branches in order to achieve the proper boundary conditions.

In the unequal flow configurations with no forced mixing, the behavior at high Peclet numbers was similar to that in equal flow. At low Peclet numbers (down to 0.6 and 0.5), the concentrations approached the values predicted by the complete mixing model, but they did not reach those values. In other words, as the Peclet number decreased, the concentrations in the two branches grew more similar but remained distinct. Apparently, the unequal flow configurations require a lower Peclet number for complete mixing than the equal flow configuration.

In the forced mixing configuration, the forced mixing was due to cross-flow from an inflow branch to the opposite outflow branch. In this case, the concentrations were fairly constant with changing Peclet number, remaining near the values predicted by the streamline routing model. The concentrations in the two outflow brânches were slightly more similar at low Peclet numbers (observed down to $P_e = 3$) than at high Peclet numbers, but they were still indicative of streamline routing. Thus, it appears that the streamline routing model is applicable at much lower Peclet numbers when forced mixing is occurring than in the previous situations. Since flow regimes are unlikely to be symmetrical in natural fractures, this behavior in an asymmetrical flow regime is significant.

The conclusions may be summarized as follows:

- Under equal flow conditions, mixing behavior makes a transition from complete mixing to streamline routing between Peclet numbers of approximately 1 and 100.
- Under equal flow conditions, upstream diffusion is significant at low Peclet numbers.
- Under unequal flow conditions without cross-flow or forced mixing, the transition zone occurs at lower Peclet numbers than in the equal

flow case. Complete mixing does not occur at $P_e = 1$ in these conditions.

- Under a different set of unequal flow conditions, where cross-flow and forced mixing occur, the transition zone appears to be lower still. At the lowest Peclet number observed, $P_e = 3$, the mixing behavior is essentially streamline routing.

In future work, it will be important to consider asymmetrical flow regimes in more detail. Since the present work only covers a few very simple situations, there is still much to learn. The following areas of research would produce important new information:

- Repeating the present work at lower Peclet numbers would help to determine the conditions under which complete mixing occurs in unequal flow and forced mixing conditions. Low Peclet numbers present a challenge, because they require small fracture apertures, very slow and accurate flow rates, and stable temperatures over long periods of time.
- Lattice Boltzmann simulations can produce variations of conditions much more conveniently than experimental work. Simulating forced mixing behavior at very low Peclet numbers may reveal important insights, giving experimentalists a better idea of what to cover in future work.
- Experiments or lattice Boltzmann simulations using unequal fracture apertures would also be useful.

References

Berkowitz, B., and L. Smith: Mass transfer at fracture intersections: An evaluation of mixing models. Water Resources Research, 30(6), 1765-1773, 1994

Castillo, R.E.: Dispersion of a contaminant in jointed rock. Ph.D. thesis, Northwestern Universtity, Evanston, Illinois, 1972.

Hull, L.C., and K.N. Koslow: Streamline routing through fracture junctions. Water Resources Research, 22(12), 131-1734, 1986.

Hull, L.C., J.D. Miller, and T.M. Clemo: Laboratory and simulation studies of solute transport in fracture networks. Water Resources Research, 23(8), 1505-1513, 1987.

Krizek, R.J., G.M. Karadi, and E. Socias: Dispersion of a contaminant in fissured rock. Proceedings – Symposium Percolation through Fissured Rock, Stuttgart, 1972.

Li, C.: Low Peclet number mixing behavior at fracture junctions. Ph.D. thesis, 172 pp., New Mexico Institute of Mining and Technology, 1995.

Robinson, J.W., and J.E. Gale: A laboratory and numerical investigation of solute transport in discontinuous fracture systems. Ground Water, 28(1), 25-36, 1990.

Stockman, H.W., C. Li, and J.L. Wilson: A lattice-gas and lattice Boltzmann study of mixing at continuous fracture junctions: Importance of boundary conditions. Geophysical Research Letters, 24(12), 1515-1518, 1997.

Wilson, C.R., and P.A. Witherspoon: Flow interference effects at fracture intersections. Water Resources Research, 12(1), 102-104, 1976.

Wise, J.S, J.L. Wilson, R. Reedy, and C. Li: Automated Control and Measurement in a Fracture Junction Study. Proceedings of the International Symposium on Dynamics of Fluids in Fractured Rocks: Concepts and Recent Advances, Ernest Orlando Lawrence Berkeley National Laboratory, February 10-12, 1999.

Appendix A: Hardware

The most important piece of hardware in these experiment is obviously the fracture junction model, which I described in the methods chapter. A great deal of peripheral plumbing supports the model, as shown in Figure A.1. Figure 2.4, in the methods chapter, shows a simplified version of this.

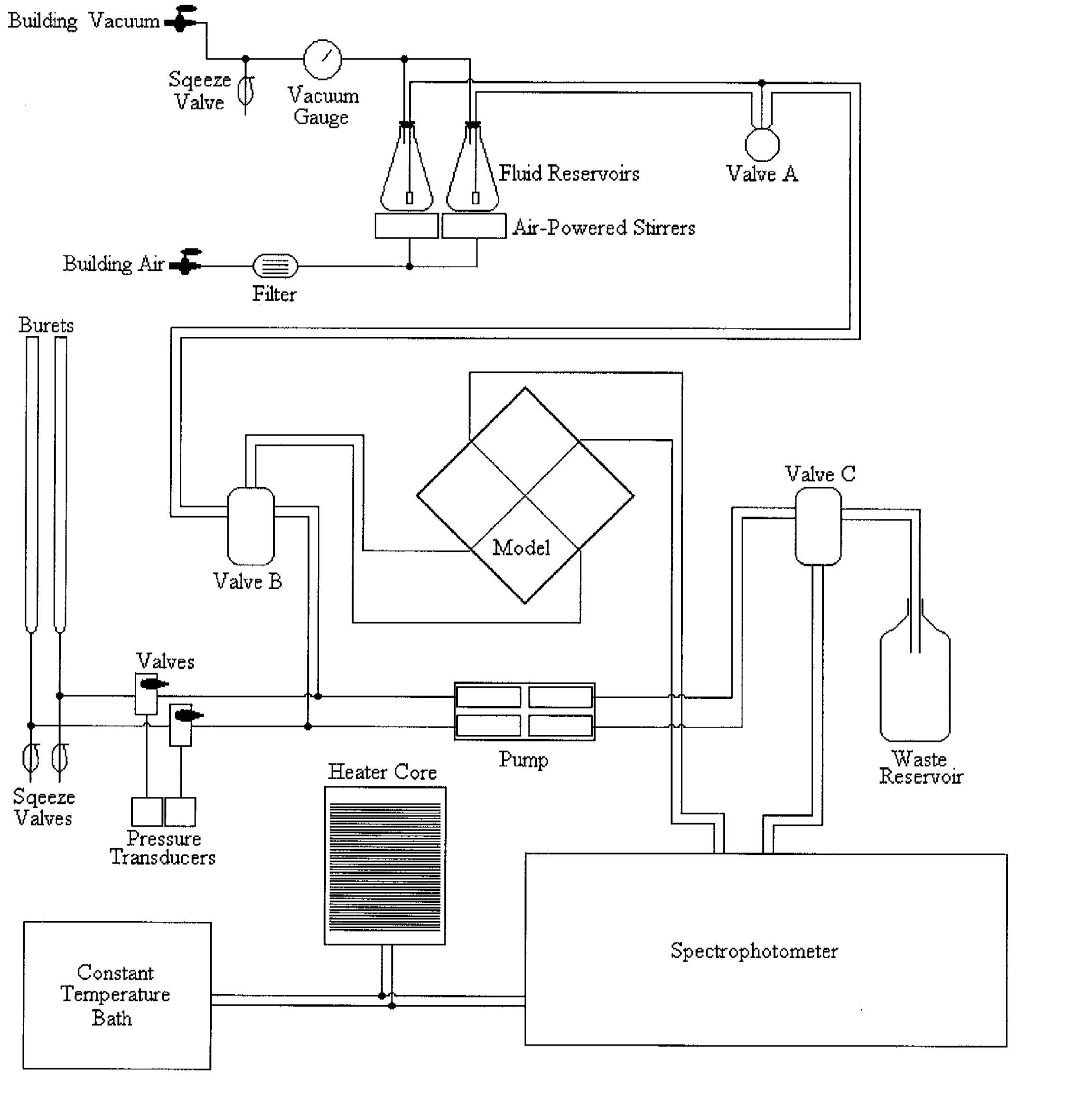


Figure A.1: The plumbing system. The individual parts are described in the equipment list in this appendix.

Two computers run the equipment and record measurements. Figure A.2 shows the connections between these computers and various pieces of equipment. At the end of this appendix is a list that describes these connections in more detail. The "terminal box" shown in Figure A.2 is a piece that I built for this project, so a diagram of the box appears in Figure A.3. It essentially consists of two long terminal strips connected by wires and resistors. Its purpose is to provide power to the thermistors and transducers and to provide an interface with the scanner.

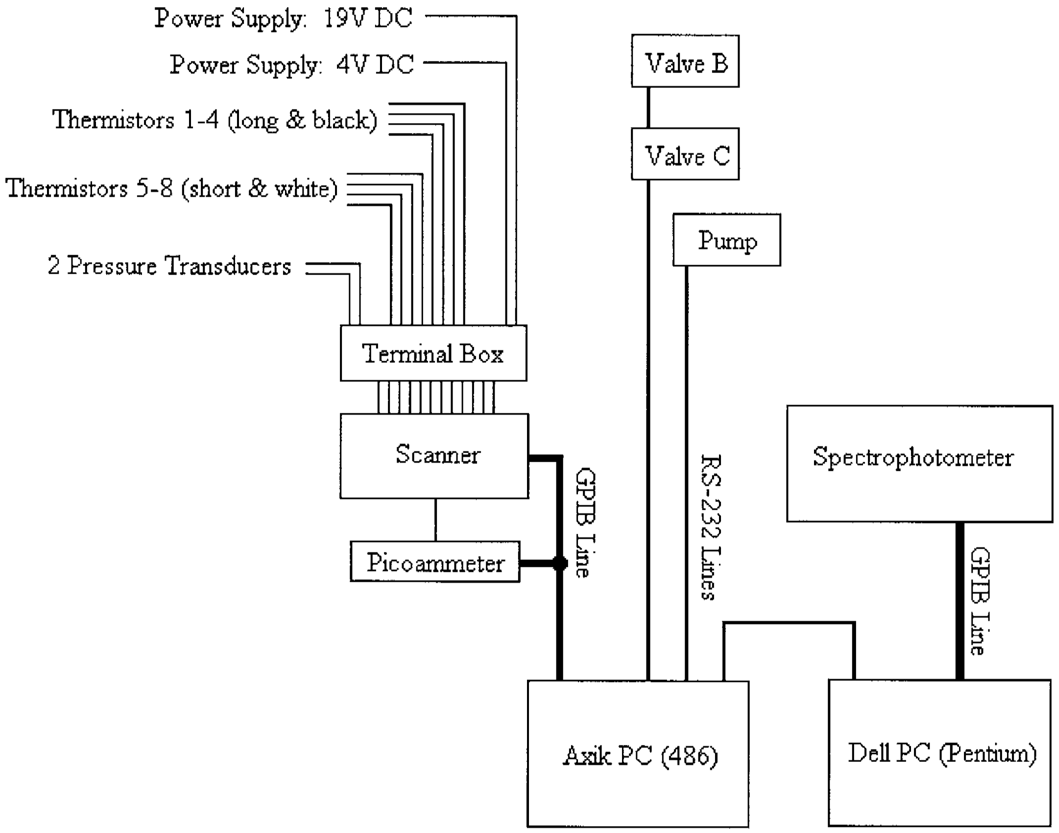
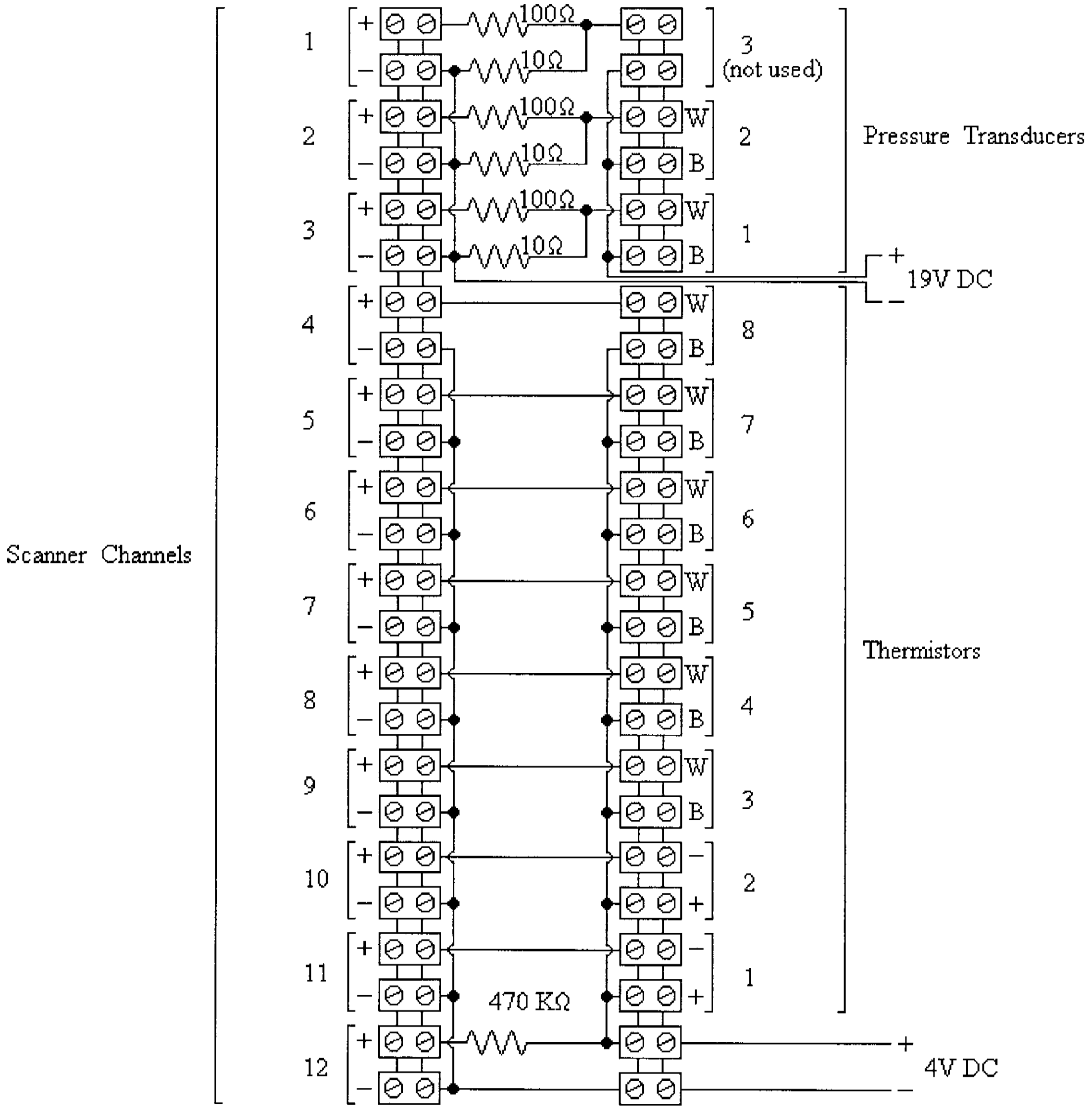


Figure A.2: Wiring diagram, showing the connections between electronic components of the system. See the connection list (below) for more detail.



For the scanner channels, "+" is the inner line and "-" is the shield line.
For transducers and thermistors, "W" is white and "B" is blue.

Figure A.3: Diagram of the terminal box that connects the scanner to the thermistors and transducers. The two long columns are screw terminal strips for connecting wires.

Equipment List

This list describes the equipment that I used for the experiments. Names that appear on the diagrams above are shown here in bold type.

Electronic Equipment:

Axik 486 PC with LabVIEW software, National Instruments GPIB communication card, and National Instruments four-port serial communication card.

Dell Pentium PC with HPChem software and GPIB communication card.

Spectrophotometer: Hewlett-Packard 8453 UV-Visible spectrophotometer with multicell transport.

Scanner: Keithley 706 scanner (for scanning analog data channels, not documents).

Picoammeter: Keithley 486 picoammeter.

Two DC power supplies.

Eight **thermistors**.

Two 4-20 mA **pressure transducers,** Setra model C206.

Terminal box (homemade) with terminal strips for connecting thermistors and pressure transducers to the picoammeter and power supplies. This box also converts the 4-20 mA transducer currents into 0.4-2 mA currents to avoid damaging the picoammeter.

PC-PC adaptor (homemade), allows two computers to be connected via serial ports.

Valve B: Hamilton 8-port Serial MVP valve, modified by replacing the original rotor with one that has only two fluid channels, on opposite sides, rather than four channels.

Valve C: Hamilton 8-port Serial MVP valve, unmodified.

Pump: KD Scientific syringe pump, modified for holding four syringes (two pushing and two pulling).

Constant Temperature Bath: VWR Scientific constant temperature water bath. It no longer works as a constant temperature bath, but it pumps water through the multicell transport on the spectrophometer in order to stir the cuvettes. It also pumps water through an automotive heater core (similar to a radiator) inside the insulated box.

Cables: GPIB, serial, and other cables for connecting the equipment. This includes a homemade cable for connecting the 486 to the KD Scientific pump, which has a telephone-type jack.

Panasonic Digital 5000 video camera, adapted to fit the microscope.

Video monitor.

Schott KL 1500 fiber optic light source, with a ring-shaped attachment designed to fit the microscope.

Power conditioners.

Air and Water Plumbing:

Air-Powered Stirrers: two magnetic stirrers, made by Kartell.

Homemade, T-shaped adaptor for connecting stirrers to building air system. This is made of copper tubing.

Air **filter**.

Vacuum gauge for monitoring reservoir pressure.

Silicone tubing and connectors.

Two modified stoppers for Erlenmeyer flasks, with holes drilled for a 1/16" PEEK tube and a 5mm glass tube.

Valve A: Hamilton 8-port manually operated valve.

Squeeze Valves: springy, plastic valves that go on the outside of silicone tubing.

Other **Valves**: Four two-position, three-port valves. Two of these are shown on the diagram. The other two tap into the lines between Valve B and the model, to allow sampling of the fluid on its way to the model. I never tried using these valves, partly because they would cause the pump to pull a vacuum on the downstream side of the model.

Four Hamilton Gastight 10ml glass syringes.

Four Hamilton Gastight 2.5ml glass syringes.

Five Hellma cuvettes with magnetic stirrer bars.

Two Hellma flow-through cuvets, type 178.711-QS.

Four modified PEEK fittings for connecting the flow-through cells to 1/16" OD PEEK tubing. These fittings have a small diameter to fit in the limited space on the flow cells.

Tubing, unions, tees, and fittings to connect the equipment. The tubing is .020" ID x 1/16" OD PEEK.

Heater Core: an automotive heater core, used for transferring heat from the insulated box to the constant temperature bath.

Burets, tubing, and connectors used for holding water to calibrate pressure transducers.

Fluid Reservoirs: two Erlenmeyer flasks, with magnetic stir bars and filter stones.

Waste Reservoir: a 2 liter jug for containing effluent water.

Other Equipment:

Fracture **model**, custom built by Coronado Machine, Albuquerque, New Mexico.

Zeiss microscope, model 47-50-52.

Insulated box made of wood, metal, and foamboard.

Temperature-controlled cabinet, built into the lab.

Connections

This is a partial list of how various pieces of equipment attach to each other. It's meant for any future New Mexico Tech students who happen to use the same equipment that I used. Refer also to figures A.1 and A.2.

Axik 486 PC:

On the four-port serial interface card, Port 1 connects to the pump, Port 2 connects to Valve C, and Port 3 connects to the Dell Pentium PC through the homemade PC-PC Adaptor. Port 4 is not used.

The GPIB port connects to the scanner and the picoammeter.

The large serial port connects to a Mettler AT400 balance. This is optional, of course – somebody else was using the balance, and I was logging data for him.

The small serial port connects to the mouse.

Dell Pentium PC:

The serial port connects to the 486 computer by way of the PC-PC Adaptor.

The GPIB port connects to the spectrophotometer.

Scanner:

The GPIB port connects to the 486 and the picoammeter.

On Card 1, all ten inputs connect to the terminal box. Note that you have to pull the card out (carefully!) to get to inputs 5-10. The top output connects to the picoammeter; the bottom output connects to Card 2. The order of the outputs probably doesn't matter, though.

On Card 2, inputs 1 and 2 connect to the terminal box, while one of the outputs (I use the top one) connects to Card 1.

Picoammeter:

The GPIB port connects to the scanner and the 486.

The BNC (coaxial) input connects to the output from Card 1 in the scanner. The input jack has three bumps instead of the usual two, so this requires an adaptor. The adaptor is shiny and cylindrical, and it has a yellow sticker with pink writing on it.

Terminal Box:

Take a look at the diagram for this. It connects to two power supplies, the scanner, the thermistors, and the transducers.

There are two types of thermistors, and it's important to get them straight to avoid confusing LabVIEW. Thermistors 1-4 are long, black thermistors. Thermistors 5-8 are short, white things that look like match heads.

The cables coming from the thermistors, transducers, and power supplies all have ground lines. Connect all the ground lines to the chassis – use the same screws that hold the terminal strips down.

Valve B:

The input line connects to the output from Valve C.

Valve C:

The output goes to the input on Valve B; the input comes from the 486 computer.

Spectrophotometer:

The GPIB line connects to the Dell Pentium.

The two fluid lines at the base of the multicell transport (next to the cable) connect to the constant temperature bath. I don't think it matters which way you hook them up. Don't use the connections at the top of the multicell transport – they're just for temperature control, which doesn't matter here. Stirring is the important thing.

The cable on the multicell transport connects to the spectrophotometer itself.

The multicell transport holds two flow-through cells and five cuvets with tiny stirrer bars. Each of the flow-through cells connects to one line in from the model and one line out to Valve C. It probably doesn't matter which direction the water flows; just be sure you know which cell corresponds to which fracture branch in the model.

Appendix B: Experiment-Specific Software Manual

Introduction:

The fracture model system used for low Peclet number mixing experiments is run by two computers: an Axik 486 running LabVIEW and a Dell Pentium Pro running HPChem. HPChem is used to control a UV/visible spectrophotometer, to calculate tracer concentrations, and to record data. LabVIEW is used for all other functions, including controlling devices and measuring temperature and pressure.

Both LabVIEW and HPChem can be used as programming environments to create high-level control software. In this way, a great deal of high-level software has been written specifically for the fracture model experiments. This manual is intended to describe the use of this application-specific material. For more general information on using LabVIEW and HPChem, consult the documentation that came with those packages. HPChem includes a maze of online help and minimal written documentation, neither of which is very useful. LabVIEW comes with extensive (and useful) written material and a dysfunctional online help system.

I'll try to make this manual as friendly as possible. Both HPChem and LabVIEW can be quite difficult to deal with at times, and I'll try to prevent some frustration. I've also tried to find all the bugs in my own scripts, since software that doesn't function correctly is even more frustrating than undocumented software. Of course, I try to *fix* the bugs as I find them, but I will point out a few that are still there. Perhaps the reader, after becoming familiar with the software, can fix these problems in the future.

Part 1: LabVIEW on the 486

Overview of LabVIEW:

LabVIEW is essentially an environment for creating process control systems. It's based around a graphical programming language called "G." By "graphical programming language," I mean that programs are written by manipulating pictures instead of text. This language can be used to assemble images of buttons, knobs, and dials into a control panel. The panel is called a "virtual instrument," or VI. If a computer has the necessary interface cards to communicate with external devices, it can be used to run anything from a voltmeter to a factory. In addition, external devices can be replaced by internal functions – in other words, LabVIEW can be used to *simulate* processes, in addition to controlling them. For our purposes, however, we'll be using it only for instrument control and data acquisition.

It's worth noting, for future reference, that LabVIEW has a variety of functions that haven't been put to use here. If you intend to write your own LabVIEW code, you might consider looking at the G Math Toolkit Reference Manual to get an idea of LabVIEW's more advanced functions.

This manual will not describe how to write VIs in LabVIEW. If you wish to write your own software, start with the LabVIEW Tutorial.

VIs you should know about:

The primary VI used in fracture junction experiments is called Mainfrac. It resides on the 486 at c:\labview\jason\mainfrac.vi. I keep this VI running and logging temperature data twenty-four hours per day, even when I'm not running an experiment. This manual is essentially about how to run this particular VI.

While Mainfrac is running, it calls a number of sub- VIs. Some of these, along with a couple that Mainfrac doesn't call, are useful for other purposes, like making sure that the hardware is functioning. All of the VIs on this list are found in the directory c:\labview\jason\instrmnt\.

Note that the Vis called by Mainfrac (Pumpchat, Sensors, and Valvchat) cannot be operated while Mainfrac is running – they're busy. If you open them, you'll see them taking commands from Mainfrac.

CALIBRAT.VI: This is used for calibrating the thermistors. It records resistance values as a function of known temperatures. See **Thermistor Calibration**.

PUMPCHAT.VI: This is used to communicate with the pump. It's called by a lot of otherVIs under Mainfrac, plus it can be used by itself.

To use it, find the list of commands in the pump manual. Type one of these commands (such as "ratei?") into the "Input" field and hit the run (arrow) button. You should see an answer come back.

This VI is very useful for troubleshooting problems with the pump or with Mainfrac.

SENSORS.VI: ThisVI is loaded automatically when you run Mainfrac. It queries the thermistors and pressure transducers and relays the information to Mainfrac.

TOMTEMP.VI: This VI was written in conjunction with Mainfrac, but for a different project (Tom Silverman's enhanced vapor diffusion experiments). The thermistors called "Tom's" on the Mainfrac stripchart are used for this purpose.

Tomtemp records temperature values and data from the Mettler AT400 balance in a directory called c:\tom\data.

Like Mainfrac, Tomtemp calls Sensors.vi automatically.

VALVCHAT.VI: This is one of the VIs you see when you hit the "Check Status/Errors" button in Mainfrac. It can also be used alone, like Pumpchat, to diagnose problems.

To use it, decide which valve (2 or 3) you want to talk to and type a command (such as "Ap") in the "Command String" field. The commands are in Section 4 of the Hamilton Serial MVP User's Manual (use the "DIN Protocol" list). Run the VI (hit the arrow button) and you should get an answer from the valve. To interpret the answer, see the valve manual.

For more information about Valvchat, see the **Controls and Indicators** list in this manual.

Starting and running LabVIEW and Mainfrac:

I'll assume that the reader has a basic knowledge of Microsoft Windows. Windows 3.1 might be archaic by the time you read this, but its operation is fairly similar to later versions (or at least Windows 95).

From Windows, find the LabVIEW menu and click the LabVIEW icon. A grey window will appear. This is a VI with nothing in it – a blank canvas, in other words. Now, pull down the file menu and open the file c:\labview\jason\mainfrac.vi. This is the main fracture model control VI. It's the only VI that you really need to know how to use in order to run experiments. This VI calls a number of sub-VIs, which in turn call other VIs, and so on.

Pull down the "Project" menu and click on "Show VI Hierarchy." You should see a gray window with lots of square icons and a tangled mass of red lines. The icon at the left edge of the diagram is Mainfrac, the VI you're currently using. The other icons represent all the sub-VIs that Mainfrac calls. As a general rule, colorful icons represent VIs that were written for this project, while black-and-white icons represent VIs that are built into LabVIEW.

Pull down the "Help" menu and click on "Show Help." Now, if you put the cursor over one of the icons, you'll see a brief description of what it does. If you double click on the icons, you can call up the VIs. Most of them aren't much to look at, though.

Now, close the hierarchy window (double click on the minus sign in the upper left corner) to get back to the Mainfrac window. Pull down the "Windows" menu (on the Mainfrac window) and choose "Show Diagram." You're now looking at the inside of Mainfrac, instead of its front panel. Note that the help window works here, too.

The first thing you'll notice about this diagram is that it's big, messy, and complicated. Don't worry, though – to run Mainfrac, you don't need to bother with this side of it. If you need to fix or modify something, you'll need to learn the basics of LabVIEW programming, a topic not covered in this manual (though I recommend that you learn it from the LabVIEW Tutorial). To run experiments, it's just important to understand what Mainfrac does and how to operate it.

To get back to the control panel, pull down the windows menu again and select "Show Panel." Close the help window, too – it doesn't work on the front panel.

Now, get ready to start the VI. If the pump, valves, and other hardware are hooked up for the fracture junction system, turn all of them on first. By the way, when turning the meter, scanner, and power supplies on and off, I always turn the power supplies on first and turn them off last. This is to make sure they're always stable while the more sensitive equipment is running. In other words, it's to avoid sending a spike to the picoammeter.

Next, click the little arrow button at the upper left corner of the Mainfrac window. This starts the program running. If the meter and scanner are on, you'll hear them start to make little chirping sounds. If you want to shut Mainfrac down for any reason, use the yellow "Software Shutdown" button at the upper left corner of the window.

A small panel called SENSORS.VI will appear. This is the sub-VI that monitors temperatures and pressures in the system. You can ignore it – just click on Mainfrac and Sensors will disappear.

You might see some weird error messages at first, but don't worry about it. Hit the "Initialize System" button and the error messages should go away. This button gets the various external devices into the states they need to be in.

In order to make sure everything is working and to familiarize yourself with Mainfrac, go through the following sequence to check out the systems.

First, take a look at the light blue "System Mode" box. It should say either "Run" or "Refill." If it says "Haywire," then refer to the section on error messages.

Now look at the "Pump Status" box. Hopefully, it says "Stopped." If it says "Running" or "Refilling," that's okay too. If it says "Weirdness," then refer to the section on error messages.

The "Current Pe" and "Injection Rate" boxes should both show numbers. If not, then hit the "Adjust" button next to the "Injection Rate" display and set a Peclet number – 100 will work.

The "Refill Rate" box should also show a number, probably ".01 ml/m." Hit the "Adjust" button and enter .1. The default value of .01 is awfully slow. Be sure you don't type the units, just the number.

The pump volume *must be checked manually***.** Take a look at one of the syringes on the injecting side of the pump and see how much water is in it. Enter this volume by using the "Set Volume" button under the syringe volume gauge. This may turn out to be frustrating – see **Pump Volume Confusion**.

The reservoir volume also needs to be set manually, but it isn't important.

Turn on the switch labelled "Temp & Press" and hit the "Check Strip Charts" button. If you wait a few minutes, you should see strip charts of temperature and pressure values. Hit the "Return to Mainfrac" button to come back.

Turn on the "Log Data" switch and set the log interval. 10 is a good value here. This will make Mainfrac record temperatures and other data every ten minutes.

Check the time and date at the top of the panel. You may need to change them in Windows – see **Date and Time** in the **Controls and Indicators** section of this manual.

To run experiments, see **The Auto Pe Section**. For a more thorough explanation of what you're looking at, see **Controls and Indicators**.

The Auto Pe Section:

This is the area with the blue-green background. This part of the panel is used during experiments, but you can ignore it if you're just setting things up. Its purpose is to automate the process of running a series of experiments at varying Peclet numbers.

Remember that gray boxes are inputs and blue boxes are outputs.

The "Starting Pe" and "Increment" boxes are where you determine what Peclet number you'll start with and how much the Peclet number will change.

"Volume / Sample" is the interval (in milliliters) at which measurements are taken. This volume is based on a single syringe, not both of them. Note that since the interval is a constant *volume,* the *time* interval between samples will vary with the pumping rate. I tried using a constant time interval at first, but that lead to a lot of data for low Peclet numbers and not much data for high Peclet numbers. Using a volume interval makes more sense when you start comparing the data from different runs.

"Samples / Pe" is simply the number of samples it will take before changing the pumping rate.

Example: Let's say you want to get data at Peclet numbers 100, 90, 80, and so on. You would set "Starting Pe" to 100 and "Increment" to –10. These inputs are real numbers, by the way; they don't have to be integers.

If you set "Volume / Sample" to .25 and "Samples / Pe" to 8, then the pump will send two milliliters of solution through the system (from each syringe) at each Peclet number.

If you turn on the "Auto Pe" and "Refill on Start" switches and then hit the "Initiate Pe Program" button, then several things will happen. Mainfrac will first refill the pump and reset the injection rate to correspond with the "Starting Pe." When the pump is full, it will start injecting and periodically telling the UV computer to take samples. The blue boxes will tell you at what volume the next sample will be taken and how many samples have been taken so far at the current Peclet number. The UV computer should also display, in the lower left corner of the monitor, its own estimate of when the next sample will be taken. This won't happen until after the first sample, though. When the spectrophotometer is taking samples, the "UV Busy" LED will turn black. Meanwhile, messages like "Thanks for the data" and "Done" will appear in the "Last UV Comment" box. These are messages sent from the UV computer. If it says "UV Timeout!" then you have a communication problem – see **The Babysitting Factor**.

If everything is working correctly, it will run until the Peclet number is lower than or equal to the increment. It will then just keep running at that Peclet number indefinitely. It's good not to make your last Pe a high one, because it might drain the reservoirs while you're home in bed. Note that the Peclet numbers won't be exactly what you expect – for example, the first Pe in the example might be 99.98 instead of 100, and the last one might turn out to be 9 or 1 instead of 10. This is because the computer and the pump play a little game of telephone with the pumping rate and it gets rounded off in translation. The computer tells the pump what rate it *wants*, then the pump tells the computer the *actual* rate. The next Pe will be the actual Pe (not the expected Pe) minus the increment.

The Babysitting Factor:

This system has several problems that are beyond my control, most of which I've been able to work around. One that eludes all attempts at a cure is this: HPChem will occasionally, for no discernible reason, stop taking input from the RS232 port. That means that when Mainfrac tells it to take a sample, it doesn't listen. I've programmed the HPChem macro to estimate the time of the next sample so that if the expected message isn't received, it will take matters into its own hands and start sampling anyway. When this happens, the HPChem display will read, "No Communication!!!!!!" Sometimes, at low Peclet numbers, it jumps the gun and thinks there's no communication when it really just hasn't waited long enough. Either way, the result is a set of concentration data that don't have matching volume, pumping rate, and temperature data (since these can only come from Mainfrac). To salvage the data, you have to look at the separate files on the two computers and correlate the times. You *have* been making sure the clocks are in synch, right? See **Log Data** in the **Controls and Indicators** section for information on Mainfrac's data files.

The way to avoid all this is to babysit the system. When you're running an experiment, you need to come in periodically – say, on your way home from the Cap – and make sure the computers are still on speaking terms. Push the "Say Hi" button on Mainfrac and watch the UV computer. A howdy message should appear briefly in place of the estimated time of the next sample. If it does, you're okay. If it doesn't, the only solution I know of is to shut down HPChem and restart it.

If you have to shut down HPChem, FOLLOW THIS PROCEDURE to avoid losing data:

1) Hit the "X" button to close the window.
2) When the dialog box comes up to close HPChem, hit "Cancel." In other words, you're pretending to close it down but you're really *not* closing it down yet. This is the only way I know of to stop the macro when the computers aren't communicating.
3) Go to the "View" menu and select "Samples." If it says "The specified register does not exist!" then you're okay. It means that the data have already been saved (unless it has been down so long that it never took any samples and your whole run was wasted… remind me not to have you take care of my plants).
4) If there are samples, go to the command line and type "janrefill". This has exactly the same effect as the "Process" button in Mainfrac.
5) Your data are now saved, so you can exit HPChem for real now – hit the "X" button and exit without saving any settings.
6) Run HPChem and the jstart macro.
7) Hit the "Sample Now" button. The spectrophotometer should start squeaking and whirring. If it does, then you're safe until the next time it decides to stop communicating.

Pump Volume Confusion

This is a bug that I haven't been able to fix. When you start Mainfrac and set the pump volume, sometimes it just doesn't believe you. Also, the pump volume value might change when you change directions. However, it works fine during experiments.

The root of this problem is the fact that the pump has no way of sensing its own position. The pump only knows how much fluid has been delivered since the last time it was reset. Mainfrac has to calculate the pump volume from this number, and that requires knowing what the *previous* volume of the pump was... You can see why this requires some user intervention to check the volume. If you turn Mainfrac off and back on, then it assumes that the pump was last reset at either 10 ml (if it's in run mode) or 1 ml (if it's in refill mode). If this assumption is correct, then there's no problem. If not, then things get confusing. By the way, the pump is reset any time it changes directions.

When you turn on the pump and Mainfrac, you'll have to be patient when you set the volume. Change directions (hit the "Run Mode" and "Refill Mode" buttons) several times to make sure the volume value is stable. You might also try running it briefly and stopping it. This is because the pump retains the old "delivered" value, rather than going to zero, until it starts moving.

Once you have the volume set believably, it should be okay until you turn it off again.

Controls and Indicators

Mainfrac and its associated VIs receive user input through *controls* (buttons, switches, and gray boxes) while providing data through *indicators* (blue boxes, virtual

LEDs, et cetera). Knowing how to run Mainfrac is essentially a matter of knowing what the controls do and what the indicators mean.

Some of the information displayed by Mainfrac comes from external instruments, while some of it is generated within the computer.

What follows is a list of all the controls and indicators in Mainfrac, Valvchat, Modechek, and Pumpchat.

#Samples @ Pe, A.K.A. **Number of Samples at This Peclet Number:** This indicates how many times the spectrophotometer has analyzed samples since the Pe was automatically changed, or since the pump refilled, or since the Peclet program was initiated, whichever happened most recently.

Auto Pe: When this switch is on, Mainfrac automatically adjusts the flow rate (and hence the Peclet number) and directs the other computer to take measurements on the spectrophotometer. Experimental parameters are determined by the controls in the green box.

The "Sample Now," "Stop UV," and "Say Hi" buttons will still work when "Auto Pe" is off.

Auto Refill: When this switch is on, the pump automatically changes direction when it's full or empty. The "Auto Pe" button has the same effect.

Back to Mainfrac: Brings up the Mainfrac front panel.

Byte Translation: Takes the character specified by "Character Number" and shows what each bit is doing. See the Hamilton Serial MVP User's Manual, pages 4-5 and 4-6, to find out why you might care. Basically, these bits indicate various possible error conditions in the valve. This is only for diagnosing problems, and hopefully you won't have any problems that the "Initialize System" button can't fix.

Character Number: Using this control, you can tell the VI which part of the "Raw Response" to use for the "Byte Translation" (below). The first character is number zero, *not one.* Note that words in brackets, like "<ACK>," constitute one character.

Check Status/Errors: This button brings up extra panels to monitor the pump and valves. If there is a problem, this is a good place to diagnose it. Alternatively, you can always just hit the Initialize System button and probably solve the problem. Readouts on the panels are included in this list.

Check Strip Charts: This one is important. It shows you real-time plots of the pressures and temperatures in the experiment.

Command String: The command being sent *to* the valve *from* Mainfrac. This is usually "Ap," which means "What position are you in?"

Current Pe: Mainfrac takes the current injection rate (displayed in another box) and converts this into a Peclet number. Of course, it has to know the diffusion coefficient for this to work. The diffusion coefficient and other parameters can be changed in RATEPEC.VI, which is one of the VIs in the "convert" directory.

The Peclet number can be changed using the "Adjust" button next to the "Injection Rate" display (see below).

Date and Time: This is at top center on the Mainfrac display. Note that this computer's clock isn't always correct; I don't know why. You should check it about once a week. The clock can be reset from Windows. From the Program Manager, go to Main...Control Panel...Date/Time. I like to keep it on the same time as the Pentium (the

UV computer). The only real significance of the date/time value is that it's recorded in the data file with the temperatures and pressures (see **Log Data** in this section and **The Babysitting Factor** under **Auto Pe Section**).

Error: This LED is to the right of the "Stop Pump" button, and it's normally kind of invisible. If it lights up in yellow, then there's a problem with the pump and/or valves. When this happens, you should hit either the "Initialize System" button (to try to fix the problem) or the "Check Status/Errors" button (to diagnose it).

Error Boolean: Turns red if something bad happens. If only one valve is screwed up and the other is okay, this indicator will flash green and red.

Error Message: This is the VI's opinion of how things are going with a valve. It usually says "Valve #3 is hunkey-dorey." See the **Error Messages** section.

Increment: Determines how much the Peclet number will change after each set of measurements.

Initialize System: This is a good button to use if anything strange is happening. It is *necessary* when the external devices, especially the picoammeter, are first turned on. It is *not* necessary every time you restart Mainfrac. It never hurts, though, unless you're in the middle of an experiment.

The initialization operation consists of six steps. If you feel like opening INITALL.VI (c:\labview\jason\instrmnt\initall.vi), you can see the steps firsthand by scrolling through the sequence frame. It first initializes the scanner, making sure all channels are open. It then calibrates the picoammeter (an operation in INITSCAN.VI). Next, it initializes the serial ports that Mainfrac uses to communicate with the valves and pump. Finally, it stops the pump, puts the pump in "inject" (forward) mode, resets the syringe diameter (the diameter can be changed in INITALL.VI), and initializes the valves. The valve initialization process includes setting both of them in run mode, which connects the pump to the model.

In addition to this initialization, there is a set of operations that Mainfrac performs automatically on startup. This mainly consists of asking the pump questions, and it does not change any settings or valve positions.

Note that if you do not use the Initialize System button, you can shut Mainfrac off and turn it back on without interrupting an experiment. This is part of the reason that the full initialization does not happen automatically on startup – it's so I can debug Mainfrac and run experiments at the same time.

Initiate Pe Program: This button begins an experimental run. It resets the pumping rate in accordance with the value in "Starting Pe" and starts the pump running. If the "Refill on Start" switch is on, then the "Initate…" button will direct the pump to refill before starting the experiment.

Note that if "Auto Pe" is off, then nothing will happen.

Injection Rate: This is the current rate setting as reported by the pump. It can be changed using the "Adjust" button.

Input: This control, on PUMPCHAT.VI, determines what message the computer sends to the pump. See the pump manual for a list of available commands.

While Mainfrac is running, you'll see a continuous series of questions appearing on this control.

Last UV Comment: This display reports the last message received from the other computer. "Thanks for the data" means that the other computer has received data and is analyzing samples. At this point, the “HP Busy” LED should be on. When the other computer finishes analyzing samples, it reports “Done” and the LED goes off.

“UV Timeout” means that no response was received from the other computer. This could mean that the other computer is not running the monitoring script in HPChem (see the HPChem section of this manual). Another possibility is that the other computer’s serial port has stopped functioning – see **The Babysitting Factor.** Closing HPChem and restarting it seems to get things going again.

To see if communication is working, push the “Say Hi” button and see if a howdy message comes up on the Pentium.

Log Data: When this switch is on, Mainfrac records data to the hard drive. The directory is c:\jason\data, and the filename is the letter “l” followed by the date. The year is only one digit, because there’s only room for eight characters.

Each record in the data file has 13 fields: date and time, eight temperatures, two pressures, the syringe volume, and the flow rate. The temperature values are in the same order as on the strip chart screen (vent, room, etc.).

LOGDATA.VI is where the file name, directory, and format are determined.

Log Interval: Determines how often data are recorded on the hard drive.

Master Error: This comes on if any of the four LEDs below it is on. It’s the same as the “Error” LED on the main panel. You can ignore it.

Moving/Stopped: Black means the pump is moving; grey means it’s stopped.

Number of Samples at This Peclet Number: See “#Samples @ Pe.”

Output: This indicator is on PUMPCHAT.VI. It's the message from the pump (see "Response"), with the extraneous characters removed.

Process: Instructs the UV computer to process and record the current spectra, then delete them from the sample register. This is the same command that Mainfrac sends when the pump is refilling.

The UV computer does this automatically between samples, if there is enough time. At high Peclet numbers, when samples are taken rapidly, it waits for the command from Mainfrac.

The relevant macro in HPChem is janrefill.

Pump Error: This means that when asked what mode it was in, the pump said something other than "I" or "W."

Pump Mode: "W" and "I" stand for "withdraw" and "inject," which are the pump's words for refill and run mode.

Pump Status: This display tells what the pump is currently doing. It normally has three states: Running, Refilling, and Stopped. Like the system mode indicator, this information comes from a continuous interrogation of the pump.

There is a fourth state, Weirdness, which will not come up under normal circumstances. It indicates that the computer has received an unexpected message, or no message at all, from the pump. It probably means that the pump is switched off. If weirdness persists after the pump is turned on, try initializing the system. If this doesn't work, you're on your own. Start PUMPCHAT.VI, get out the pump manual, and try asking the pump questions.

Pump Stop: This virtual LED indicates that the pump has stopped itself without the user's direct involvement. This means either that it reached its specified limit (1 ml or 10 ml) or that it overshot and reached its physical limit (about .5 ml and 10.5 ml). The latter occurs when the syringe volume is set incorrectly.

If Mainfrac is in Auto Pe or Auto Refill mode, the Pump Stop LED should flash briefly whenever the pump changes direction.

Raw Response: This is the uncensored version of what the valve said. It includes several bytes that don't translate into ASCII characters, so they just look like black rectangles.

Refill Rate: This is the current rate setting as reported by the pump. It can be changed using the "Adjust" button. When entering the new flow rate, just type the number and not the units. It will assume you mean milliliters per minute.

I recommend a refill rate of .1 ml/m. If it's too fast, it will suck air in through the valves. Due to its low viscosity, it's easier for air to leak in than for water to leak out.

Reservoir Volume: This indicates the amount of fluid remaining in each of the Erlenmeyer flasks that supply the pump. Unlike the syringe volume, this value must be reset each time Mainfrac is turned on – the information is not stored when the VI is turned off. You'll need to visually inspect the reservoirs (hopefully they both have the same volume of fluid) and use the "Set Res. Volume" button.

Whenever Mainfrac increases the value for the syringe volume (when the syringe is refilling), it subtracts the same amount from the reservoir volume. Thus, the reservoir volume is the difference between the last set volume and the cumulative volume withdrawn by the pump.

Syringe Volume: This gauge indicates the volume of fluid in each of the two syringes on the upstream side of the pump – in other words, it's the amount available for experiments. This value is inferred from other information that the pump provides, and it isn't necessarily correct. **The syringe volume must be checked manually**, especially if the pump has been turned off. Pull out one of the syringes and read its volume, then calibrate the volume setting using the "Set Volume" button. **Incorrect volume settings can lead to pump problems** because the pump will try to push the syringes farther than is physically possible.

Reset UV: Clears the "UV Busy" state. This button is useful if there has been a communication problem.

Response: This indicator, located on PUMPCHAT.VI, shows what the pump is telling the computer. While Mainfrac is running, you'll see the answers to questions (i.e. the flow rate) appearing here.

Note that messages from the pump always end with the letter "E." According to the pump manual, this indicates an error condition. However, I've never seen the pump *not* return an "E."

Run/Refill: This is a boolean indicator (a virtual LED). If it's black, the pump is in run mode. If it's grey, the pump is in refill mode.

Sample Now: Directs the other computer to measure samples on the spectrophotometer.

Samples / Pe: Determines how many measurements will be taken at each Peclet number.

Say Hi: Sends a friendly message to the UV computer. If the fracture junction macro is running under HPChem, this message will briefly appear at the lower left (where it says "This run will end...").

This button is very useful to determine if the computers are on speaking terms. Since the UV computer periodically stops receiving messages for no apparent reason, I push this button often just to make sure things are working.

Software Shutdown: When this button is pressed, Mainfrac waits until all current operations are completed, then shuts down. Using this button is preferable to using the "Abort Execution" button (the little one up top with a stopsign on it), which is likely to leave unfinished business.

Note that if the pump is running when you hit the "Software Shutdown" button, the pump will continue running even though Mainfrac stops.

Starting Pe: When you hit the "Initiate Pe Program" button, Mainfrac will reset the Peclet number to this value.

Stop UV: Tells the other computer to stop running the fracture junction macro. This is useful because HPChem has no "stop" button for macros.

Note that, like Mainfrac, the HPChem macro can be stopped and restarted without interrupting an experiment, *as long as it's between measurements.*

System Mode: This display sums up the positions of the pump and valves. It has three states: Run, Refill, and Haywire. "Run" means that the pump is set to run forward and it is hydraulically connected to the model. "Refill" means that the pump is set to run backward and it's hydraulically connected to the fluid reservoirs. If it says "Haywire," then referr to the Troubleshooting section of this manual.

The information in this display comes from a continuous interrogation of the pump and valves.

Temp & Press: Determines whether or not Mainfrac will measure temperature and pressure. If you don't have the external devices (thermistors, etc.) hooked up yet, leave this switch off.

Thermometer: This is the temperature at the fracture junction. It's the average value from two thermistors. You can determine which thermistors are polled by changing wires in STRIPCHT.VI.

Time and Date: See "Date and Time."

Time Since Write: This reports the time (in minutes) since temperature and pressure data were recorded on the hard drive

Translated Response: This is essentially a sentence from the valve. It's translated from the "Raw Response" (below).

UV Busy: When this LED turns black, it means that Mainfrac told HPChem to make a measurement and HPChem has not yet told Mainfrac that it's done. If this LED is on (black) and the spectrophotometer isn't whirring and squeaking, then something is amiss. Pushing the "Reset UV" button should put things in order.

V @ Next Sample, A.K.A. **Volume at Next Sample:** When the syringe volume reaches the specified volume, Mainfrac will instruct the other computer to analyze samples on the spectrophotometer.

Valve 2 Com: Valve #2 is not communicating. Note that due to a design change, there is no Valve #1.

Valve 3 Com: See Valve 2 Com.

Valve Mismatch: The two valves are not in the same position.

Valve Number: Shows which valve (2 or 3) it's talking to.

Valve Positions: Position 1 is run, position 2 is refill. You shouldn't see any other numbers here. Because of the way the valves work, though, all even-numbered positions are the same as 2, and all odd-numbered positions are the same as 1. Note that if the valves are not in the same position, this will produce a "Haywire" mode indication. The Initialize System button will fix this.

Volume / Sample: Determines how often samples are measured on the spectrophotometer. Each time the pump pushes this much fluid (in milliliters) through the model, a new measurement will occur.

Error Messages

Some of these are messages that appear in indicator boxes and others are the names of virtual LEDs that occasionally light up. This list covers error messages in Mainfrac, Pumpchek, and Valvchat.

Beeping Valve: If a valve gets *really* confused, it will make a loud, beeping sound. Use the "Initialize System" button to reset it.

Error Boolean: This is just a general "something's wrong" light. It won't tell you anything specific.

Haywire: For normal operation, the states of the pump and the two valves all have to be in agreement – i.e. all set to refill. If they aren't in agreement, or if there is a communication problem, then a "Haywire" message will result.

Try hitting the "Run Mode," "Refill Mode," and "Initialize System" buttons (one at a time, please). Any one of these should fix the problem.

If that doesn't work, then hit the "Check Status/Errors" button. This will bring up a new set of panels that show information about the pump and valves.

Look at the LEDs on the right side of the screen. Normally, they're all green. If you see red LEDs, then look them up below.

By the way, "Master Error" comes on whenever one of the others is on. Ignore it.

If the LEDs are all green but you're in Haywire mode, it probably means that the pump is doing one thing but the valves are doing another. If the pump mode is "I," then the valves should be in position 1. If the pump mode is "W," then the valves should be in position 2.

If you can't fix this problem by reinitializing or changing modes, then shut off Mainfrac, start Pumpchat, and get out the pump manual. You're on your own from here.

Master Error: This just means that one of the *other* error LEDs is on. Ignore the Master Error and see about the other one.

<NAK> or weird response from valve #x. See "Weird response…"

Pump Error: Mainfrac is always asking the pump what mode it's in, and the pump responds with either "I" (inject) or "W" (withdraw). If it responds with something else or doesn't respond, then you get a pump error.

If characters other than "I" and "W" appear, it's usually because the pump and computer are trying to talk at the same time. This is a temporary condition. If it doesn't clear up in a few seconds, then start PUMPCHAT.VI, get out the pump manual, and start asking the pump questions. You'll have to shut down Mainfrac in order to do this.

If no message is coming from the pump, then either the pump is off (see if the display on the pump is lit up) or the serial line isn't connected properly.

Something weird is happening. This message will appear if a valve is sending a strange number of characters to Mainfrac. As far as I know, there is *no way* you will ever see this message. If you're seeing it, then you're screwed.

UV Timeout! This means that Mainfrac tried to talk to the UV computer and the UV computer ignored it. See "The Babysitting Factor" under "Auto Pe Section."

Valve x Com: There's a communication problem. If reinitialization doesn't help, then see if the valve is getting power (check the green light on the valve) and see if the serial line is connected properly. That includes making sure it's in the right port.

Valve #x is hunkey-dorey. Congratulations! The valve is working.

Valve #x is not responding. I can think of three reasons you might get this message:

1) The valve is just confused. Reinitialize.
2) The valve is not getting power. See if its little, green light is on.
3) The serial line between the valve and the computer isn't properly hooked up. Make sure the connections are good and it's in the right port.

Valve Mismatch: The valves aren't in the same position, and they should be. See the valve positions at the middle of the screen, which should both be either 1 or 2.

If reinitializing doesn't fix this, then shut off Mainfrac, start VALVCHAT.VI, get out the valve manual, and start asking the valves questions.

Weirdness: The pump is saying strange things. This should only last a few seconds, and it's normal when the flow rate changes or the pump changes directions. If the problem persists, consult your physician or PUMPCHAT.VI.

Weird response from valve #x. The valve is on and the serial line is connected,but the valve is talking gibberish. Or, possibly, you've got the serial lines mixed up and you're talking to some other device. The most probable cause is a confused valve – reinitialize. If reinitializing doesn't work, then try unplugging the power from the valve, plugging it back in, and *then* reinitializing.

Troubleshooting

This is a list of what symptoms occur with each problem. If you suspect a particular problem, you can verify it here. If you have no idea what's wrong, just read through this list until it starts to sound familiar.

"Status Panel" refers to what you see after you hit the "Check Status/Errors" button. "LED" means "The LED is on," i.e. it's red, black, or yellow.

If you determine that something is offline, check to make sure that it's on and that its communication line is plugged into the right port. *All* of the equipment lights up when it's on, so power is easy to check. On the pump, it's the display that lights up. When you bring it back online, you'll need to reinitialize (hit the "Initialize System" button).

Picoammeter offline: The whole system runs *really* slowly. It takes several minutes to react when you push a button. You might think it's dead.

If you really need to run Mainfrac without the picoammeter, then shut down SENSORS.VI. To do this, go to the "Windows" pull-down menu, select SENSORS.VI, and hit the button that reads "STOP" (not the stopsign button).

If you do this, you'll find that when you go to check the strip charts, you can't get back. Use the "Windows" pull-down menu to get back to the Mainfrac panel.

Pump offline: Pump Status is Weirdness, Current Pe is NaN (this stands for "Not a Number"), rate indicators are blank. Status Panel: Pump Mode is blank, Pump Error LED is on.

Scanner offline: All temperature values on the strip chart are completely random, ranging from negative values to 2299.98.

Valve offline: System Mode is Haywire. Status Panel: Valve Mismatch, Valve x Com, and Error Boolean LEDs. Error Message is "Valve #x is not responding."

Thermistor Calibration:

Make sure the thermistor system (power supply, scanner, picoammeter, connection box, and thermistors) is running. The power supply should be set at 4 volts.

Put all the thermistors in the constant temperature water bath, along with a regular thermometer. Turn on the water bath and give it a temperature.

Enter a path and file name (i.e. "c:\therm.dat") in the "File Name" field.

Hit the "Initialize" button. This will get the picoammeter and scanner running the way they're supposed to.

Note the value in the "EMF to thermistors" display. This is a value (in volts) determined by measuring the current through a fixed resistor. It should show about four volts, same as the power supply. If it doesn't say four volts, there's a problem, probably with either the power supply or the fixed resistor. In that case, get out the voltmeter and go to work.

Now, the scanner should be chirping and the strip chart on the VI should be running. You should see two bunches of four values, because there are eight thermistors of two types.

Look at the temperature in the water bath, type it into the "Temperature Input" control, and hit the "Record" button several times (you'll have to wait for the button to pop back up). This button records data to the file you specified.

Adjust the water bath, wait for it to stabilize, change the temperature input, and hit "Record" a few more times. Keep doing this until you're satisfied.

When you're done, get a floppy and copy the file onto the UV computer (the Dell Pentium) so that you can use Excel. If you're not an Excel expert, find someone who is.

On the Pentium, find the file called cali5-21 and open it in Excel. Here you see the file (unedited) that I last used to calibrate the thermistors. If you ignore Set 1, which is screwed up, you'll see that the thermistors fall into two groups – the ones with big heads (1-4) and the ones with little heads (5-8).

Load *your* file into Excel, graph the data, and find those trendlines. Use "Power" trendlines for thermistors 1-4 and exponential trendlines for thermistors 5-8.

Note that Excel will tell you the resistance as a function of temperature. Using a pencil, rearrange the equations for temperature as a function of resistance. Write them in the forms $T = e^{[(R-b)/a]}$ and $T = [\ln(R/a)]/b$, where T is temperature, R is resistance, and a and b are constants.

On the 486, load GETTEMP.VI and bring up the diagram window. Scroll through the sequence frames (the box that looks like a piece of film) and note that each frame has a different channel number. These channel numbers refer to the scanner.

You need to know which channels correspond to which thermistors. This is determined by the way the wires are hooked up at the connection box. As I write this, channel 11 is thermistor 1, channel 10 is thermistor 2, and so on.

The constants (a and b) that you determined for the thermistors go in the blue and orange boxes. For the first four, b is on the left and a is on the right. For the other four, a is on the left and b is on the right. You should be able to see how the operations in LabVIEW (the icons) correspond with the the equations you wrote.

Go to the front panel of GETTEMP.VI and run it several times. The values shown on the strip chart should match the temperature of the water bath. If so, then you're done.

Part 2: HPChem on the Pentium

Overview of HPChem:

HPChem is designed to control Hewlett-Packard analytical equipment and to process data for chemical analyses. HPChem has its own built-in macro language, just as Excel comes with Visual Basic. By using macros, you can instruct HPChem to run autonomously and collect data during an experiment. Usually, it obeys these instructions.

This manual is intended to explain how to use macros, but not how to write them. If you do want to write them, I can suggest three sources of information. The best source is probably the set of macros that I've written. That's not to say that my macros are pedagogical wonders, just that you'll have more luck learning by example than sorting through Hewlett-Packard's feeble attempt at online help. The second thing to try is to type **show** at the command line and see what commands and macros are available. If you can make an educated guess at what command you need in a given situation, then you can look the command up and see if it exists. Finally, you can try the Macro Programming Guide, which is an Adobe Acrobat file. You'll see it when you go to turn on HPChem.

Starting and running HPChem:

I'm going to assume that you already have Mainfrac (the LabVIEW VI) running on the 486. If you do, then make sure the two computers are connected to each other. Be sure to use the special, homemade adaptor, which switches the receive and transmit lines. If you *don't* have the other computer running, then you'll have to use your imagination.

Now, go through this list:

Make sure the computer is connected to the spectrophotometer with a GPIB cable.

Turn on the spectrophotometer.

From the **Start** menu, choose **Programs \ HP UV-Visible Chemstations \ Instrument 1 online**. Note that the Macro Guide, which you might need someday (but not now), is next to **Instrument 1 online**.

Type your name when it asks for it. You don't need a password.

If you just turned on the spectrophotometer, you'll see a dialog box that says, "Loading this method will change current lamp settings!" It's okay to change the settings – hit the **Use New** button.

There will be another box that says, "There are already standards loaded!" Hit the **Discard** button.

Click on the little spectrum at the extreme upper left corner of the monitor – the one that looks like the "Dark Side of the Moon" album cover. Choose **Command Line**. This will bring up a line at the bottom of the screen where you can enter text and tell HPChem what to do.

Click on the command line to activate it. Type **listmessages on**. You should now see a white box on the screen. Output from HPChem will appear in this box.

Type **jstart**. When it asks for a time and date, enter some time in the future. This is the time and date at which the macro will automatically stop running. I generally just set it to run for a year – this requires minimal typing.

After you hit the **Accept** button, things will start clicking and whirring. The spectrophotometer will scan a blank and then wait for a message from the other computer.

Under normal circumstances, that's all you need to do with this computer to get things running. Of course, these aren't normal circumstances, because you probably just set it up. You'll need to put solutions in it and do some calibration, but we'll get to that later.

<u>What you just did:</u>

The command line is essentially an interpreter for the macro language. This language is, fortunately, a little like BASIC. One way to use the command line is to give commands directly, like when you typed **listmessages on**. Another way to use it is to write a macro (using a separate text editor), load the macro into memory, and type the name of the macro on the command line. When you typed **jstart**, you called a macro. This macro in turn called other macros, and so on.

On the Windows desktop, there's a file called **Macros**. Open this file. Click on the shortcut to **umacinit**. You'll see a text file containing a macro called **UM_AutoStart**. HPChem looks for this file and runs the macro on startup. The macro then loads several other files, including three files containing other macros. One of these files – **operate.mac** – contains the macro **jstart**. Because it had already loaded **operate.mac**, HPChem knew what you meant when you typed **jstart**.

Things to do in HPChem:

With both HPChem and LabVIEW running, hit the **Sample Now** button in Mainfrac. This will cause the spectrophotometer to click and whir. It will shoot a spectrum on each cuvet and save the data to a file. This is what normally occurs during experiments. If you don't have any solutions in the spectrophotometer, you might be looking at some bizarre spectra right now. This isn't a problem.

In Mainfrac, hit the **Stop UV** button. This will make the macro stop running in HPChem. If the **Stop UV** button doesn't work, which happens fairly often, then you can hit the **X** button in the upper-right corner and then hit the **Cancel** button. In other words, pretend that you're going to shut down HPChem, but then don't.

Now, take a look at what commands and macros are available. Type **show** at the command prompt. You'll see a dialog box with a list of commands in it. If you click on one of these commands, it will tell you what input is necessary to use the command. Click on **Macros** in the domain box, and you'll see a list of macros. Hit **j** on the keyboard, and you'll see all the macros that start with the letter "j".

The names of all the macros written for this project start with the letter "j." This stands for "junction" or "Jason," and it helps to distinguish them from preexisting macros. The ones that start with "jan" are analysis macros. Note that one macro is called **jstart** – that's the one you ran earlier.

You don't need to run any macros – I just wanted you to know where to find them. If you want to see (or modify) the actual text of the macros, look at the files in the **Macros** folder on the desktop. Be aware that if you do modify a macro, you have to load it into memory or HPChem will keep running the old version. Macros like **jloadoperate** (the first macro in the **operate** file) are designed to load the files.

By the way, most of the "j" macros are meant to be called by other macros in specific situations. That means that if you try to run one now, you'll probably get a weird error message.

Now, take a look around. From the **View** menu, choose **Standards**. This will show you the two standard spectra (concentration = 1 and concentration = 0) used to calibrate HPChem for analysis. You may see each spectrum twice – if so, it isn't important.

At the command line, type **draw 1, reference**. This will illustrate four spectra that are used for processing data. Two of the spectra (they're probably blue) are from flow-through cells containing mobile phase with no KI. The lowest spectrum is a normal cuvet, also with mobile phase and no KI. I call these three the "zero spectra." For analysis, the macros will subtract the zero spectra from the sample spectra to isolate the absorbance due to KI from the absorbance due to water, glass, and potassium phosphate. The other spectrum shown, with a peak at 226 nm, is a normal cuvet containing both mobile phase and KI.

<u>Time for chemistry:</u>

Load up the spectrophotometer with solutions. The first cell in the multicell transport (MCT) should contain K_2HPO_4 solution with no KI. This is used for "blank" spectra to calibrate the photodiode array. The second cell should contain the same solution as the first one (just to be sure), and the third should contain a calibration solution (K_2HPO_4 plus KI). The next two cells should contain arbitrary concentrations. It doesn't matter what concentrations you use – they can even be unknown, if you like. They'll be used to check for consistency in the measurements. The last two holes should contain the flow-through cells.

On the computer, click on the picture of the first cell, then push the **Blank** button. This calibrates the photodiode array. Next, push **Run Automation**. This will shoot spectra for the six remaining cells.

To highlight a particular spectrum, click on it or click on the left edge of the Sample Spectra Table. To zoom in on the spectra, just draw a box around the area you want to see. To zoom out, go to the View menu and select **Samples**.

Now, try analyzing the spectra. First, hit **View – Standards** (in other words, click the View menu and then Standards). Hit the **Calibrate** button. Type **jannormalize** on the command line. This is the macro that subtracts the zero spectra out of the measured spectra. . Next, hit **View – Samples.** You'll see that the spectra are lined up now. Now, hit the **Analyze** button on the Sample Spectra Table. When the computer finishes thinking, you'll see the measured concentrations of KI. Hopefully, they're approximately what they should be. It's entirely possible that they're totally different, though, since you're using my old standards to calibrate your new measurements.

Note that during an experimental run, the analysis isn't this labor-intensive. The macros (mainly **janprocess**) will take care of it for you.

Updating the calibration:

If you've done everything I told you to do above, then you already have a spectrum that will work for calibration. If you're starting from scratch, do this:

1) Select the first cell in the window with the pictures of cuvets.

2) Scan a blank (hit the **Blank** button).

3) Hit the **Run Automation** button.

4) Type **jannormalize.**

Now, we're all on the same page.

Next, copy the first spectrum into the standards register. Do this by typing the command **copyobj samples[1], standards**.

Now, view the standards (using the **View** pull-down menu). You'll see the two old spectra and your new spectrum. If you used the same concentrations that I used, then they should be pretty similar. In the concentration column – the one that reads "KI(C/Co)" – assign a value of one to your new spectrum. It should look like the line above it, in other words.

Highlight line 2 by clicking on the left edge of the line. Delete it. You should now have one spectrum that's almost a straight line and another that has two big peaks. Hit the **Calibrate** button, and you're essentially done.

Go back to the Sample Spectra Table (**View… Samples**) and hit the **Analyze** button. You should now see accurate concentration values.

Save the standards as **currstnd.std**. This part should be self-explanatory. HPChem will now load these standards automatically the next time it starts.

Another way to calibrate the machine is to shoot a large number of spectra over a long period of time and find the mean spectrum. To do this, run HPChem and LabVIEW for a while as if you were running an experiment (or it can be a real experiment, if you want). Then, load the spectra – they'll be in the directory **c:\windows\desktop\spectra\**, with the date as the name. Next, type **jsortdata**. This runs a macro that divides up the spectra and saves them in different files. It also keeps the spectra in memory. The macro will ask you for a file name; anything that doesn't exceed seven characters will do. I made the default the letter "t" followed by the date. This lets you find these files easily, because they all start with the same letter. Anyway, run the macro and then type **copyobj specs2, samples** and take a look at the samples. The last spectrum on the list is the mean spectrum for the KI cell. Copy this spectrum to the standards register and go through the process described above.

The register called "reference" (which you saw if you typed **draw 1, reference** earlier) is a collection of mean spectra that I produced using the method in the previous paragraph. A valve in the fracture junction apparatus allows you to run mobile phase (no KI) through both of the flow cells. If you fill the cells with mobile phase and run the apparatus for a while, then you can get a set of zero spectra, find their means, and use the spectra to replace my old spectra in the reference register. One way to do this is to load your new spectra into the samples register using the **copyobj** command, delete the reference register (**delreg reference**), and then type **copyobj samples, reference**.

Janprocess, or How you'll actually analyze data:

You don't actually have to go through the laborious method I described above to analyze the data, because the **janprocess** macro will do it for you.

Imagine that everything has been running and collecting data for several days. You come into the lab wanting to see the results. The first thing to do is to push the **Stop UV** button in Mainfrac (LabVIEW). As always, you'll have to use the **X** (close window) button in HPChem if the computers aren't communicating with each other.

After the data-collection macros (the ones you ran by typing **jstart**) stop running, type **janprocess** on the command line. When it asks, give it the date of the data you want to process. If you have several days' worth of data, then you'll have to run the macro once for each day. When it finishes with the first set, just hit the up arrow and **[Enter]** to repeat the command.

This macro will delete whatever spectra are currently in the sample register, load the spectra from the date of interest, determine concentration values, and record those concentration values in a file with the extension **.con**. You'll find the concentration file in the **Data** folder. To view the data, use the Excel macros described elsewhere in this manual.

When you've finished, type **jstart** to go back to collecting data. If the computers weren't communicating before, then you'll have to close HPChem and reopen it before running **jstart**. In either case, hit the **Say Hi** button in Mainfrac to make sure that the computers are communicating now.

Get to know HPChem:

A lot of HPChem is pretty self-explanatory. Familiarize yourself with what the various menus and buttons do. The **Clear** button is especially handy, and you'll find that you can change things like the automation sequence and the wavelengths used for analysis. You can also turn the lamps on and off from the **Instr** menu. This is handy when you're not using the spectrophotometer, because deuterium lamps are expensive.

When you shut down the software, tell it not to save anything – you might have changed something important in your wanderings!

Appendix C: Instructions

This appendix is meant to explain, step by step, how to prepare for an experimental run with the fracture junction model.

Mix the buffer solution. Using a one-liter Erlenmeyer flask, measure eight liters of reverse osmosis water into a large container. Weigh 13.92g K_2HPO_4 on a Mettler AT400 balance and dissolve the K_2HPO_4 in the water. This will produce eight liters of 10mM K_2HPO_4 buffer solution.

Mix the tracer solution. Using the one-liter Erlenmeyer flask, measure four liters of buffer solution into a four-liter, brown bottle. Pour the remaining buffer solution (approximately four liters) into a second bottle. In the first bottle (the more accurately measured one), dissolve .069g KI. This will produce a 0.1mM solution of KI in the buffer. You should now have four liters of buffer and four liters of buffer plus KI.

Notes: The absolute concentrations of these chemicals are not particularly important. It is vital, however, that the buffer concentrations in the two solutions are identical. This is why it is necessary to start with one large batch of buffer, rather than mixing the two solutions separately. If an eight-liter container is not available and you have to use two four-liter bottles, make sure that the solutions are identical before proceeding. Using a third bottle, combine two liters from each of the first two bottles. Then combine the remaining solution into one of the first two bottles. Shoot samples on the spectrophotometer and keep mixing until the two spectra are identical.

The brown bottles probably help prevent photosynthesis in the solutions. Even with well-filtered water, solutions will eventually turn green if exposed to sunlight in clear containers.

Load the spectrophotometer. Fill two Hellma QS cuvettes (the kind with little stirrer bars in the bottom) with buffer solution. Be sure to rinse each cuvette several times with the buffer, and don't lose the stirrer bar. I use a plastic dropper to fill the cuvettes and then suck the water back out. Place the buffer cuvettes in the first two positions on the multicell transport of the HP 8453 spectrophotometer. In the third position, place a cuvette filled with the tracer (buffer plus KI) solution. In the fourth and fifth positions, place cuvettes containing mixtures of the two solutions. These two cuvettes should be *different* from each other. The exact concentrations in the last two cuvettes are unimportant.

Make sure the two flow cells are also in the multicell transport and turn the lever to close it. There should be no empty slots.

Notes: The cuvettes will be used to calibrate the spectrophotometer during the experiments. The mixed cuvettes (the two in the fourth and fifth positions) will provide a check on the consistency of measurements. As long as they're constant, the actual concentrations in the fourth and fifth cuvettes are important.

Fill the reservoirs. In the temperature-controlled box, one reservoir (Erlenmeyer flask) is connected directly to the reservoir selector valve. Fill this reservoir with tracer solution. The other reservoir is connected to both the reservoir selector valve and the upstream computer-controlled valve. Fill this reservoir with buffer solution.

Notes: One syringe will only be able to receive the buffer solution. The other syringe, which is connected to the reservoir selector valve, will be able to receive either buffer or tracer.

Check valve settings: Using Mainfrac, the apparatus should be set to Refill Mode. The reservoir selector valve should be set to the buffer reservoir.

Notes: Before starting an experiment, the system must be run with buffer in both syringes (no tracer) in order to collect background spectra from the flow cells.

Load syringes. Before starting, the syringes must be loaded by hand. This will be easiest if the upstream syringes are almost empty, but it isn't necessary.

Remove one of the upstream syringes from the pump and pull on it, slowly but firmly, so that it fills with buffer solution. You'll probably see bubbles entering the syringe; this is normal. The more slowly you do this, the less air you'll get. The goal here is to make sure that the lines are flushed with buffer and that whatever is initially in the lines (i.e. whatever solution was last used) *does not get back into the reservoir.* Pull several milliliters of buffer through the system and don't push any of it back to the reservoir. When you think you've got enough, disconnect the syringe from the line and empty the syringe (you'll need to have an empty beaker handy).

Now that the lines are flushed, it's necessary to rinse the syringe. To do this, pull a milliliter or two of buffer into the syringe, disconnect it, point it upward, and draw air in until the plunger reaches the end of its stroke. Rather than filling the syringe with buffer, the idea here is to use a small amount of buffer to get the whole syringe wet. Empty the syringe into the beaker and repeat this two or three times.

Finally, you need to fill the syringe and place it on the pump. Draw in several more milliliters than were in the syringe before you started – in other words, extend the plunger farther than it was extended before. You probably have some air in the syringe, and it's now safe to push buffer back into the reservoir. With the syringe still connected to the line, point the syringe upward and push the bubble out. Next, lay the syringe on the pump. With the plunger braced against the pump, push the cylinder until the syringe is short enough to drop into place. Hopefully, you had enough extra buffer in the syringe that the air has now traveled all the way back to the reservoir and bubbled out of the filter stone. If not, then the air is still in the lines and you'll be seeing it again.

Repeat this process for the other upstream syringe. Check the volume of the syringes (they'll be the same) and set the volume in Mainfrac. Remember, Mainfrac can't determine the syringe volume without user input.

Notes: When rinsing and filling the syringe, after the lines have been flushed, you might want to draw buffer from a beaker, rather than through the lines. It goes faster with a beaker, but make sure that the solution in your beaker is the same as the solution in the reservoir.

Collect background spectra. In HPChem, clear all samples and start the monitoring program (type "jstart" at the command line and type some future date in the dialog box). In Mainfrac, set the Starting Pe to 50 and the Increment to –100. This will prevent the Peclet number from changing. Give it the usual sampling setup (0.25ml per sample, 8 samples per Peclet number), turn on Auto Pe, and initiate the Pe program.

Let the system run overnight and record spectra at Pe = 50. This will first flush any old fluid out of the fracture model and then produce a set of almost-identical spectra for the flow cells with buffer (and no tracer) in them. Once you've started it running, *do not open the multicell transport* for any reason.

Notes: If a flow cell moves, its spectrum will change. I think this is because it has such a small window for light to pass through. The multicell transport must not be opened after you've started collecting background spectra, because opening it will cause them to move slightly. Given enough time with the multicell transport in operation, they'll probably move anyway. For this reason, it's a good idea to repeat the collection of background spectra after you've collected experimental data.

The purpose of the background spectra is to provide a baseline for comparison to tracer spectra. If you know what a cell looks like with just buffer in it, then you can subsequently determine which part of a spectrum is due to the tracer and which isn't. This is necessary in order to measure tracer concentrations.

<u>Process background spectra.</u> After the system has run for a while (at least overnight), stop the HPChem macro (push the Stop UV button in Mainfrac) and load the spectra into HPChem. The spectra will be in the directory C:\windows\desktop\spectra, and the file names will be the dates on which they were collected. If you collected data on more than one day (i.e. overnight), load the files in chronological order and display them all at once.

The early spectra might show a lot of variation, but the late time spectra should be consistent. If you see high, scraggly spectra, that means that there were bubbles in the system (at least, I think that's what it means). If there were bubbles, hopefully they were only at early times. They usually get flushed out after a while.

Select and delete the early spectra and keep the later, consistent spectra. Select and delete using the Sample Spectra Table (boxes of text), not the window showing graphs of the spectra. You should end up with a set of spectra that form six narrow lines, with very little variation for each cell. Make sure that the resulting list begins with "K2HPO4" and that the total number of spectra is an integer multiple of six. The spectra were recorded in batches of six, and you need to keep that way or the software will get confused later.

At the command line in HPChem, type "jsortdata" and enter a file name in the dialog box. The default is a "t" followed by today's date. I don't remember why I chose a "t," but it puts the files at the bottom of the alphabetized list, which is convenient.

The sortdata macro separates the spectra into seven files. Six of the files (labeled "a" through "f") represent the six cells on the spectrophotometer, while the seventh (labeled "m") contains the mean values from the other six. Sortdata also deletes all the samples, so don't be alarmed when you see a blank screen.

Load the file of mean spectra. Its name is something like "t10-28m.sd" and it's probably the last file on the list. You should now see six spectra, one for each flow cell.

At the command line, type "jmakestandards". This macro copies the mean spectra for cuvettes 2 and 3 to the standards register, tells HPChem that their concentrations of KI are 0 and 1, respectively, and calibrates HPChem. It also saves these new standards to a file called "currstnd.std".

Clear the sample register (use the "Clear" button at the upper left corner of the display and select "Samples".) Otherwise, the mean spectra will get added to your real data when you start running the system.

Fill tracer syringe. You can now set the system up to run experiments. Stop the system, put it in Refill Mode, and turn the reservoir selector valve to connect the tracer reservoir to the pump. The syringe that's now connected to the tracer reservoir needs to be flushed and refilled, so conduct the "Load syringes" procedure (above) on this syringe.

Check the checklist. Make sure all of the following are true:

- Mainfrac is running.
- The syringe volume shown in Mainfrac matches the actual syringe volume (assuming a 10ml syringe). Remember the real value so you can double check it later (after the pump starts running), because sometimes Mainfrac gets confused.
- The Temp & Press switch in Mainfrac is on.
- The water bath is running, in order to stir the cuvettes.
- The air is on, in order to stir the reservoirs.
- The waste reservoir is *not* full.
- The constant temperature box is closed.
- The cabinet is closed.

Start the system running. Start the HPChem macros. Do this by typing "jstart" at the command line and entering a future date (i.e. this time next year) in the dialog box. The message "Estimated time of next sample: unknown" should appear on the message bar at the bottom of the HPChem window.

Set the parameters in the green box in Mainfrac. If in doubt, try Starting Pe = 100, Increment = -10, Volume / Sample = .25, and Samples / Pe = 8.

Turn on the Auto Pe and Refill on Start buttons in Mainfrac. Push the Initiate Pe Program button. The pump should start refilling when you do this. Make sure the syringe volume is still correct.

Push the Say Hi button in Mainfrac. After about six seconds, a howdy message should appear on the HPChem message bar. If this doesn't happen, it means there's a communication problem. Consult the software manual in this thesis.

Sit back and watch. The system is now running, collecting data, and analyzing the data automatically. At some point – perhaps after you've run a range of Peclet numbers – it will be a good idea to switch the reservoir selector valve back to the buffer reservoir, flush and fill the syringe, and run buffer through the flow cells again for comparison to your original background data. This is just to make sure that the flow cells haven't moved or otherwise changed their absorbance characteristics.