Stanford Geothermal Program
Interdisciplinary Research in
Engineering and Earth Science
STANFORD UNIVERSITY

# EXPERIMENTAL DETERMINATION OF THE EFFECTIVE TAYLOR DISPERSIVITY IN A FRACTURE

By

John R. Gilardi

A Report Submitted to The Department of Petroleum
Engineering of Stanford University in Partial
Fulfillment of the Requirements for the
Degree of Master of Science

June 1984

ACKNOWLEDGEMENTS

I would to thank my research advisor, Dr. R. N. Horne, for the advice, support, and encouragement he gave me throughout this work.

Many thanks are due to Jawahar Baruah for his design of the multiplexer board; to Alec Ryals for his time discussing design ideas and his excellent machine work; and to many fellow students and department staff who have helped me finish this project.

Financial support was provided by the Geothermal and Hydropower Technologies Division of the Department of Energy, Stanford-DOE Contract DE-AT03-80SF11459.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## Section 1: INTRODUCTION

Reinjection of waste hot water is commonly practiced in most geothermal fields, primarily as a means of disposal. Surface discharge of these waste waters is usually unacceptable due to the resulting thermal and chemical pollution.

Although reinjection can help to maintain reservoir pressure and fluid volume, in some cases a decrease in reservoir productivity has been observed (Horne, 1962). This is caused by rapid flow of the reinjected water through fractures connecting the injector and producers. As a result, the water is not sufficiently heated by the reservoir rock, and a reduction in enthalpy of the produced fluids is seen.

Tracer tests have proven to be valuable to reservoir engineers for the design of a successful reinjection program. By injecting a slug of tracer and studying the discharge of surrounding producing wells, an understanding of the fracture network within a reservoir can be provided.

In order to quantify the results of a tracer test, a model that accurately describes the mechanisms of tracer transport is neccessary. One such mechanism, dispersion, is like a smearing out of a tracer concentration due to the velocity gradients over the cross section of flow. If a dispersion coefficient can be determined from tracer test data, the fracture width can be estimated.

The purpose of this project was to design and construct an apparatus to study the dispersion of a chemical tracer in flow through a fracture.

## Section 2: LITERATURE SURVEY

The effects of water reinjection in geothermal systems worldwide is discussed in a paper by Horne(1983), which also includes a summary of tracer testing procedures and results.

In order to derive a model to accurately describe the transport of tracer through a fracture, the physics of dispersion must be understood. Taylor(1953) presented a classic study of dispersion in flow through a capillary tube. He showed that convective dispersion combines with transverse molecular diffusion in what we now know as "Taylor Dispersion" (Fig. 1). He showed that the tracer concentration is dispersed symmetrically about a plane that moves with the mean flow velocity. Taylor presents the equation governing the effective longitudinal dispersion:

$$\eta \frac{\partial^2 C}{\partial z^2} = \frac{\partial C}{\partial t} \qquad (1)$$

where,

C = concentration

z = translated distance = x-ut

x = distance

t = time

u = mean velocity of flow

$\eta$ = net longitudinal dispersivity (derived *for* pipe flow in Taylor's model)

The solutions to eqn (1) for different initial and boundary conditions can be found in Carslaw and Jaeger(1959). For a step input, found in Carslaw and Jaeger(1959). For a step input,

$$C = C_o + \frac{1}{2}(C_1 - C_o)\left[erfc\left[\frac{x-ut}{2(\eta t)^{1/2}}\right] + \exp^{ux} erfc\left[\frac{x+ut}{2(\eta t)^{1/2}}\right]\right] \qquad (2)$$

where,

$C_o$ = base concentration

$C_1$ = injected concentration

C = concentration at x

erfc = complimentary error function

Horne and Rodriguez(1983) used a method similar to Taylor's to derive an expression for the net longitudinal dispersivity, $\eta$, for flow in a fracture:

$$\eta = \frac{2}{105}\frac{b^2u^2}{D} \qquad (3)$$

where,

b = fracture half-width

u = mean flow velocity

D = coefficient of molecular diffusion

They also showed that, due to the effects of transverse molecular diffusion, any concentration gradients across the fracture would be equalized after a non-dimensional time, $t_D$ = 0.5 (Fig. 2), where

$$t_D = \frac{D}{b^2}t \qquad (4)$$

Fossum and Horne(1982) show how the subroutine VARPRO can be used to determine both linear and non-linear parameters from a set of experimental data. VARPRO uses a non-linear least squares method of curve fitting.[12] Fossum and Horne(1982) matched the calculated response to field data from tracer

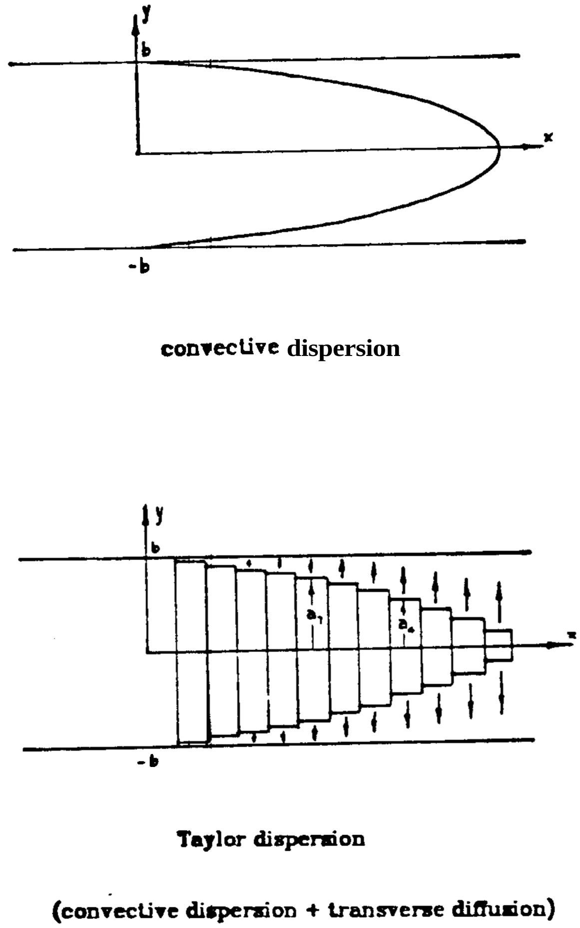



**Fig. 1.** Dispersion Schematics

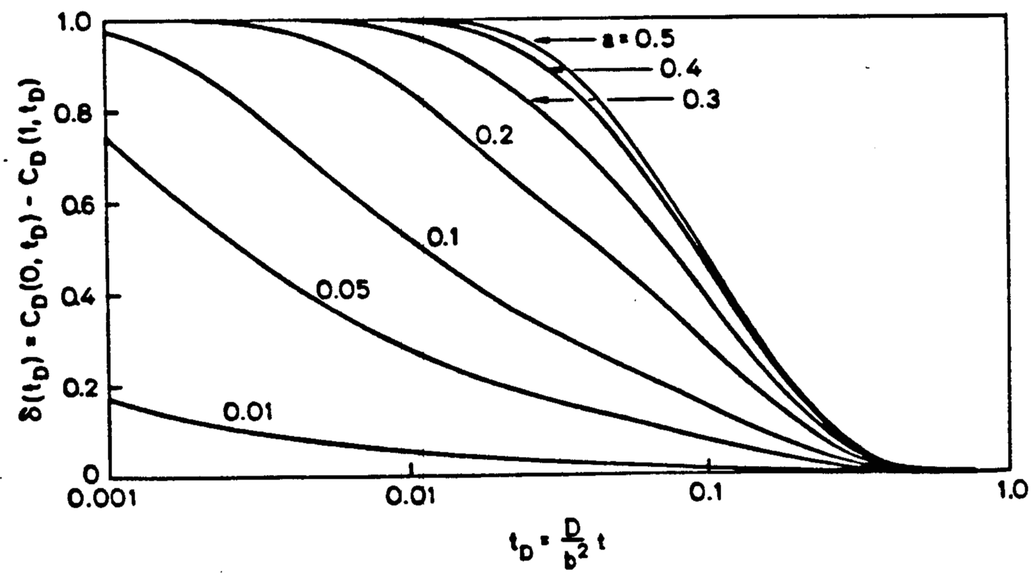



Fig. 2. $\delta(t_d)$ vs. $t_d$ (Horne & Rodriguez, **1953**)

The present study set out to examine and confirm the applicability of Equation (3), which is only approximate, and to initiate broader investigations into dispersion in fractures. To these ends, an experimental program was undertaken.

The results of several experiments to study dispersion were found in the literature. Bear(1961) performed both one- and two-dimensional studies of dispersion through porous media and produced results which agreed with his theory. Hull and Koslow(1982) present the results of a study of dispersion in a network of channels.

## Section 3: DESIGN

The design objectives were aimed at building an apparatus capable of studying dispersion through a fracture in both one- and two-dimensions. The possibility of testing both chemical and fluorescent tracers was another requirement.

### 1. HELE-SHAW CELL

The size of the model fracture, particularly its aperture, was constrained by the results of Horne and Rodriguez (1983). Any concentration gradient across the width of the fracture will be equalized after a non-dimensional time, $t_D = 0.5$ (Fig. 2). The real time it takes to become equalized is proportional to the square of the fracture half-width:

$$t = \frac{0.5b^2}{D} \tag{5}$$

Using a diffusion coefficient for potassium iodide $(KI) \simeq 2x\,10^{-9}m^2$ /sec and a fracture half-width of 0.25 mm, the time required is about 16 sec. By using an aperture of 0.54 mm and flowrates of approximately 50 cc/min we could keep the apparatus small enough to fit on a lab bench! The cell is 6 ft long by 1 ft wide. Fig. 3 shows an overall view and Fig. 4 a detail of the design.

The lower plate is 1 in. thick cast aluminum alloy. It is hard anodized to prevent corrosion and provide a tough, non-conductive finish. The upper plate is 1/4 in. float glass and is separated from the aluminum by a gasket made up of three layers of plastic electrical tape. A series of aluminum clamps holds the cell together while four adjustable legs support it horizontally on the lab bench.

### 2. VALVES

The inlet and outlet ports were implemented by drilling holes 11 in. through the width of the plate. A 0.25 *in.* slit was then sawed through the surface (Fig. 4) Since the pressure drop across the length of the drilled hole is negligible com-

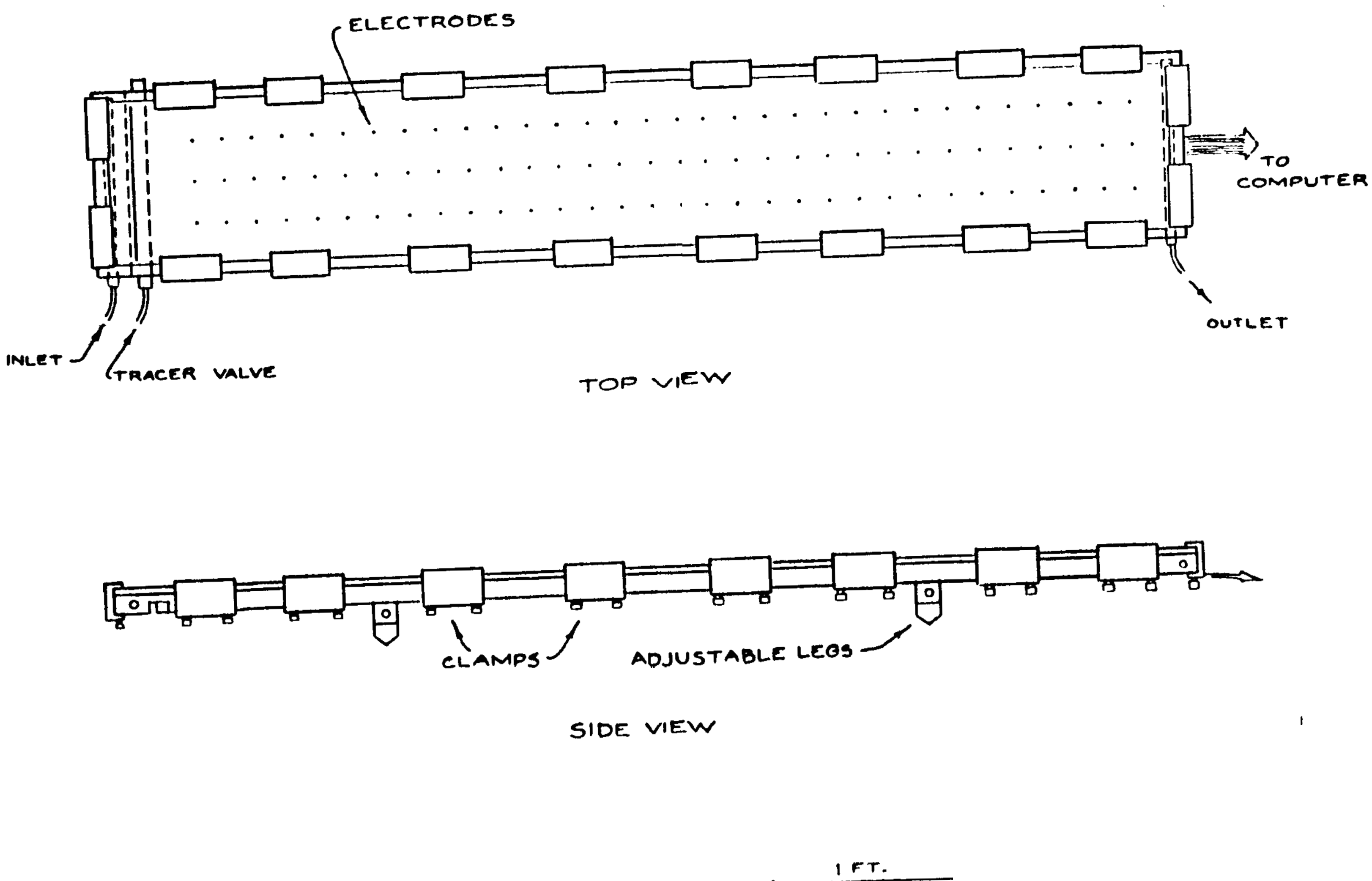



FIG. 3 HELE-SHAW CELL

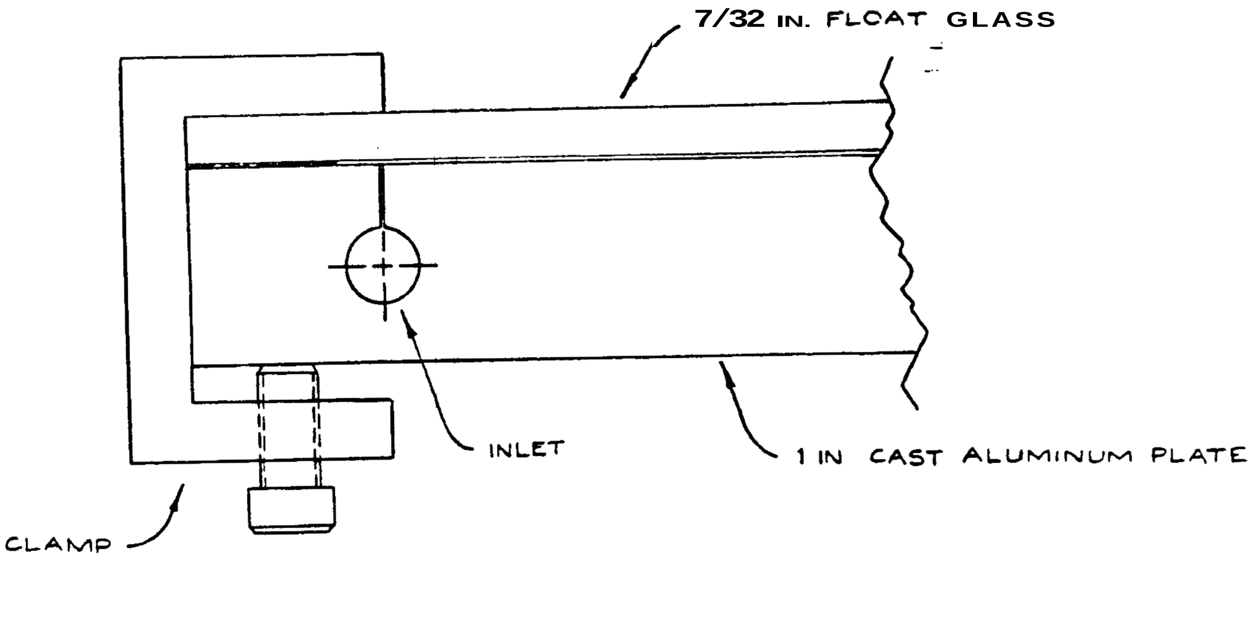



FIG. 4. DETAIL VIEW

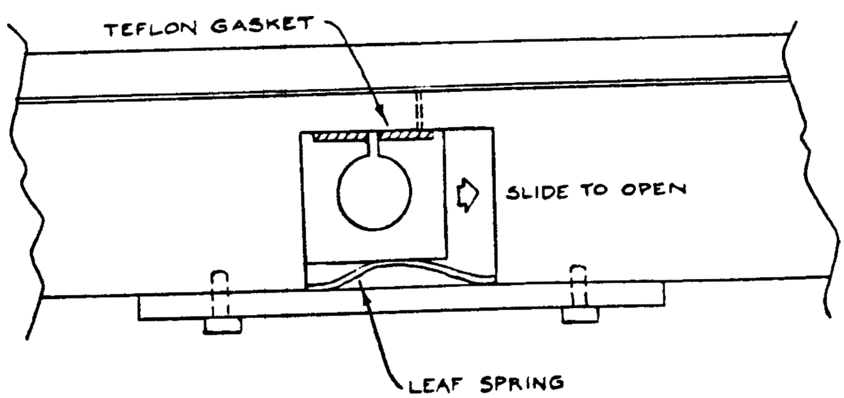



FIG. 5. TRACER VALVE

pared with that across the slit, water will flow into the cell at uniform velocity over its width.

An on-off valve (Fig. 5) was designed to allow instantaneous injection of tracer. The valve is activated by hand and can be locked in either on or off position.

### 3. ELECTRODES

In order to continuously monitor the tracer concentration as it flows through the cell, an array of 96 electrodes was employed. For a KI tracer the conductivity of solution will increase linearly with the log of concentration. Thus we are able to measure the tracer concentration at any electrode location and any chosen time.

The coaxial electrodes were constructed using brass conductive elements and a teflon insulator (Fig. 6). The brass surfaces were electroplated with gold to prevent corrosion and polarization. Each electrode is press fit in to the aluminum plate and mounted flat to within .0015 in. The central electrode is connected to the data aquisition system, and the outer electrode is grounded to the plate.

An instantaneous current is flowed across the electrode while the resulting voltage is measured (requiring less than 0.1 sec.). Voltages can be measured once each second and are stored in a Compaq personal computer. The data is displayed on the screen so it is possible to "watch" the tracer as it flows through the cell. The curcuitry and electronics are described in Appendix A, and the computer scanning algorithms are described in Appendix B.

### 4. CONSTANT FLOWRATE SOURCE

Two constant pressure reservoirs, one for the base concentration and one for the tracer, were constructed (Fig. 7). The flowrate can be adjusted by changing the height of the center **tube.**

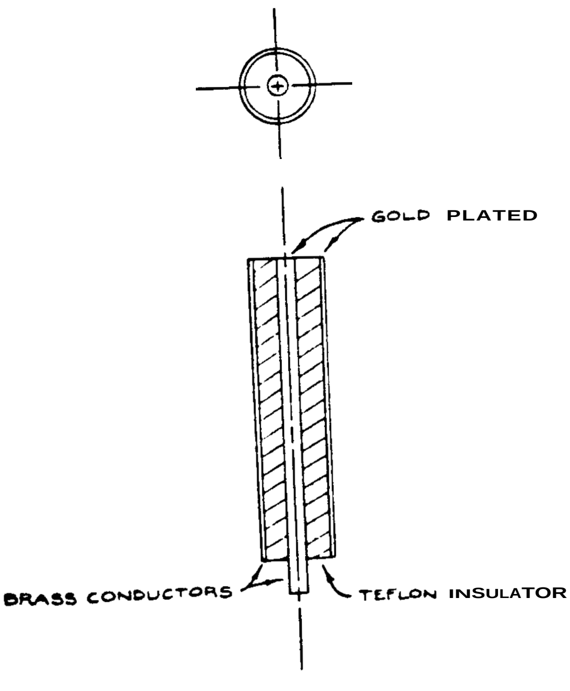



FIG. 6 ELECTRODE DESIGN ½ IN.

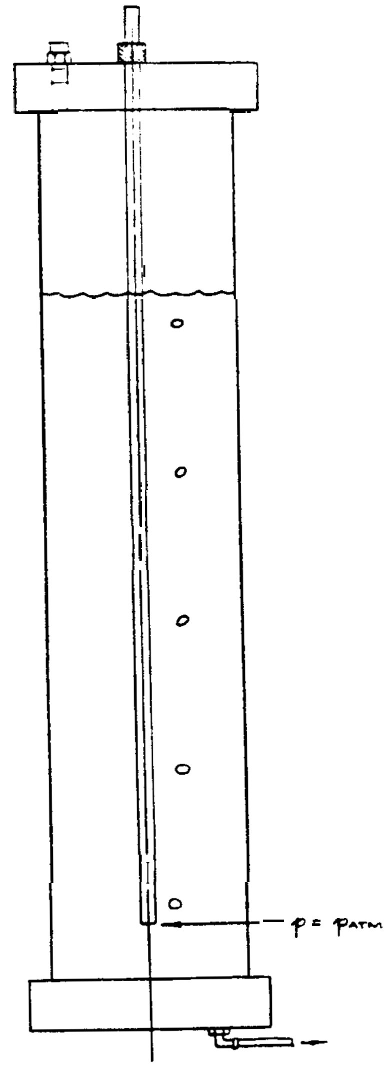



FIG 7. RESERVOIR DESIGN 1 IN.

## Section 4: PROCEDURE

I. Solution Preparation

(a) Prepare solutions of desired concentrations using distilled water and iodide standard. A base concentration of 175-200 ppm should be used. Voltage readings from lower concentrations tend to be too unstable for accurate analysis. By injecting ~300 ppm the electode response remains in the linear portion of the voltage vs. log concentration curve, thus simplifying analysis. Using the 1000 ml volumetric flask, the concentration is:

$$C(ppm) = \frac{(x)\ liter\ of\ iodide\ standard * 12690\ mg/l}{1\ liter}$$

(b) Clean reservoirs and fill with solution.

II. Assembly

(a) Wipe clean the aluminum and glass plates with wet sponge and assemble; clamps should be finger tight.

(b) Flush the Hele-Shaw cell with $CO_2$ (at p <2 psi or glass may shatter).

(c) Begin flowing water slowly, making sure to clear both the inlet and tracer valve of air, otherwise bubbles will become trapped in the cell. (distilled water was flowed until the cell was void of all air bubbles to save the prepared solutions for test experimental runs.) Pounding on the glass with the butt of your hand, or tapping the glass with the rubber mallet while flowing at high rate can prevent the water front from fingering and forming air pockets.

(d) Now start flowing the solution of base concentration; allow 2 pore volumes (~ 500 cc) to flow before starting to SCAN (see next step).

111. Prepare computer for run

(a) Plug in the multiplexor board power supply (Appendix A describes the design and operation of the multiplexor); turn on printer, then computer.

(b) When "clock" appears (~ 60 sec) hit <F10>.

(c) Type b:

(d) Remove system diskette from the A drive; replace with a blank diskette,

IV. Ready to run

(a) To begin recording, enter ***SCAN.*** Note time. SCAN will measure and store the voltage at each electrode, once per second, and will fill the C (internal) diskette after about *7* min. The voltages will be plotted against the location of each electrode on the screen (they will vary for each electrode since each **has** different sensitivity). *SCAN* **will** stop automatically after *7* min., or sooner if you hit <F1>.

(b) Allow base concentration to flow for –1 min. before injecting tracer. To inject, open gate valve, slide tracer valve into position and shut off inlet valve. Record time of injection.

(c) When run is over, enter FIXUP. FIXUP processes the data (–15 min) and stores it in a new file named labfix.dat on the A diskette.

**(d)** Enter PLOTFIX. This plots the voltage vs. time for each electrode, individually.

***(e)*** Take diskette out and label it; these are **your** results.

**(f)** To start a new run, insert a blank disc in the a drive, shut off the tracer, and flow the base concentration (~500 cc). Repeat steps 7-11.

(g) After runs are finished, disassemble and wash down thoroughly. Unplug the multiplexor board, shut off computer and printer.

**Section** *5:* **DATA PREPARATION**

The processed data (labfix.dat) can be matched to the model given by Equation **(2)** to provide estimates of the mean speed **of flow** (u), and the effective dispersivity ($\eta$). A FORTRAN program, **CURVEFIT,** performs this operation, and simultaneously calibrates the measured voltages to concentrations. The program is run as follows:

**(a)** Copy labfix.dat from drive **A** to drive C (C must be erased first).

(b) Create a file P A W S on C as follows:

1. N = number **of** unknowns (I2), [ =2 (u and $\eta$), can be **3** if $t_0$ is unknown **as** well].

2. Initial estimates of u, $\eta$; one per line (F10.4).

3. Base concentration ($C_0$) and injected concentration (C,) ; one per line (F10.4).

4. Number of electrodes to be analyzed and electrode numbers (4012); e.g. 080104052122232932 will analyze 8 electrodes - 1,4,5,21,22,23,29 and 32.

(c) Load a blank diskette in drive **A.**

(d) Type a:

**(e)** Type b:curvefit

**(f)** Type c: params <CR>, con <CR>, c:labfix.dat <CR>, prn <CR>.

**(g)** The program **will** output a summary of estimated parameters on the printer, and will create output files (1.dat through 32.dat) on **A.** These may be used **for** plotting. The **program** takes several hours to run.

**(h)** On completion, type b:plot, this will plot the results on the screen for a specified range of electrodes.

## Section 6: RESULTS

The results of seven one-dimensional experiments are presented in this section. The results from CURVEFIT are displayed in Tables 4-10, and are compared with the dispersivity predicted by Equation (3) in Figures 8-14. Graphs of the data collected from each electrode and the curve that fits it are also presented. Since it was a one-dimensional study, only the central row of 32 electrodes was used.

The actual fracture dimensions are shown in Table 1. Since the glass plate used for thes runs was slightly curved, the aperture was measured at the center-line.

Table 1. FRACTURE DIMENSIONS (cm)

| Aperture | width | length |
|---|---|---|
| 0.0515 | 24.46 | 179.1 |

Table 2 shows the concentrations of the base, ($C_0$), and injected ($C_1$) solutions for each run. The flowrate was measured by recording the time required to fill a 50 ml flask at the outlet. The time of injection, $t_0$, was only recorded for runs 8,9, and 10.

Table 2. MEASURED DATA

| Run # | $C_0$ (ppm) | $C_1$ (ppm) | $q$ (cm$^3$/sec) | $t_0$ (sec) |
|---|---|---|---|---|
| 3 | 203 | 305 | 0.80 | – |
| 4 | 203 | 305 | 1.25 | – |
| 6 | 177 | 305 | 0.82 | – |
| 7 | 177 | 305 | 1.38 | – |
| 8 | 177 | 305 | 0.66 | 60 |
| 9 | 177 | 305 | 0.48 | 42 |
| 10 | 177 | 305 | 0.57 | 43 |

The velocities listed in Table 3 are values from CURVEFIT that best match the data (since the fracture cross-section did not have constant aperture, the flow velocity was not constant over the width of the cell, and $u = q/A$ may not be accurate). These best-fit values are used in Equation (3) to provide an estimate of the dispersivity ($\eta$). The estimated times of injection, $t_o$, used in PARAMS are also listed in Table 3.

Table 3. ESTIMATED VALUES

| Run | u (cm/sec) | $\eta$ ($cm^2$/sec) | $t_o$ (sec) |
|---|---|---|---|
| 3 | 0.542 | 0.210 | 73.0 |
| 4 | 0.824 | 0.486 | 66.6 |
| 6 | 0.539 | 0.208 | 79.5 |
| 7 | 0.890 | 0.567 | 73.4 |
| 8 | 0.451 | 0.146 | 60.0 |
| 9 | 0.375 | 0.101 | 42.5 |
| 10 | 0.416 | 0.124 | 42.5 |

The dispersivities predicted by Equation (3) are compared with those estimated by CURVEFIT in Figures 8-14. Only selected electrodes are plotted, as some are inconsistent due to faulty data collection or transmission. Electrodes 1 and 2 were usually inconsistent, probably because the tracer front had not yet become equalized.

| Table 4. RUN 3 Results | | | | | |
|---|---|---|---|---|---|
| Electrode | $C_0$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\eta$ ($cm^2$/sec) | apparent $t_0$ (sec) |
| 1 | 193.6091 | 305.1784 | 0.6992 | 1.0367 | 72.9238 |
| 2 | 201.2811 | 304.9338 | 0.6131 | 0.6220 | 72.6622 |
| 3 | 202.9373 | 305.0957 | 0.5908 | 0.4181 | 72.1451 |
| 4 | 200.0074 | 300,0581 | 0.5835 | 0.3585 | 72.6364 |
| 6 | 200.0959 | 300.0331 | 0.6846 | 0.2718 | 72.4763 |
| 6 | 202.9865 | 305.0680 | 0.5716 | 0.3113 | 72.3291 |
| 7 | 200.8080 | 449.3800 | 0.3034 | 2.8423 | 73.1011 |
| 8 | 199.9427 | 300.0960 | 0.5661 | 0.3005 | 73.4000 |
| 8 | 203.0065 | 305.0846 | 0.6711 | 0.2633 | 72.6024 |
| 10 | 203.0301 | 305.0726 | 0.6663 | 0.2030 | 72.8913 |
| 11 | 203.0107 | 305.1142 | 0.6647 | 0.3166 | 72.6204 |
| 12 | 200.0911 | 300.0483 | 0.6638 | 0.2449 | 73.4000 |
| 13 | 203.0176 | 305.1285 | 0.5593 | 0.2173 | 72.4066 |
| 14 | 203.0000 | 305.1503 | 0.5545 | 0.2358 | 73.1522 |
| 15 | 203.0166 | 305.1168 | 0.5543 | 0.2118 | 72.9312 |
| 16 | 203.0227 | 305.1229 | 0.5513 | 0.2623 | 72.9191 |
| 18 | 203.0147 | 305.1194 | 0.5498 | 0.2511 | 72.0139 |
| 18 | 203.0314 | 305.1045 | 0.5463 | 0.2341 | 72.6125 |
| 20 | 203.0683 | 305.1092 | 0.5455 | 0.2260 | 72.0204 |
| 21 | 203.0394 | 305.1175 | 0.5428 | 0.2326 | 72.6915 |
| 22 | 202.9910 | 305.1639 | 0.5442 | 0.2481 | 72.8799 |
| 23 | 202.8789 | 305.1857 | 0.6401 | 0.2652 | 72.8374 |
| 24 | 203.0310 | 305.1272 | 0.5410 | 0.2411 | 72.8148 |
| 25 | 202.9533 | 305.2214 | 0.5405 | 0.2708 | 72.2095 |
| 26 | 203.0517 | 305.1082 | 0.5406 | 0.1829 | 72.8609 |
| 27 | 203.0475 | 305.1186 | 0.5409 | 0.2078 | 72.5939 |
| 28 | 203.0491 | 305.1325 | 0.5416 | 0.1899 | 72.5580 |
| 28 | 203.0428 | 305.1451 | 0.5429 | 0.1909 | 72.8735 |
| 30 | 203.0459 | 305.1377 | 0.5435 | 0.1865 | 72.8586 |
| 31 | 203.0372 | 305.1767 | 0.5450 | 0.1787 | 72.3092 |
| 32 | 203.0167 | 305.2894 | 0.5471 | 0.2055 | 72.1761 |

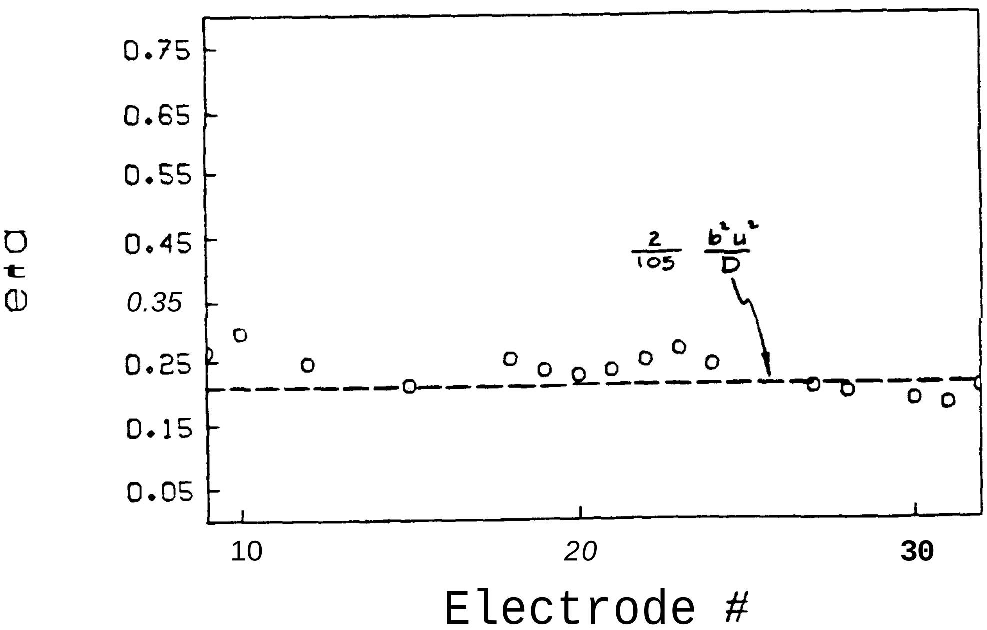
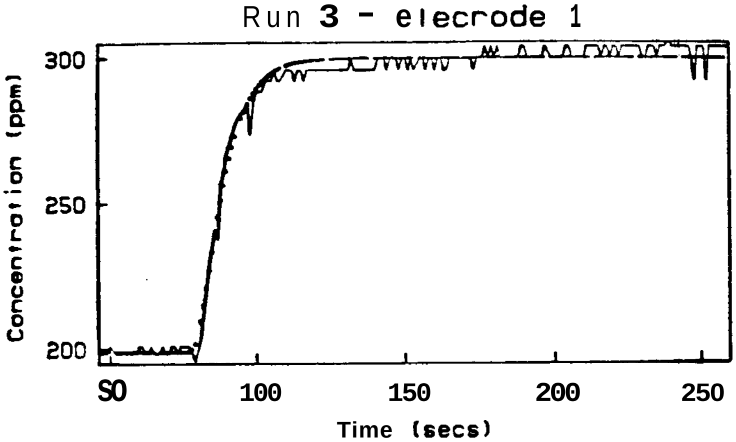
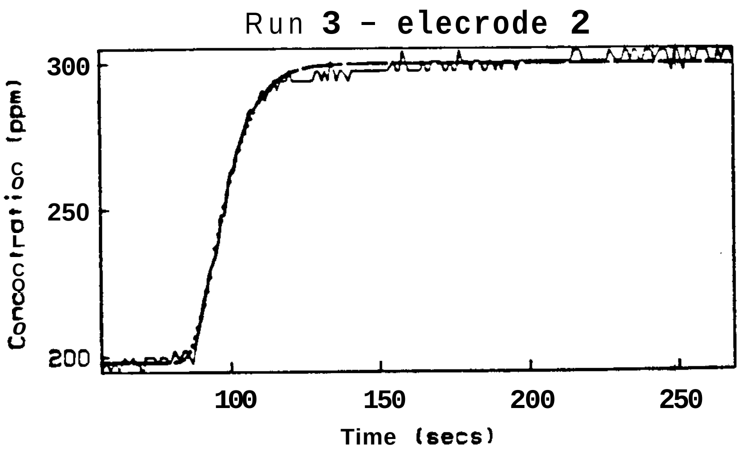
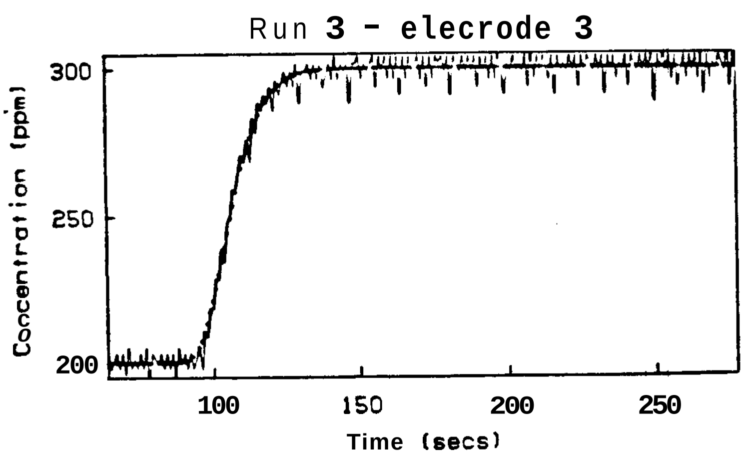
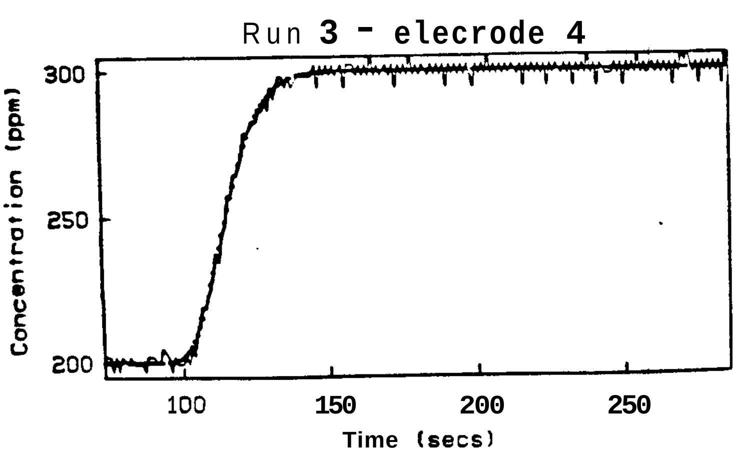


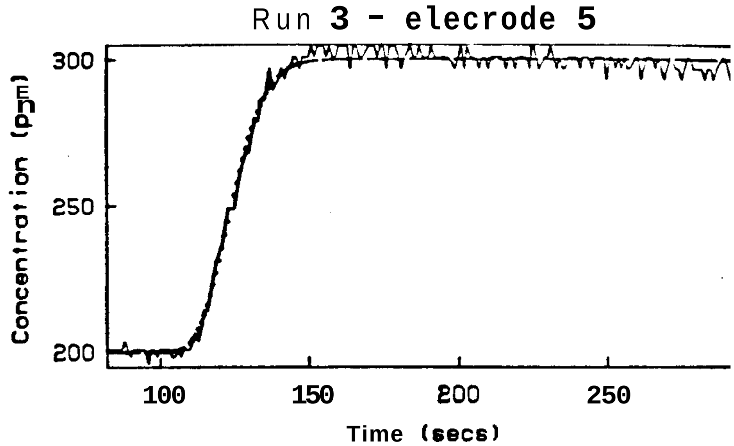

**Fig. 8. Comparison of estimated and calculated dispersivity.** Run **3.**

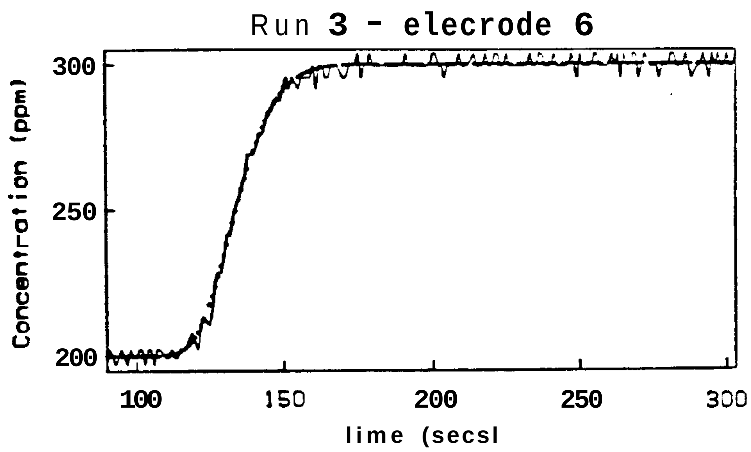

Run 3 - elecrode 1
Concentration (ppm)
300
250
200
SO
100
150
200
250
Time (secs)

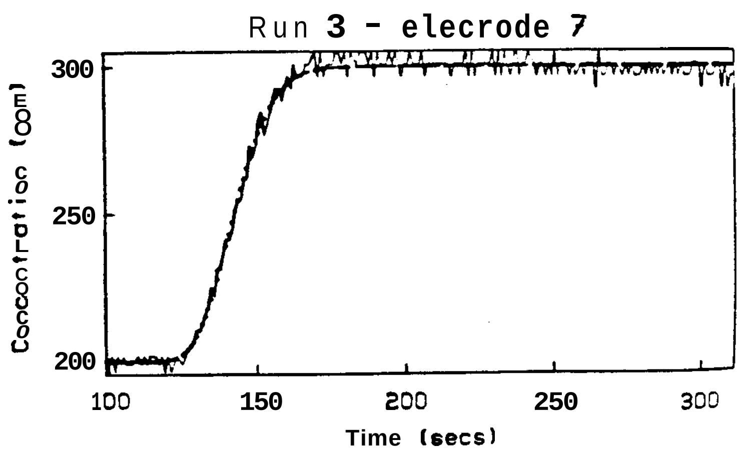

Run 3 - elecrode 2
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

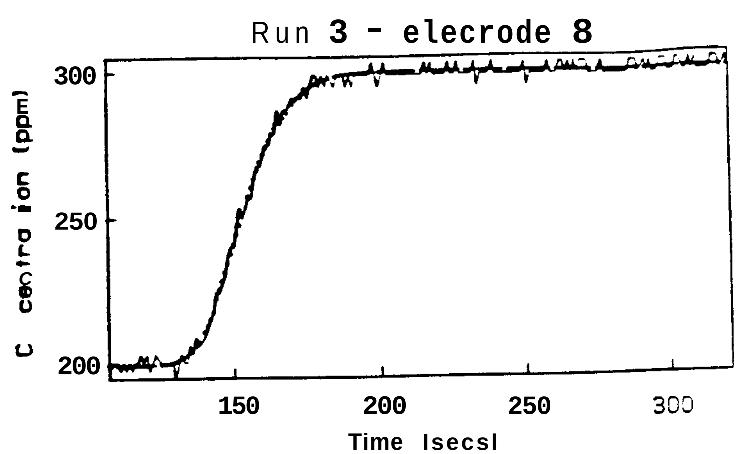

Run 3 - elecrode 3
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)


Run 3 - elecrode 4
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

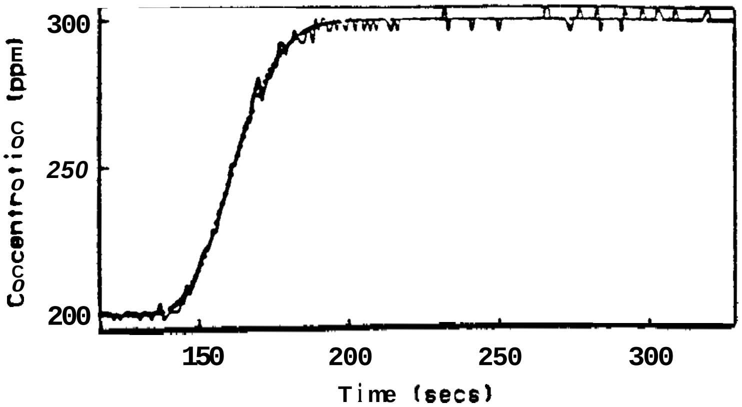

Run 3 - elecrode 5
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)


Run 3 - elecrode 6
Concentration (ppm)
300
250
200
100
150
200
250
300
lime (secsl

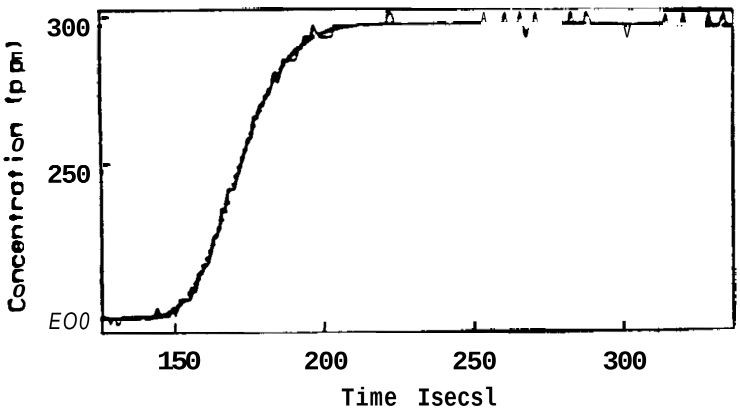

Run 3 - elecrode 7
Concentration (ppm)
300
250
200
100
150
200
250
300
Time (secs)


Run 3 - elecrode 8
C centra ion (ppm)
300
250
200
150
200
250
300
Time Isecsl

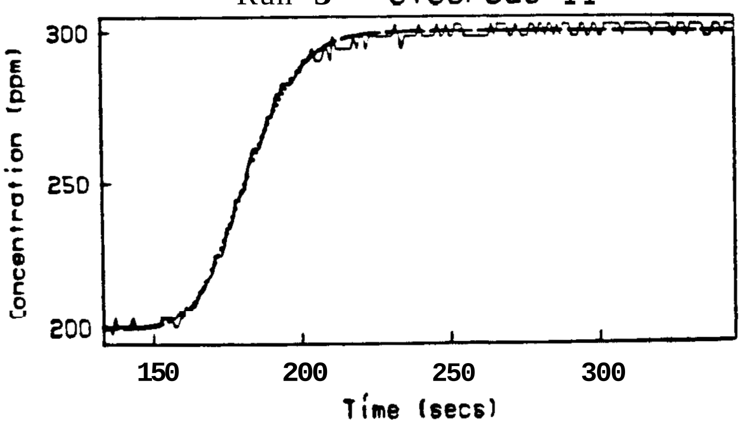
Run 3 - elecrode 9



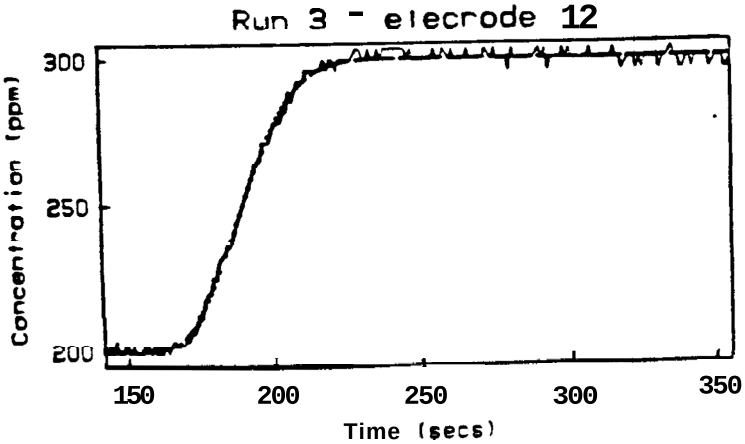

Run 3 - elecrode 10



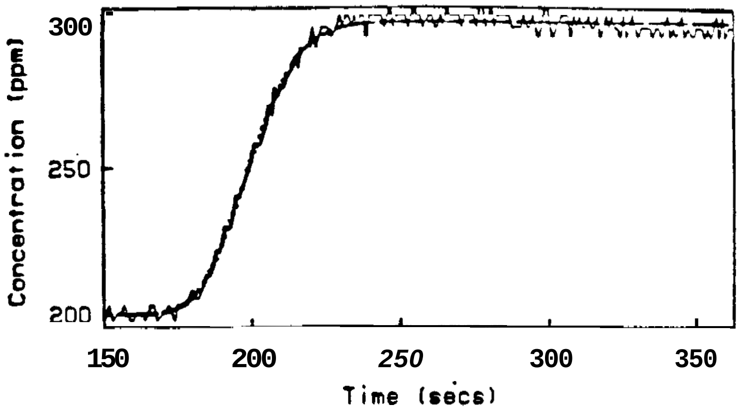

Run 3 - elecrode 11



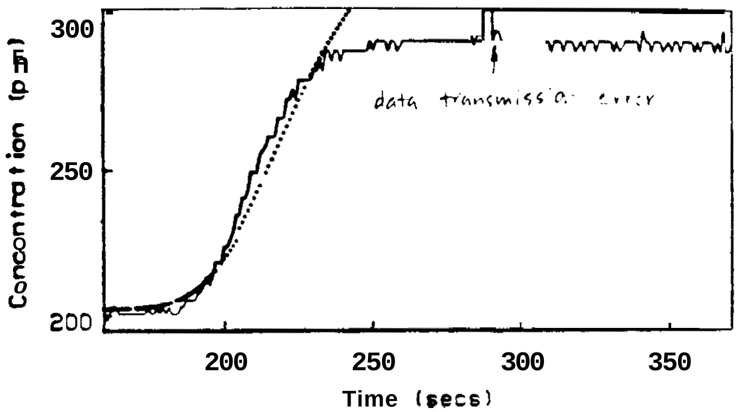

Run 3 - elecrode 12



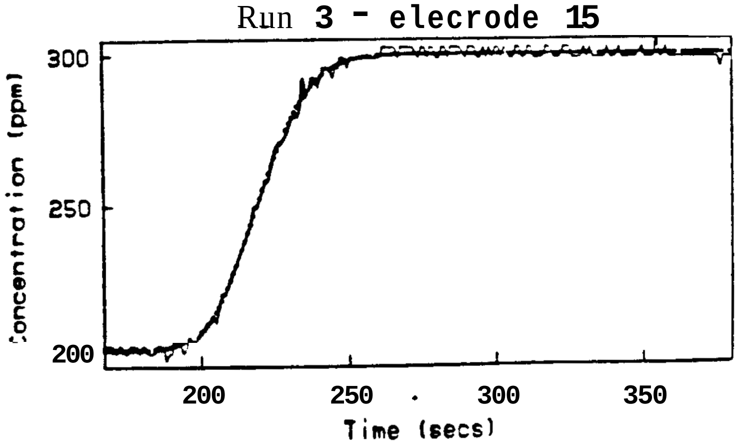

Run 3 - elecrode 13



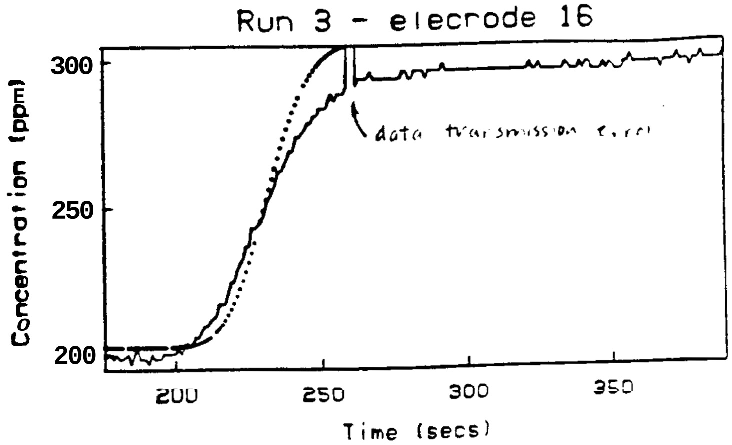

Run 3 - elecrode 14



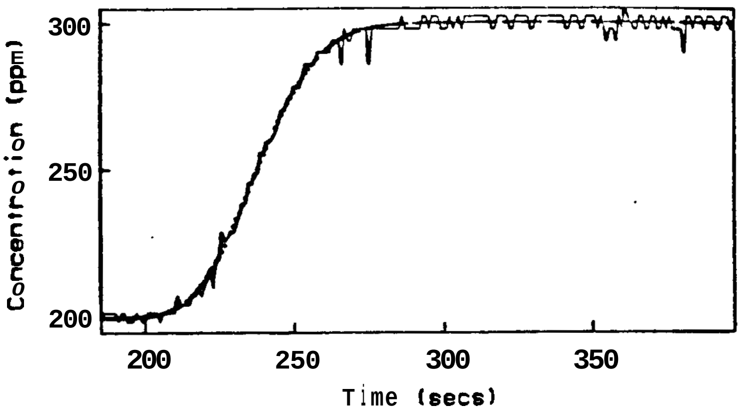

Run 3 - elecrode 15



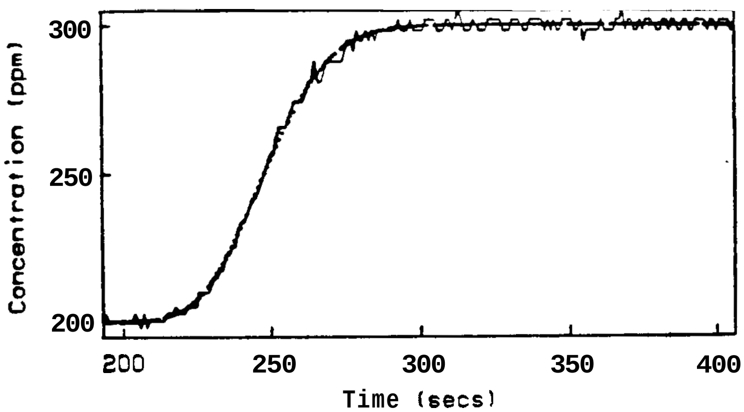

Run 3 - elecrode 16



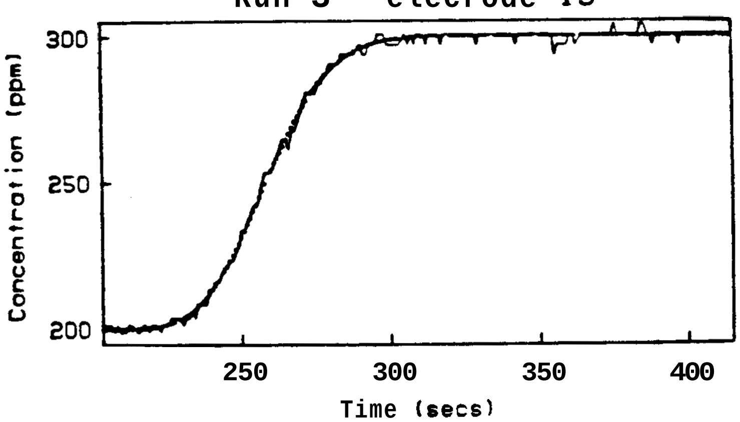
Run 3 - elecrode 17



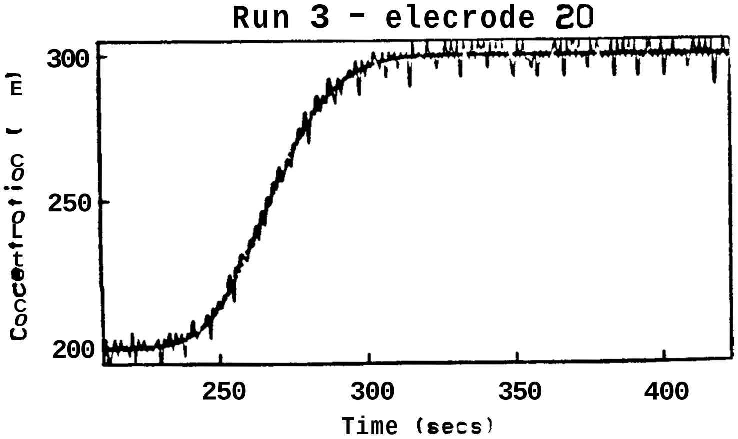

Run 3 - elecrode 18



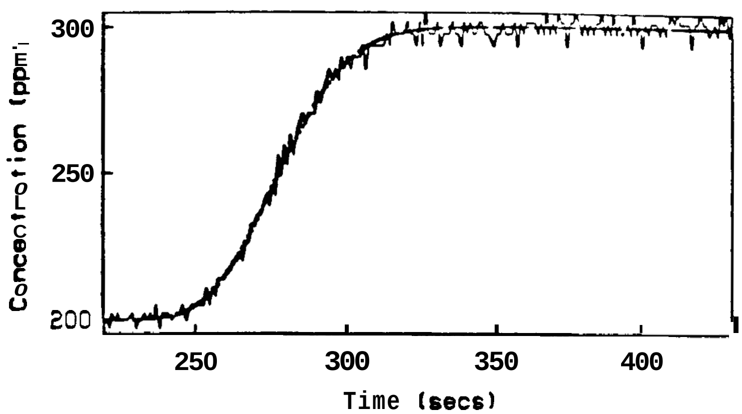

Run 3 - elecrode 19



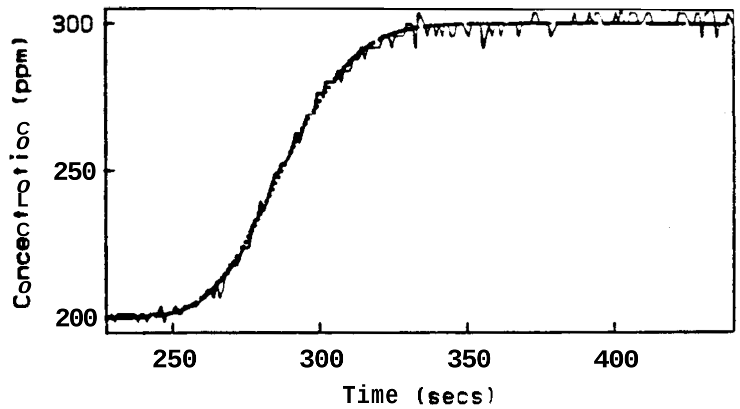

Run 3 - elecrode 20



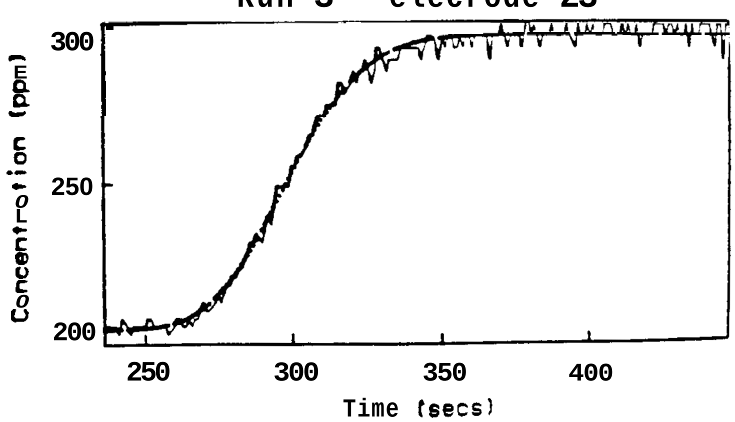

Run 3 - elecrode 21



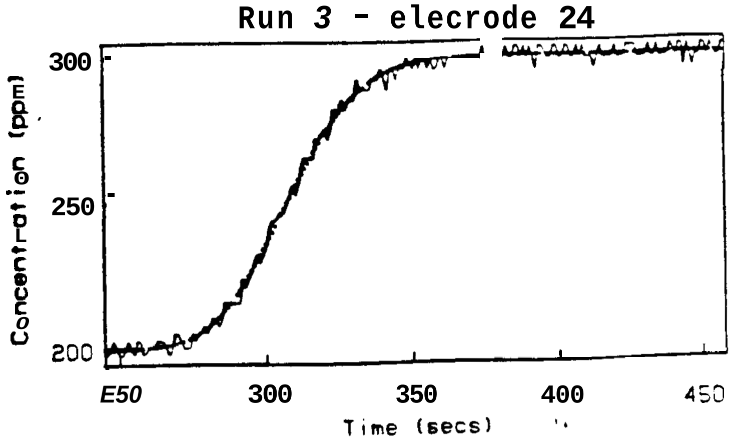

Run 3 - elecrode 22



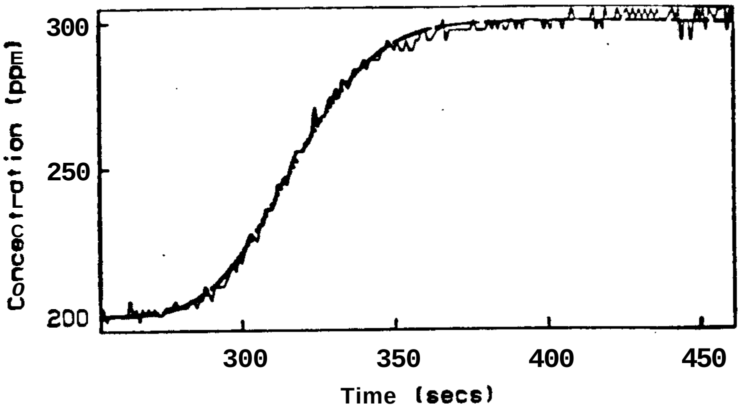

Run 3 - elecrode 23



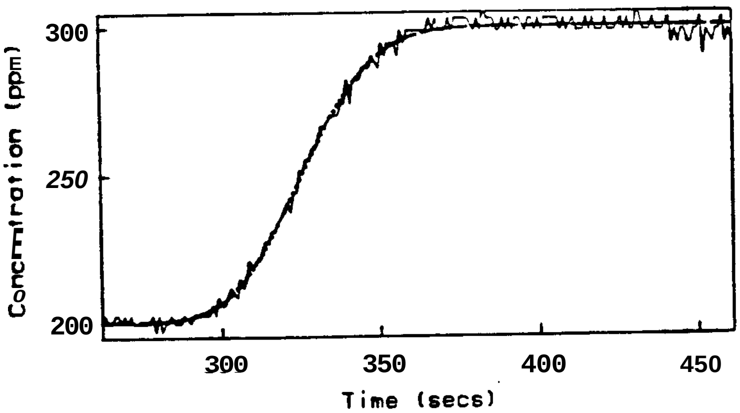

Run 3 - elecrode 24



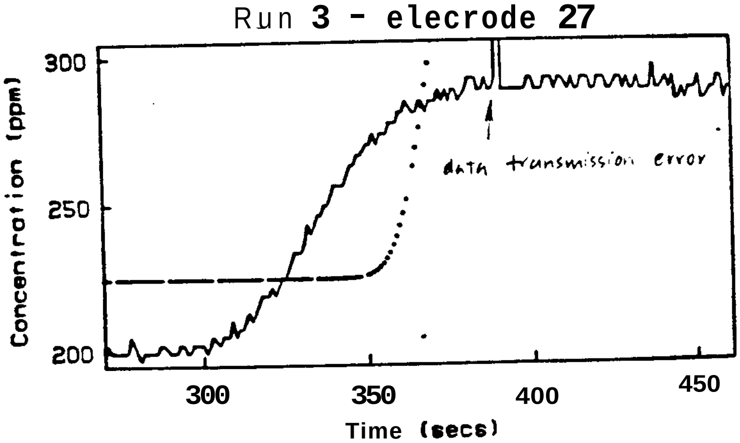
Run 3 - elecrode 25



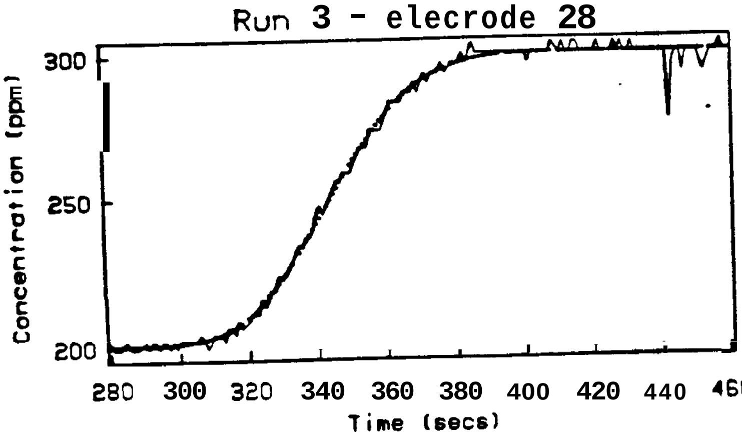

Run 3 - elecrode 26



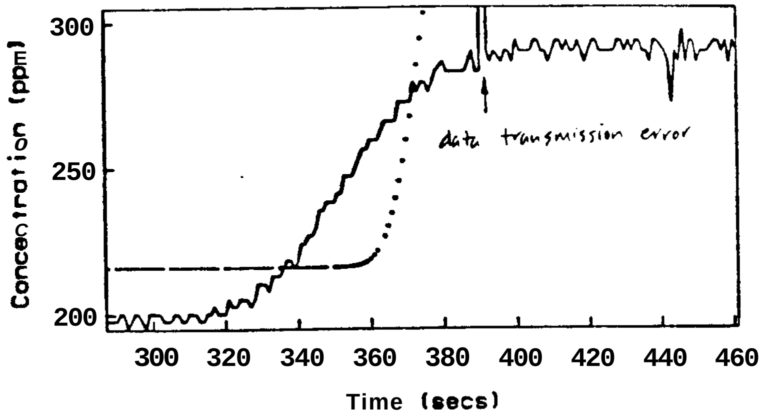

Run 3 - elecrode 27



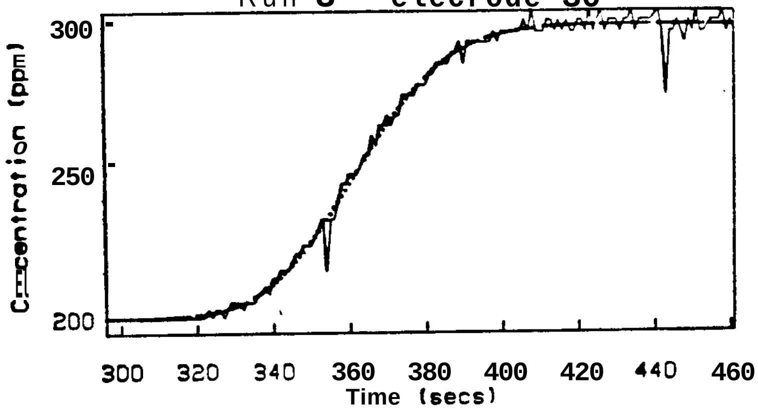

Run 3 - elecrode 28



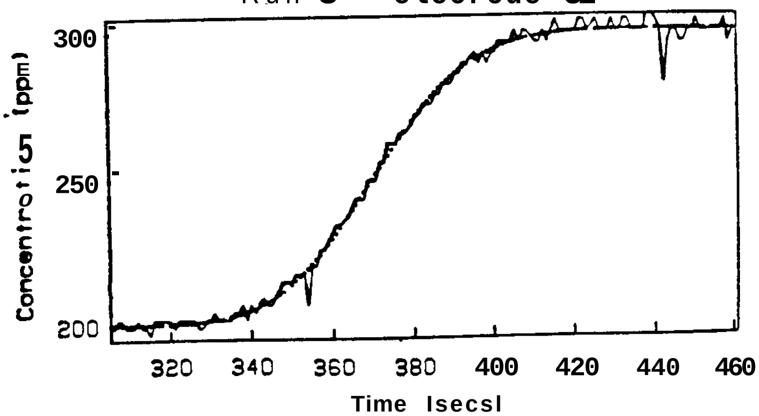

Run 3 - elecrode 29



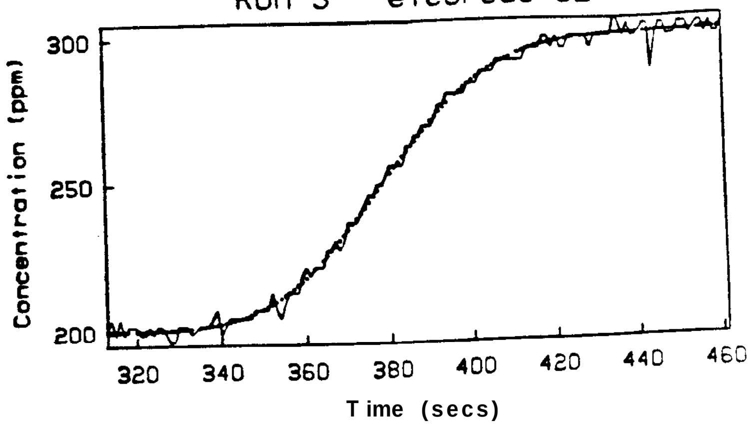

Run 3 - elecrode 30




Run 3 - elecrode 31




Run 3 - elecrode 32



| Table 6. RUN 4 Results | | | | | |
|---|---|---|---|---|---|
| electrode | $C_o$ (ppm) | $C_1$ (ppm) | u(cm/sec) | $\eta$ ($cm^2/sec$) | apparent $t_o$ (sec) |
| 1 | 199.2807 | 300.1214 | 0.7671 | 0.9746 | 66.6000 |
| 2 | 196.0567 | 300.9046 | 0.6923 | 2.2386 | 66.6000 |
| 3 | 200.0594 | 300.0597 | 0.8099 | 0.5473 | 66.6000 |
| 4 | 200.2185 | 300.0896 | 0.8563 | 0.2747 | 66.6000 |
| 5 | 200.0233 | 300.0435 | 0.8318 | 0.4877 | 56.6000 |
| 6 | 200.0099 | 300.0408 | 0.8148 | 0.5452 | 66.6000 |
| 7 | 199.6614 | 299.8199 | 0.8558 | 0.3780 | 66.6000 |
| 8 | 200.0930 | 300.0335 | 0.8297 | 0.4787 | 66.6000 |
| 8 | 199.8557 | 300.0919 | 0.8164 | 0.7602 | 66.6000 |
| 10 | 199.8767 | 300.0824 | 0.8122 | 0.6532 | 56.6000 |
| 11 | 199.8754 | 300.0808 | 0.8163 | 0.7479 | 66.6000 |
| 12 | 199.9134 | 300.0852 | 0.8305 | 0.6706 | 66.6000 |
| 13 | 200.0458 | 300.0534 | 0.8392 | 0.6296 | 66.6000 |
| 14 | 199.3993 | 299.4178 | 0.8236 | 0.6332 | 66.6000 |
| 16 | 200.1344 | 300.0222 | 0.8343 | 0.4576 | 56.6000 |
| 16 | 200.0068 | 300.0674 | 0.8255 | 0.6355 | 66.6000 |
| 17 | 200.0289 | 300.0669 | 0.8352 | 0.6581 | 66.6000 |
| 18 | 200.0545 | 300.0576 | 0.8299 | 0.5820 | 66.6000 |
| 19 | 199.9969 | 300.0705 | 0.8276 | 0.6619 | 66.6000 |
| 20 | 200.0546 | 300.0801 | 0.8263 | 0.6019 | 66.6000 |
| 21 | 200.3710 | 300.0504 | 0.8425 | 0.3211 | 66.6000 |
| 22 | 200.0699 | 300.0692 | 0.8275 | 0.4723 | 66.6000 |
| 23 | 200.0753 | 300.0618 | 0.8253 | 0.4432 | 66.6000 |
| 24 | 200.1798 | 300.0337 | 0.8296 | 0.3778 | 66.6000 |
| 26 | 200.1206 | 300.0454 | 0.8245 | 0.4633 | 66.6000 |
| 26 | 200.0903 | 300.0614 | 0.8268 | 0.4436 | 66.6000 |
| 27 | 200.0851 | 300.0662 | 0.8276 | 0.4348 | 66.6000 |
| 28 | 200.1322 | 300.0412 | 0.8281 | 0.4114 | 66.6000 |
| 29 | 200.1982 | 300.0331 | 0.8323 | 0.3844 | 66.6000 |
| 30 | 200.0716 | 300.0772 | 0.8316 | 0.4726 | 66.6000 |
| 31 | 200.1124 | 300.0662 | 0.8383 | 0.4417 | 66.6000 |
| 32 | 200.1661 | 300.0574 | 0.8388 | 0.4319 | 66.6000 |

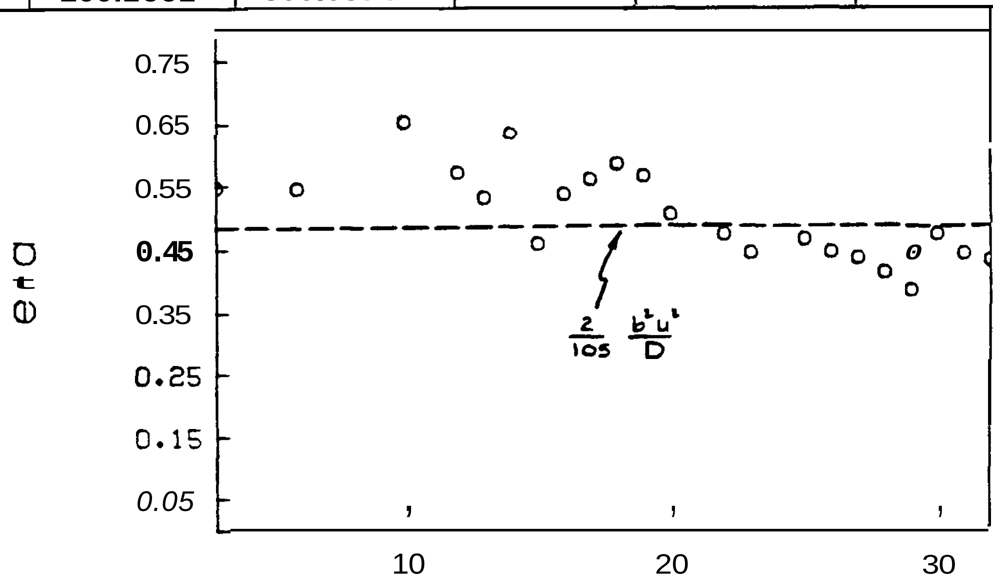
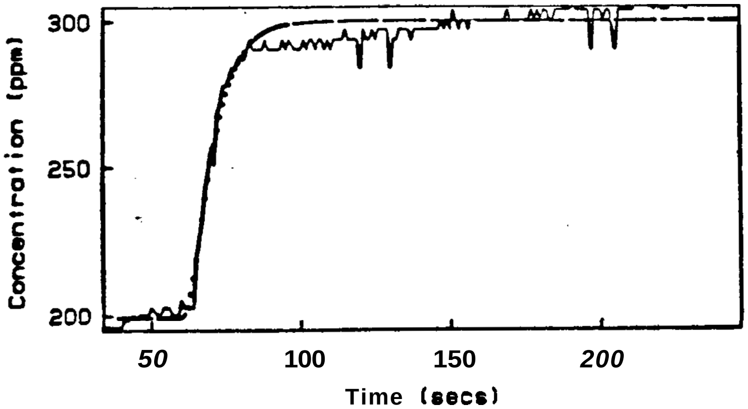
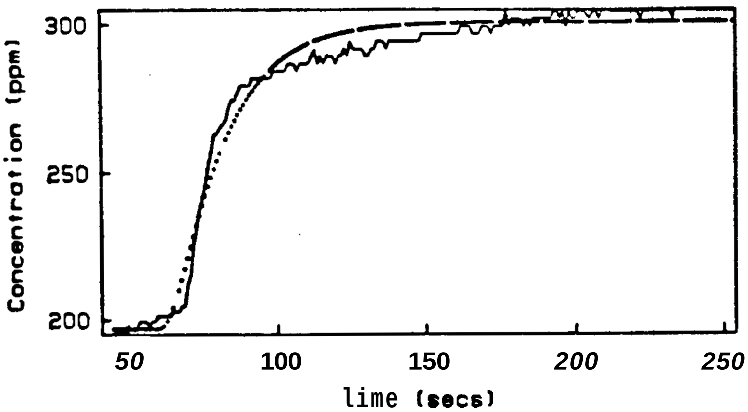



Fig. 9. Comparison of estimated and calculated dispersivity. Run 4.

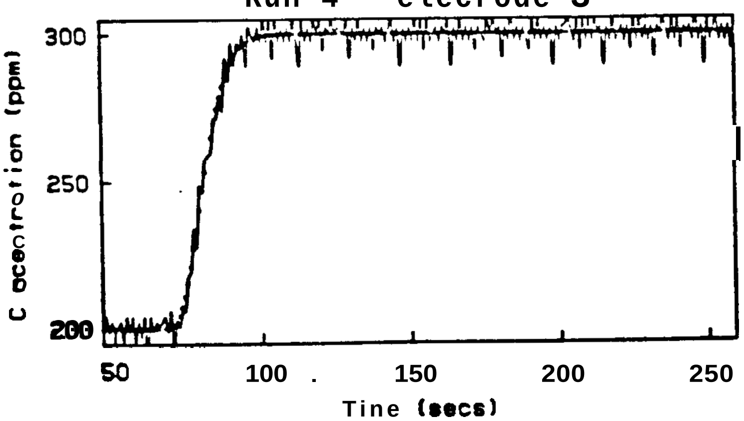
Run 4 - elecrode 1



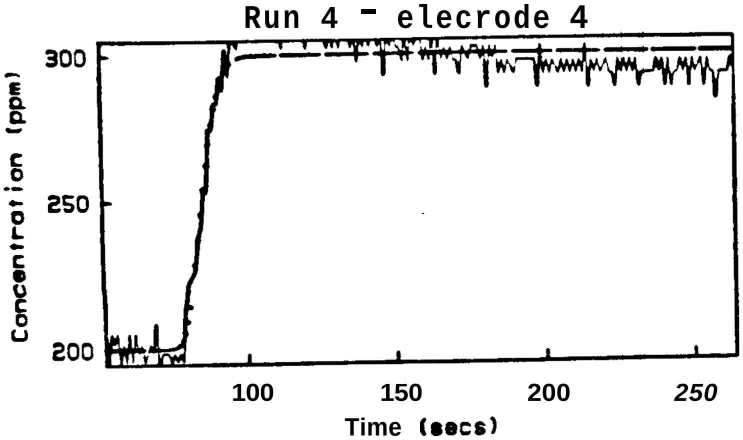

Run 4 - elecrode 2



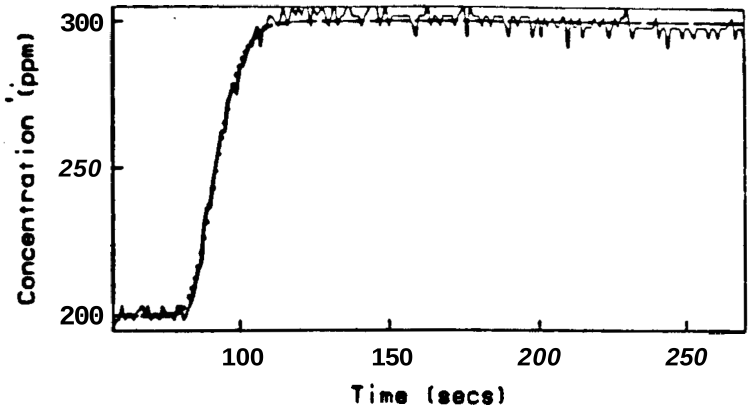

Run 4 - elecrode 3



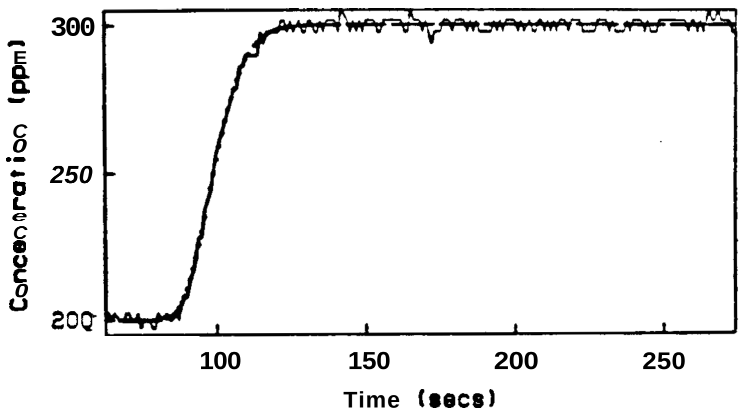

Run 4 - elecrode 4



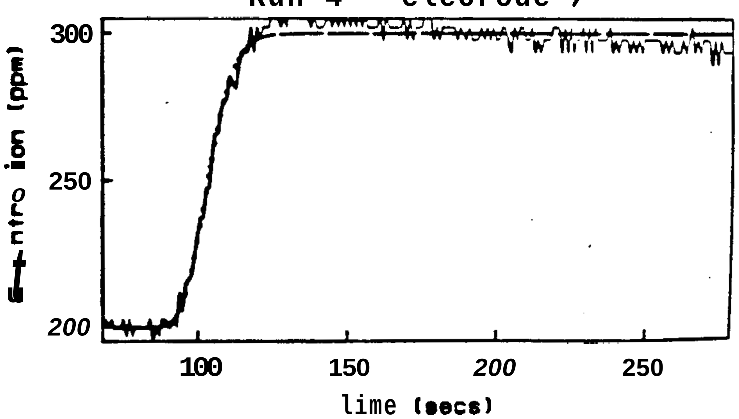

Run 4 - elecrode 5



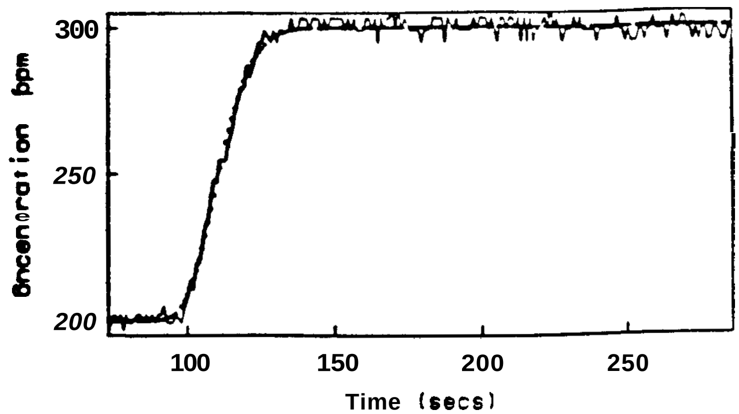

Run 4 - elecrode 6



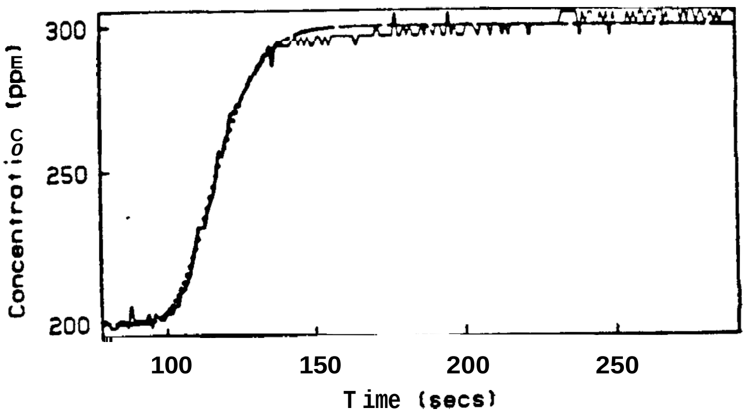

Run 4 - elecrode 7



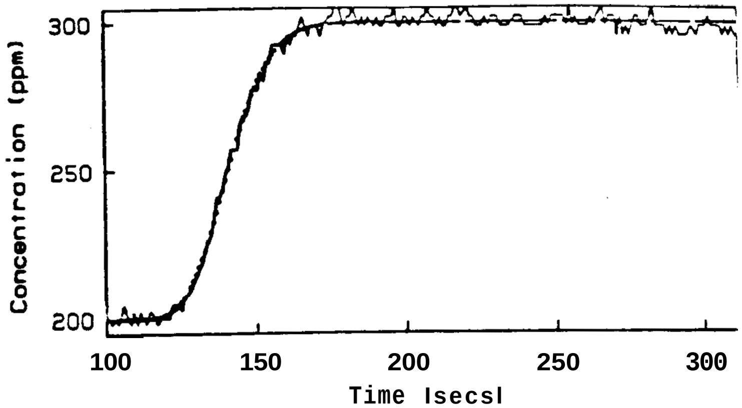

Run 4 - elecrode 8



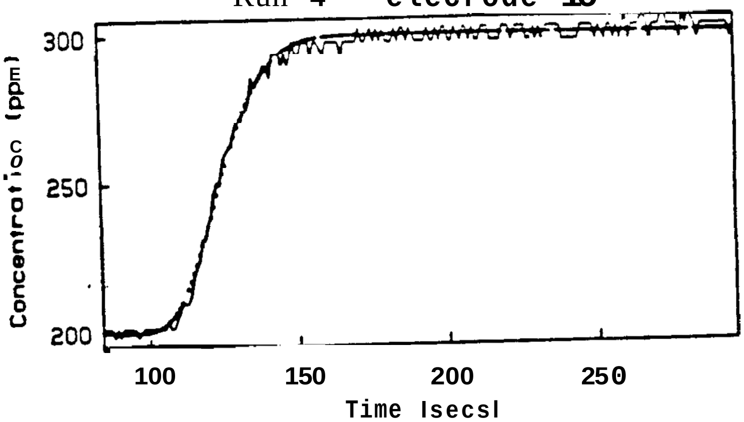
Run 4 - etecrode 9



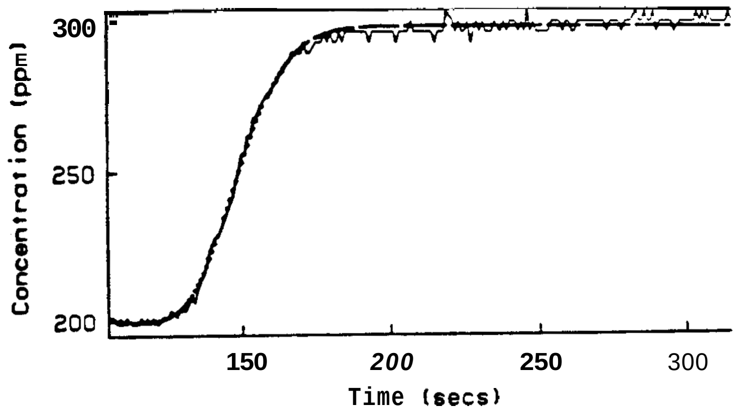

Run 4 - elecrode 13



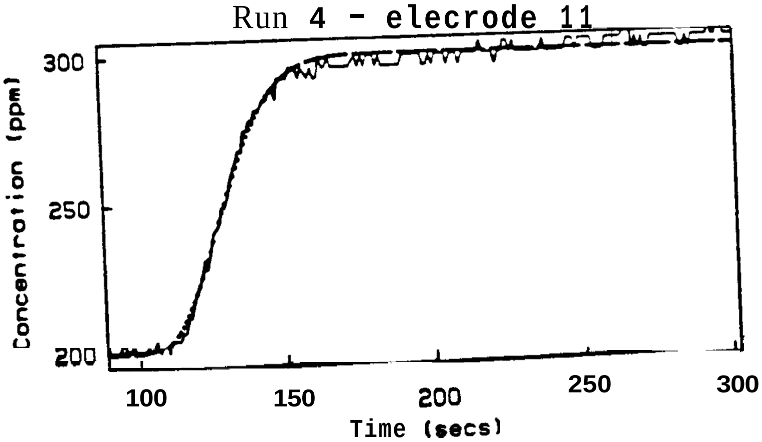

Run 4 - elecrode 10



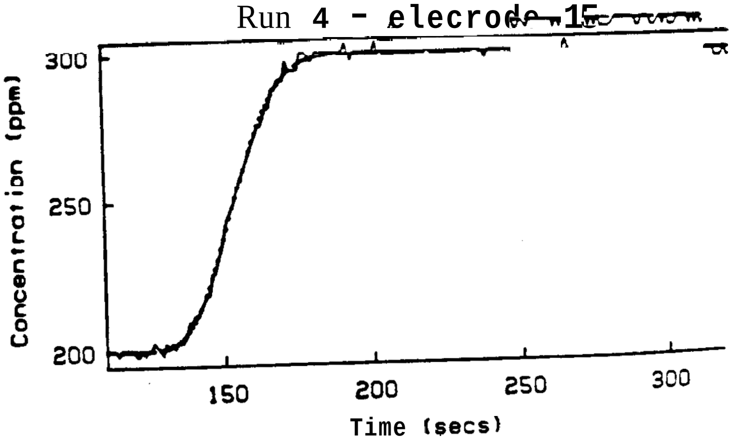

Run 4 - etecrode 14



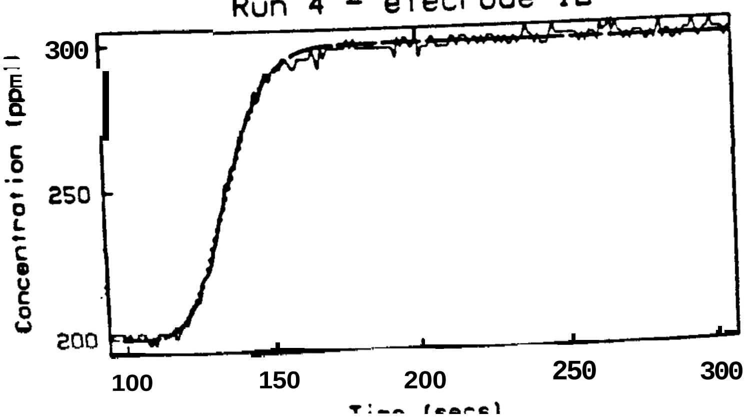

Run 4 - elecrode 11



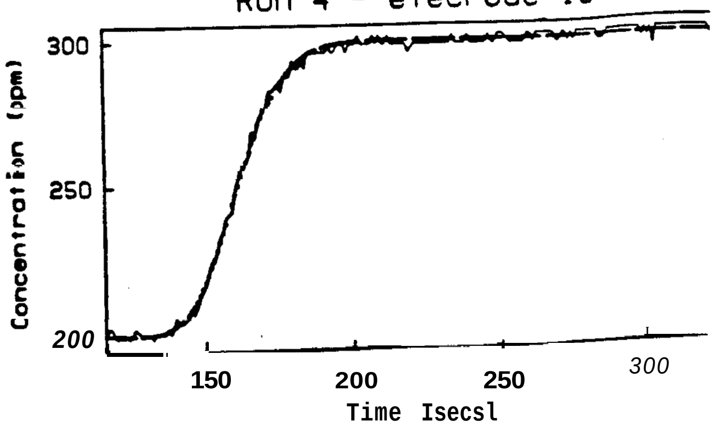

Run 4 - elecrode 15



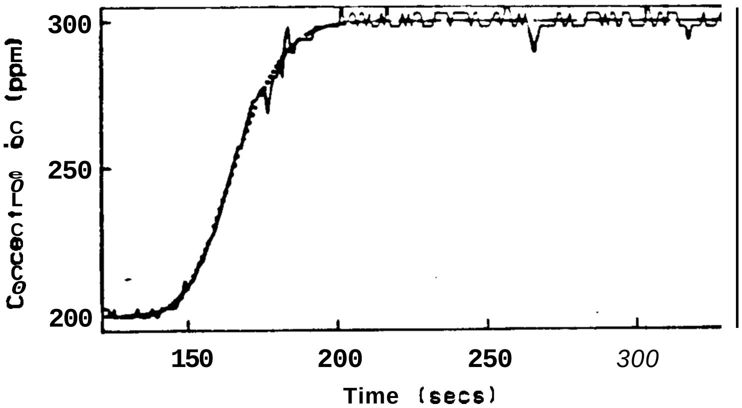

Run 4 - elecrode 12



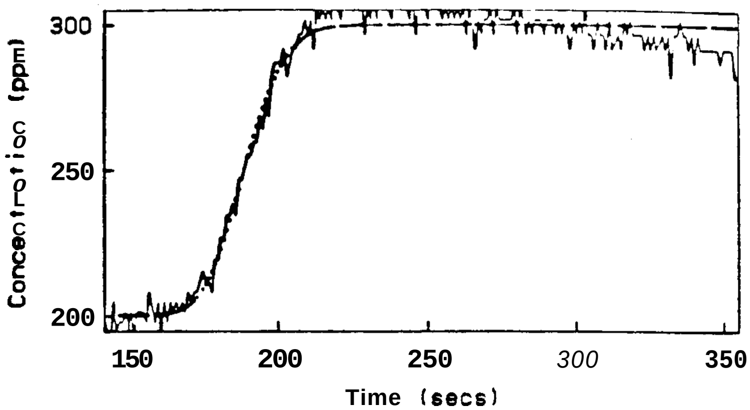

Run 4 - elecrode 16



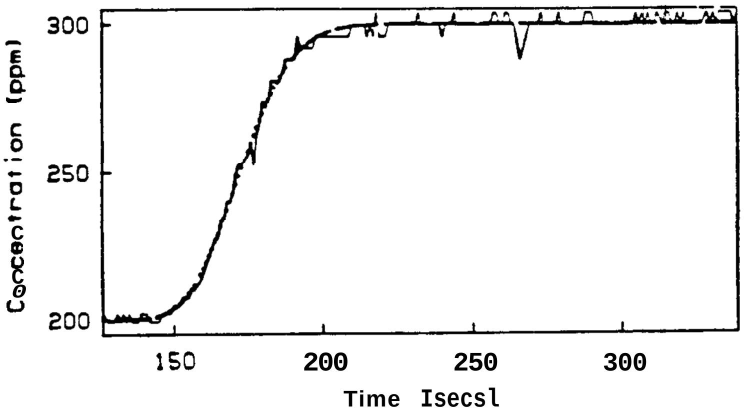
Run 4 - elecrode 17



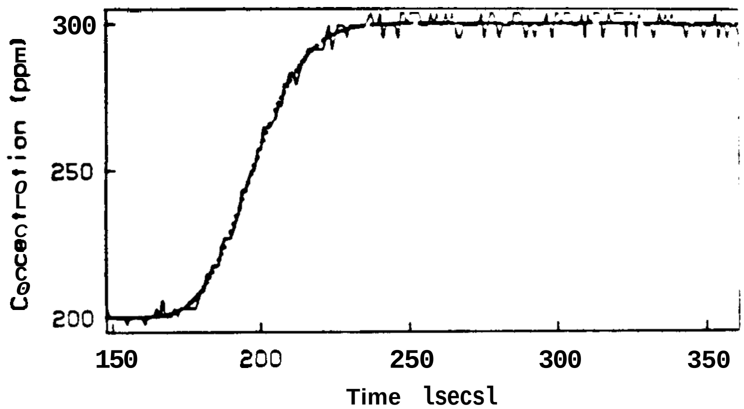

Run 4 - elecrode 21



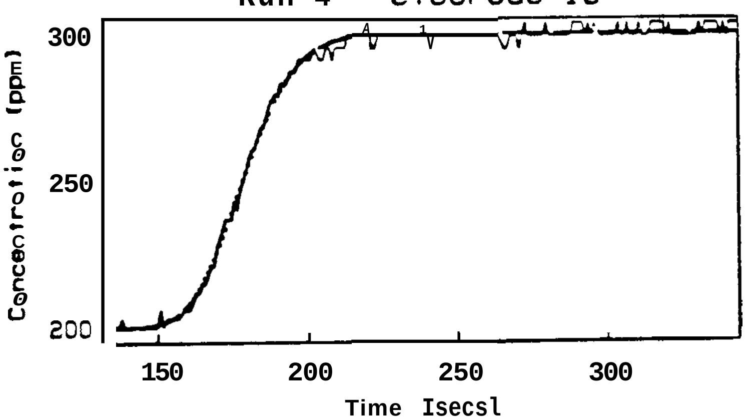

Run 4 - elecrode 18



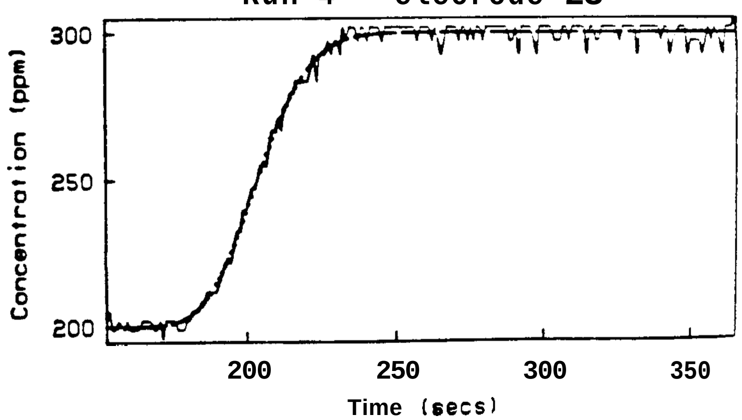

Run 4 - elecrode 22



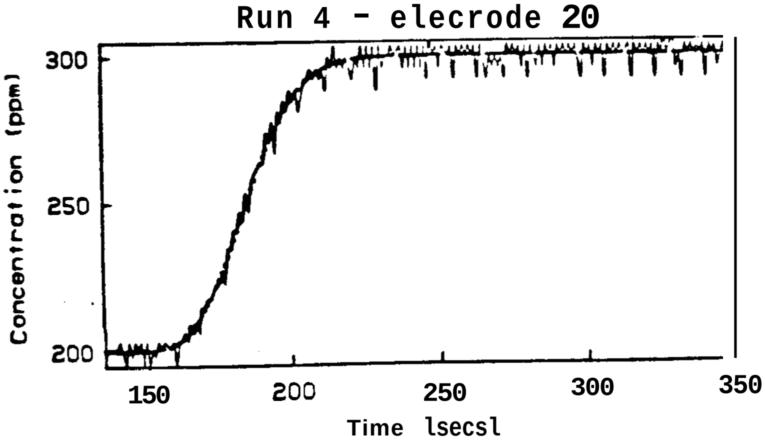

Run 4 - elecrode 19



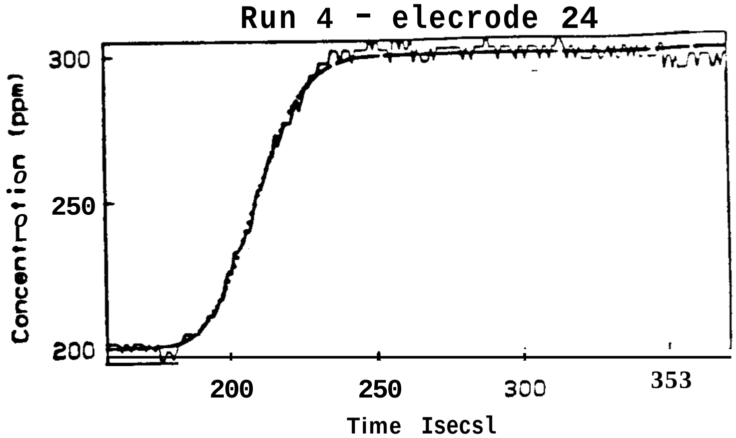

Run 4 - elecrode 23



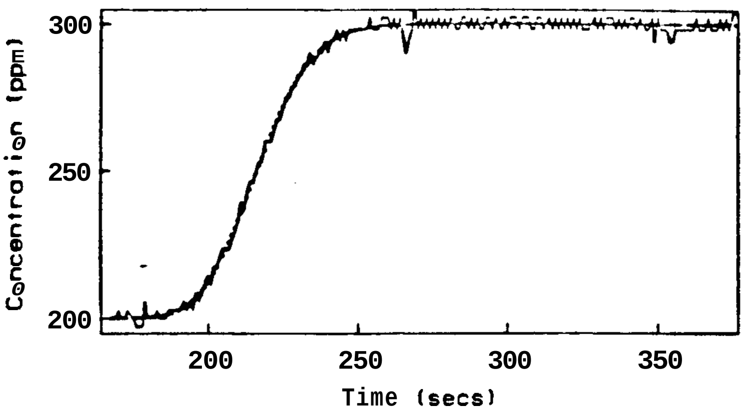

Run 4 - elecrode 20



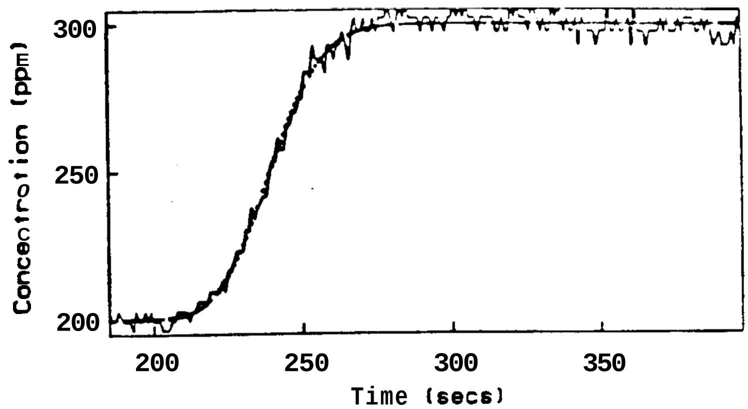

Run 4 - elecrode 24



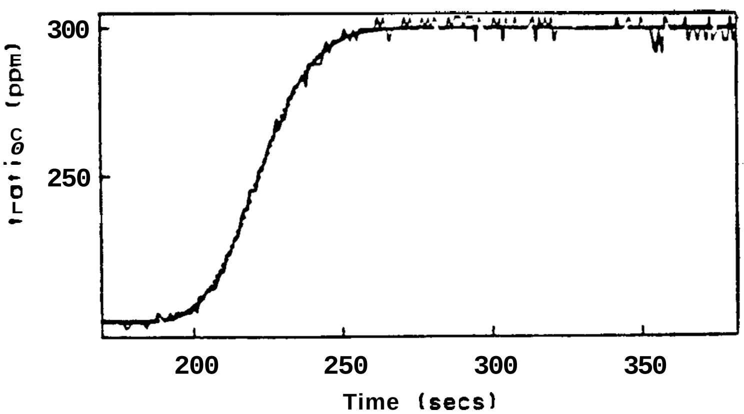
Run 4 - elecrode 25



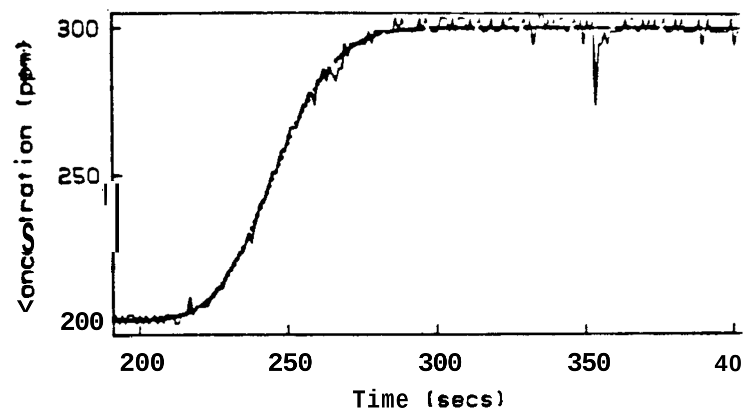

Run 4 - elecrode 29



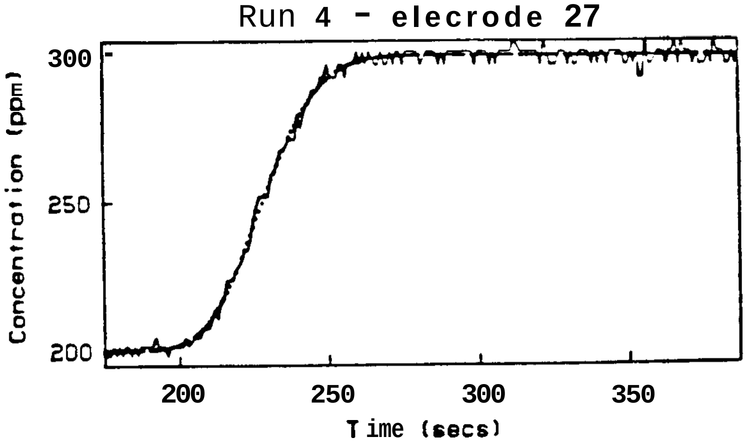

Run 4 - elecrode 26



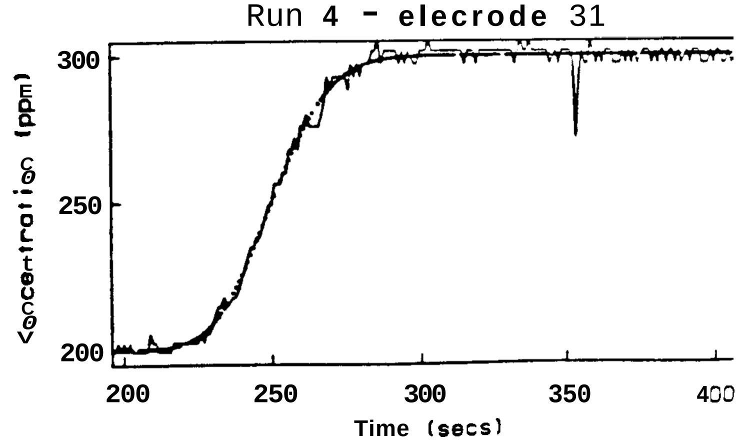

Run 4 - elecrode 30



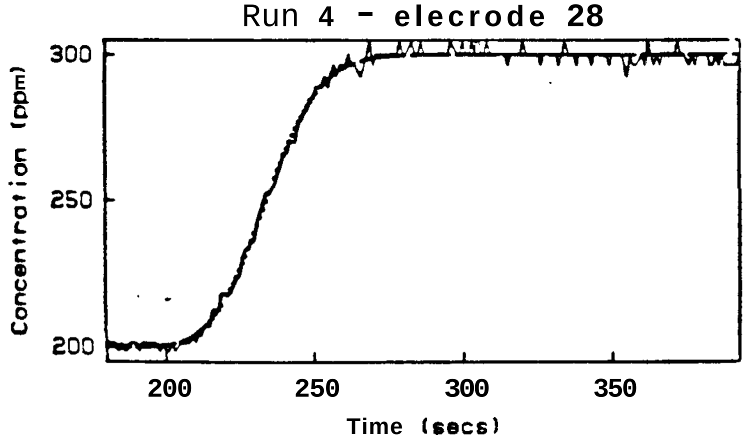

Run 4 - elecrode 27




Run 4 - elecrode 31




Run 4 - elecrode 28




Run 4 - elecrode 32

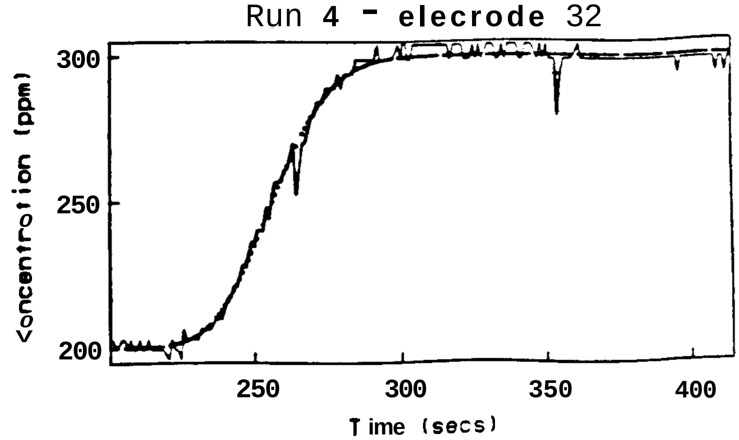


Table 6. RUN 5 Results

| Electrode | $C_o$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\eta$ ($cm^2/sec$) | apparent $t_o$ (sec) |
|---|---|---|---|---|---|
| 1 | 172.4794 | 305.4789 | 0.4567 | 0.4547 | 79.4469 |
| 2 | 172.5788 | 306.0025 | 0.4574 | 0.6216 | 79.3727 |
| 3 | 176.8712 | 305.1535 | 0.5017 | 0.2523 | 78.1311 |
| 4 | 177.0194 | 304.9398 | 0.5313 | 0.1454 | 79.7869 |
| 6 | 176.9580 | 305.1213 | 0.5216 | 0.2398 | 77.1715 |
| 6 | 176.9227 | 305.1701 | 0.5223 | 0.2561 | 78.5297 |
| 7 | 177.0075 | 305.1239 | 0.5370 | 0.2177 | 78.9163 |
| 8 | 176.9869 | 305.1637 | 0.5361 | 0.2446 | 79.3120 |
| 9 | 176.8473 | 305.2737 | 0.5305 | 0.3134 | 77.8365 |
| 10 | 176.9455 | 305.3065 | 0.5356 | 0.3051 | 78.2525 |
| 11 | 176.9516 | 305.2965 | 0.5380 | 0.3001 | 79.2787 |
| 12 | 176.9802 | 305.2385 | 0.5385 | 0.2701 | 78.9465 |
| 13 | 177.0278 | 305.1734 | 0.5407 | 0.2603 | 79.0606 |
| 14 | 176.9525 | 305.2049 | 0.5385 | 0.2628 | 79.0822 |
| 15 | 176.9865 | 305.2269 | 0.5396 | 0.2245 | 78.9550 |
| 16 | 176.9783 | 305.2457 | 0.5380 | 0.2272 | 79.0627 |
| 17 | 176.9859 | 305.2278 | 0.5433 | 0.2246 | 79.2794 |
| 18 | 176.9790 | 305.2643 | 0.5404 | 0.2555 | 78.9707 |
| 19 | 177.0272 | 305.2033 | 0.5403 | 0.2345 | 79.5479 |
| 20 | 177.0523 | 305.2374 | 0.5403 | 0.2236 | 79.1419 |
| 21 | 177.0293 | 305.2169 | 0.5406 | 0.2265 | 79.8696 |
| 22 | 177.0075 | 305.2479 | 0.5399 | 0.2226 | 79.1860 |
| 23 | 177.3219 | 305.0014 | 0.5539 | 0.1139 | 79.4677 |
| 25 | 177.2622 | 305.1656 | 0.5412 | 0.2284 | 78.9000 |
| 26 | 177.2921 | 305.1879 | 0.5403 | 0.2453 | 79.9000 |
| 27 | 177.8224 | 305.1204 | 0.5427 | 0.2068 | 79.9000 |
| 28 | 177.8538 | 305.1572 | 0.5426 | 0.2306 | 79.9000 |
| 29 | 177.5213 | 305.0650 | 0.5453 | 0.1940 | 79.9000 |
| 30 | 177.7206 | 305.1760 | 0.5471 | 0.1798 | 79.9000 |
| 31 | 177.8365 | 305.2394 | 0.5516 | 0.2109 | 79.9000 |
| 32 | 179.4189 | 305.0844 | 0.5596 | 0.2430 | 79.9000 |

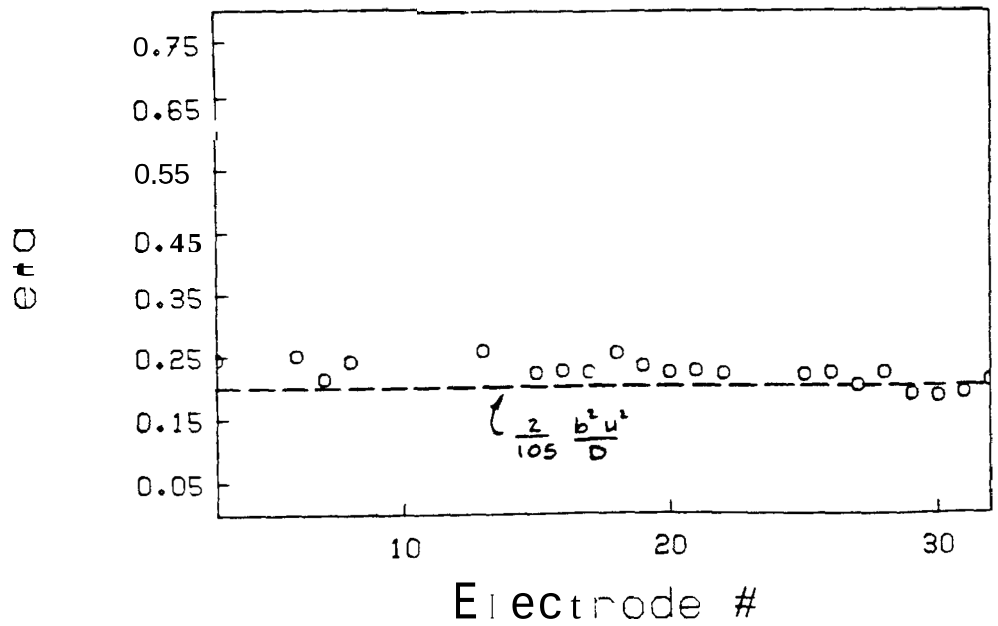
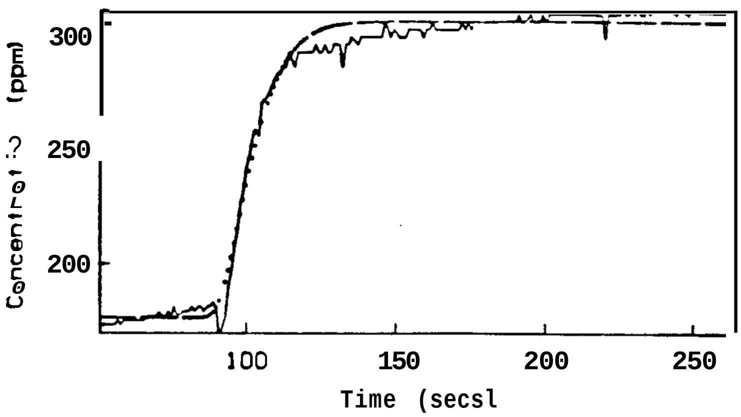
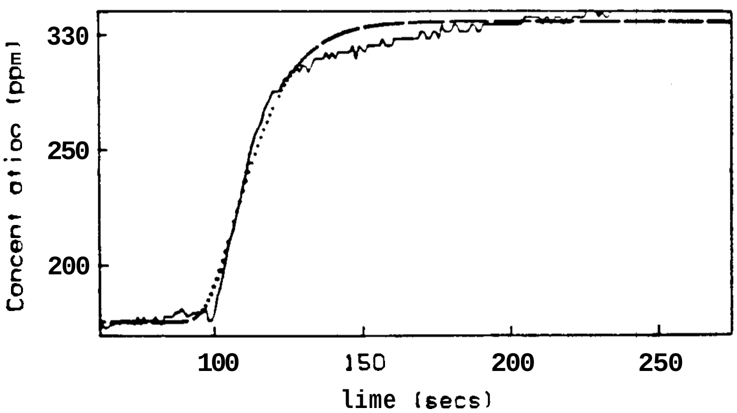



**Fig. 10. Comparison of** estimated **and calculated dispersivity. Run 5.**

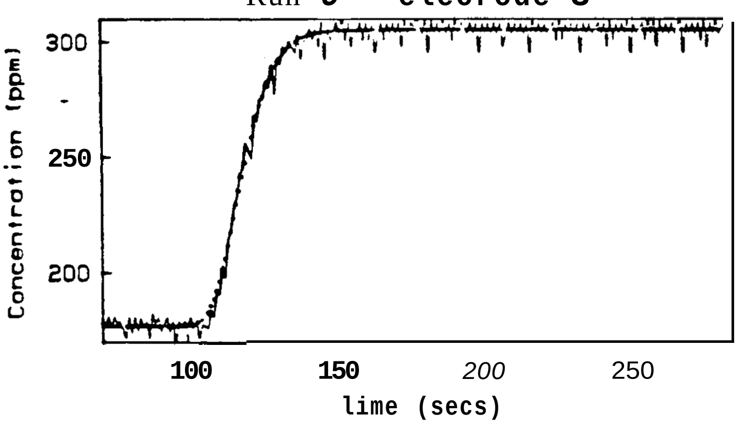
Run 5 - elecrode 1



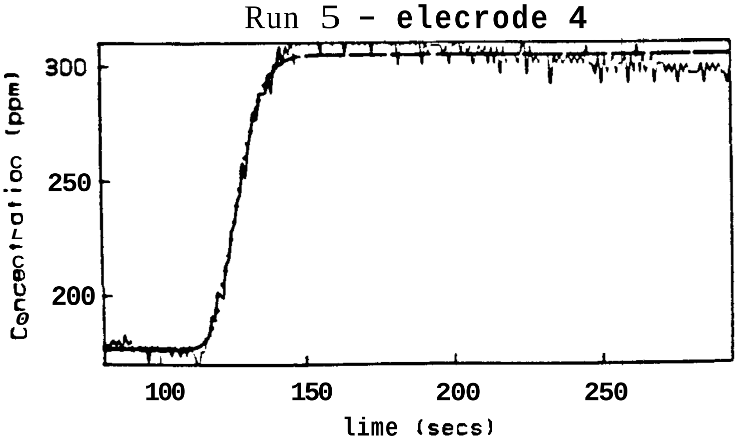

Run 5 - elecrode 2



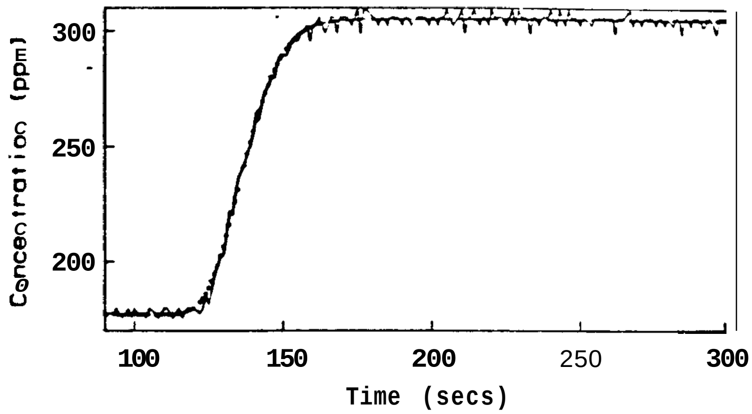

Run 5 - elecrode 3



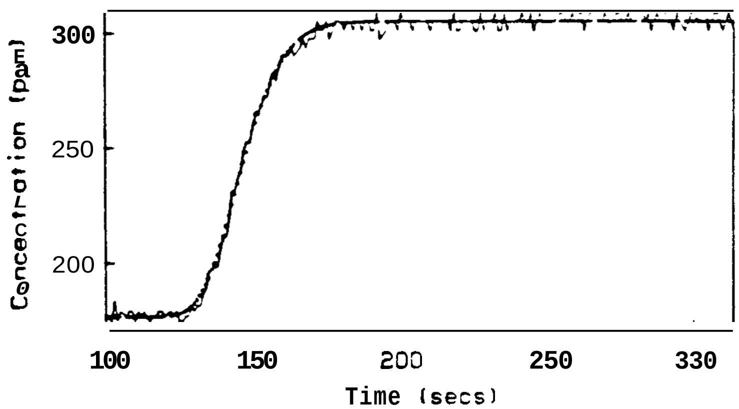

Run 5 - elecrode 4



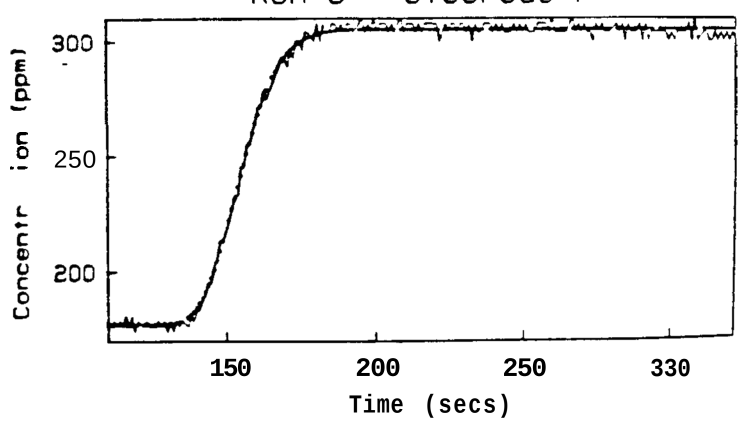

Run 5 - elecrode 5



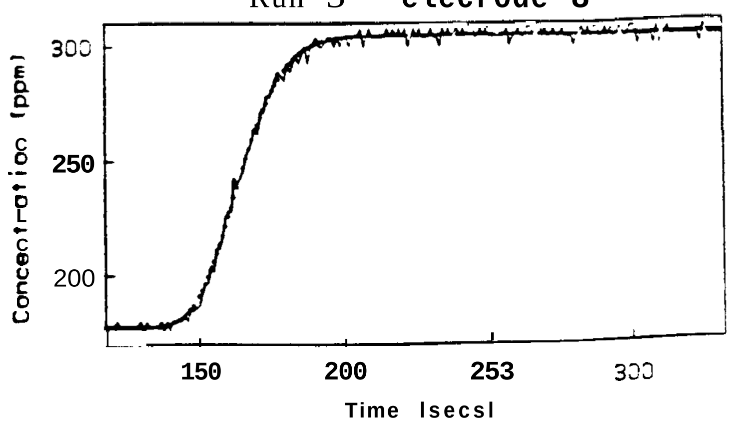

Run 5 - elecrode 6



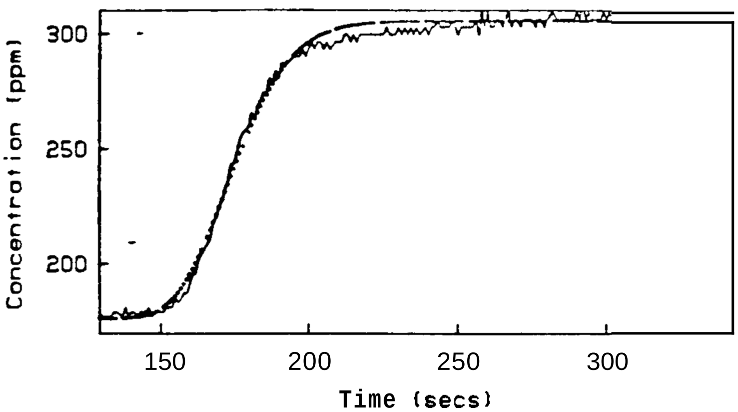

Run 5 - elecrode 7



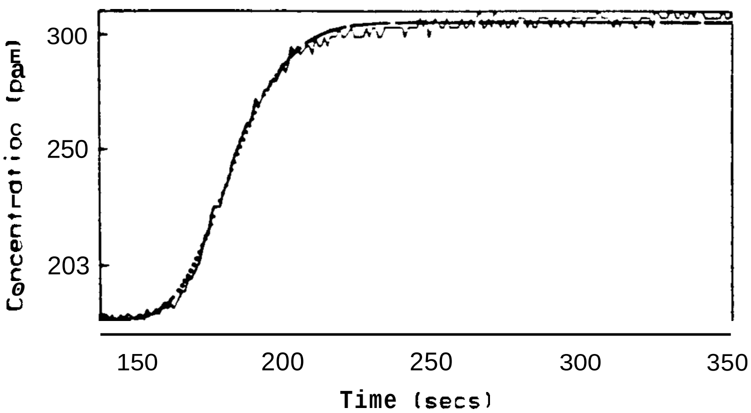

Run 5 - elecrode 8



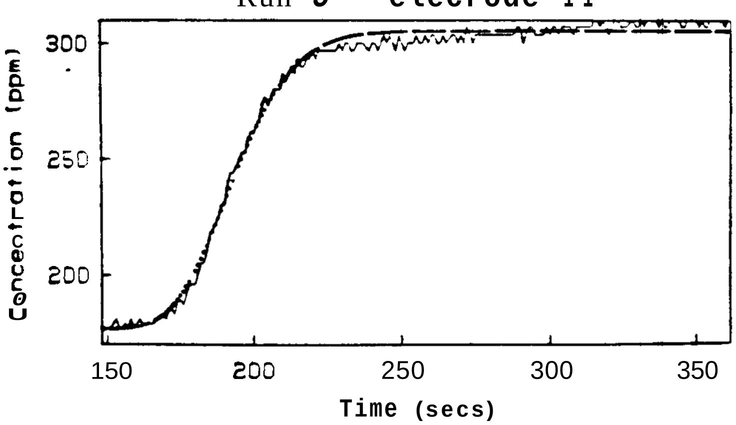
Run 5 - elecrode 9



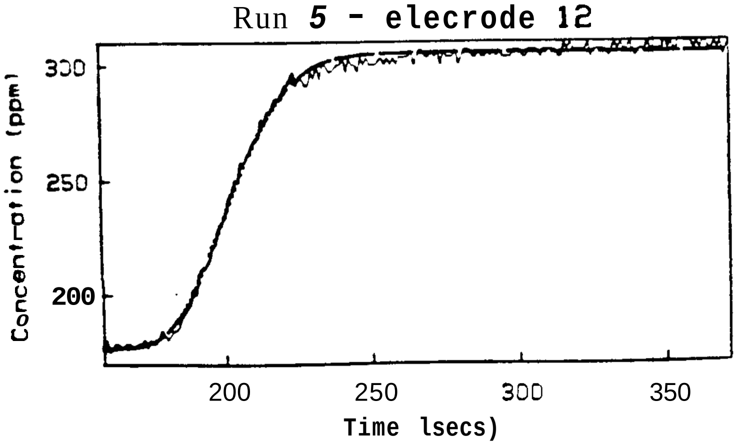

Run 5 - elecrode 10



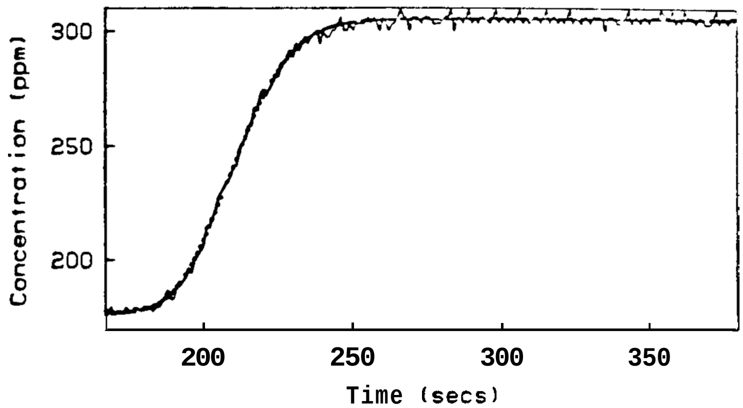

Run 5 - elecrode 11



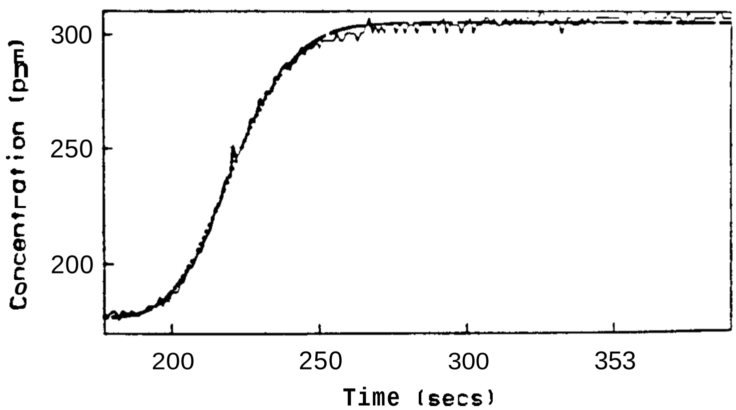

Run 5 - elecrode 12



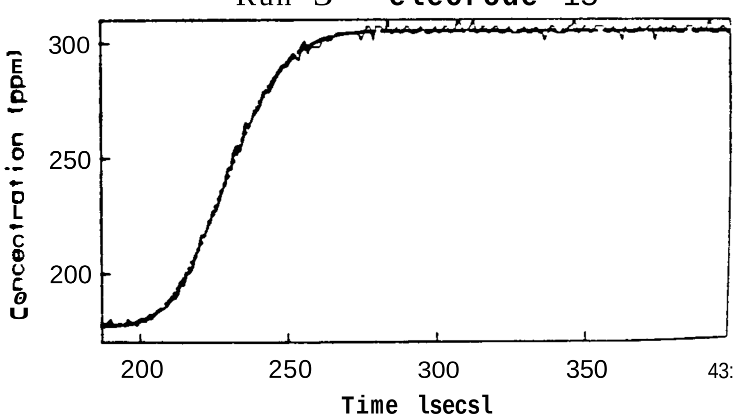

Run 5 - elecrode 13



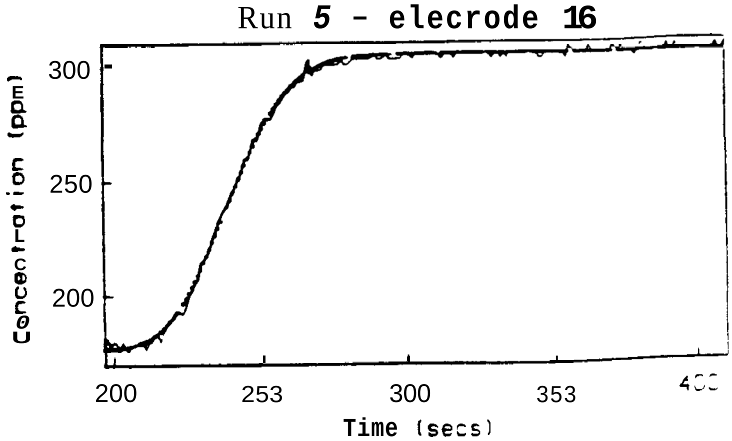

Run 5 - elecrode 14



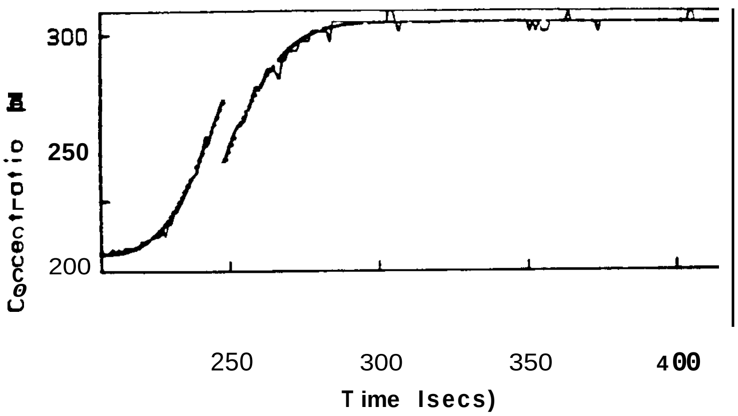

Run 5 - elecrode 15



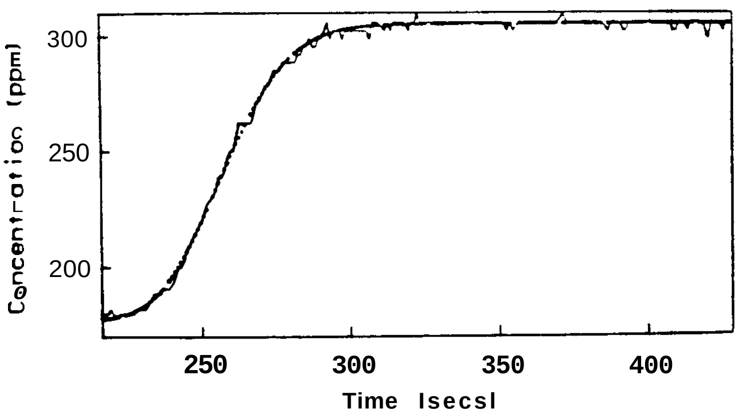

Run 5 - elecrode 16



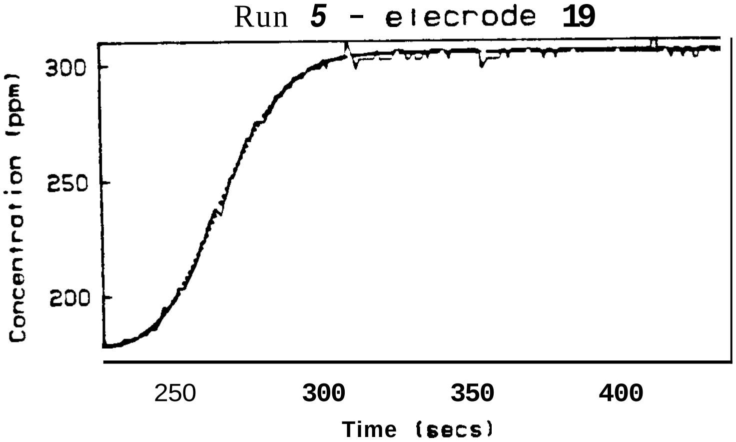
Run 5 - elecrode 17



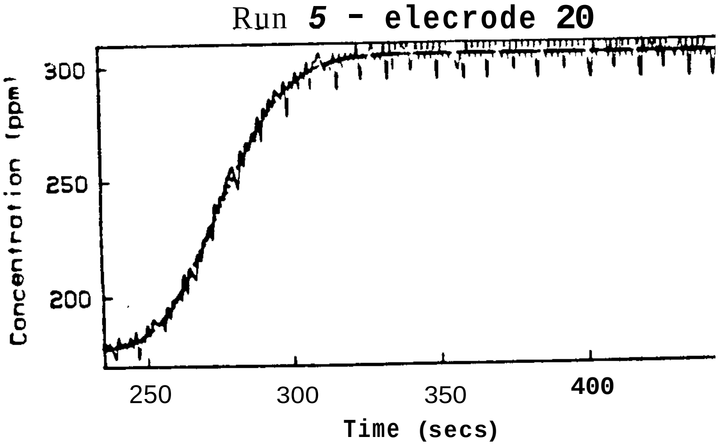

Run 5 - elecrode 18



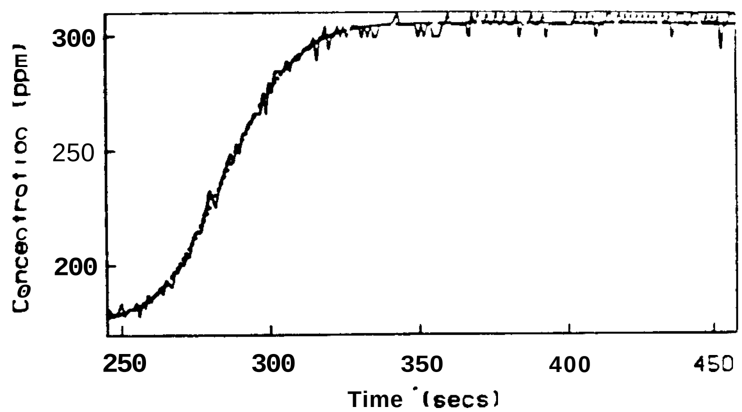

Run 5 - elecrode 19



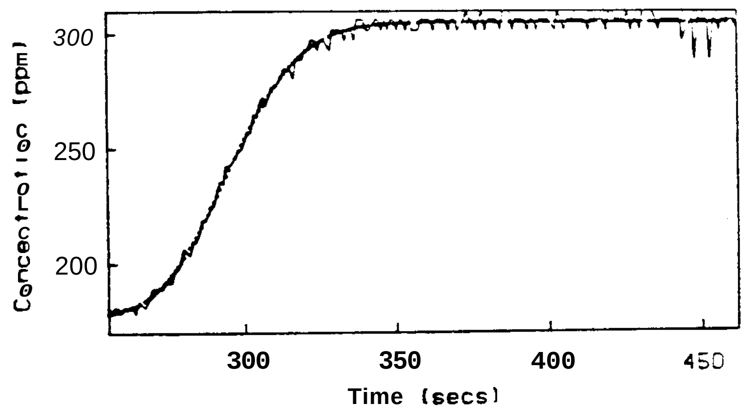

Run 5 - elecrode 20



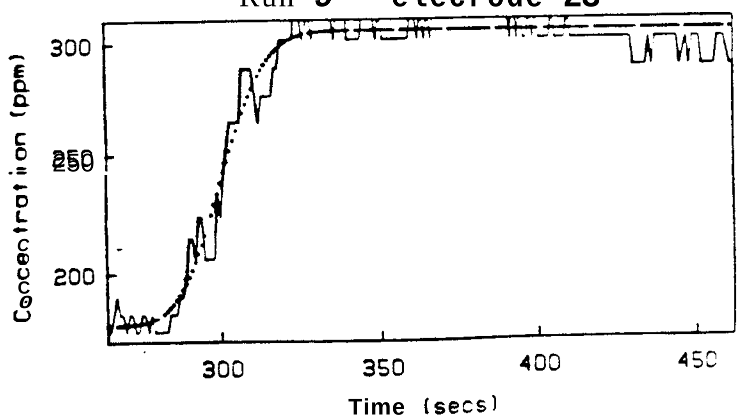

Run 5 - elecrode 21



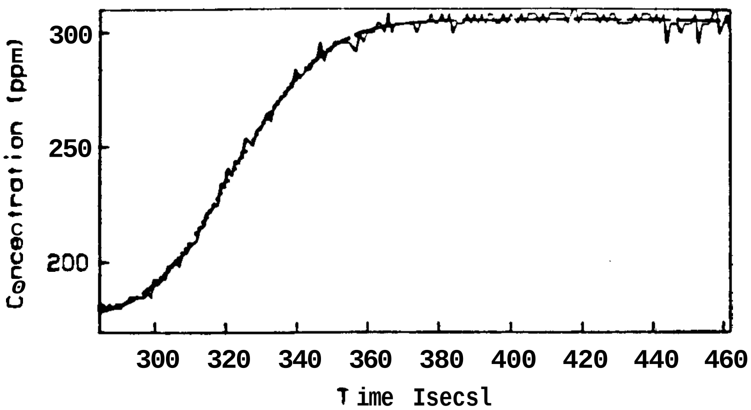

Run 5 - elecrode 22



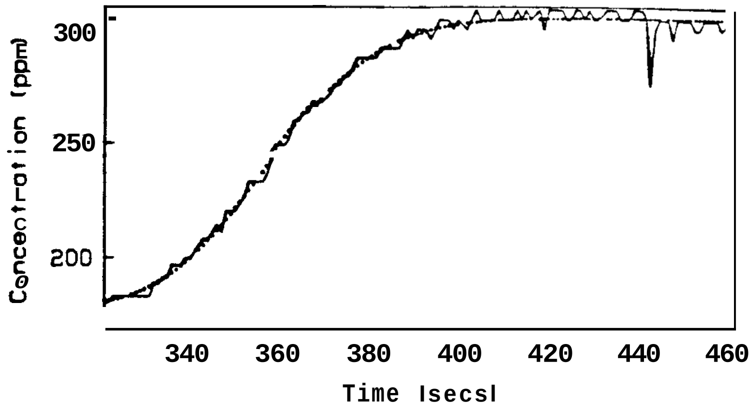

Run 5 - elecrode 23



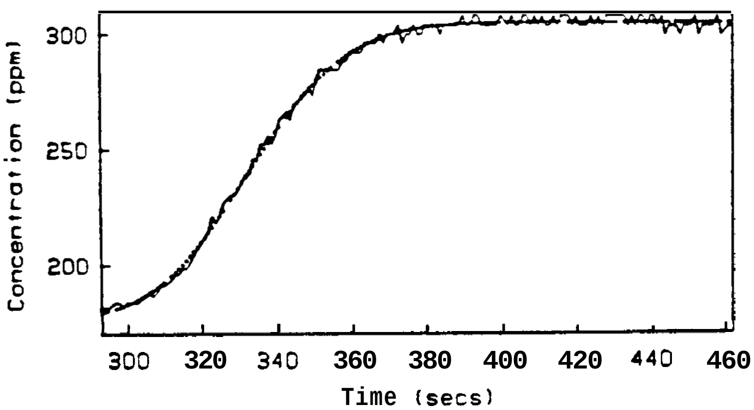
run5 - **electrode** 25



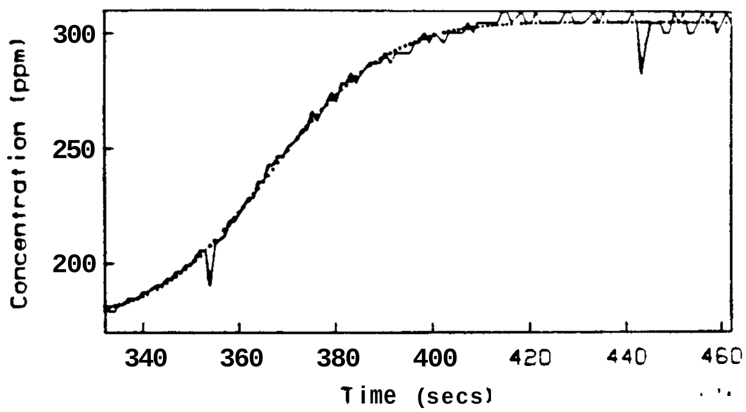

run5 - **electrode** 29



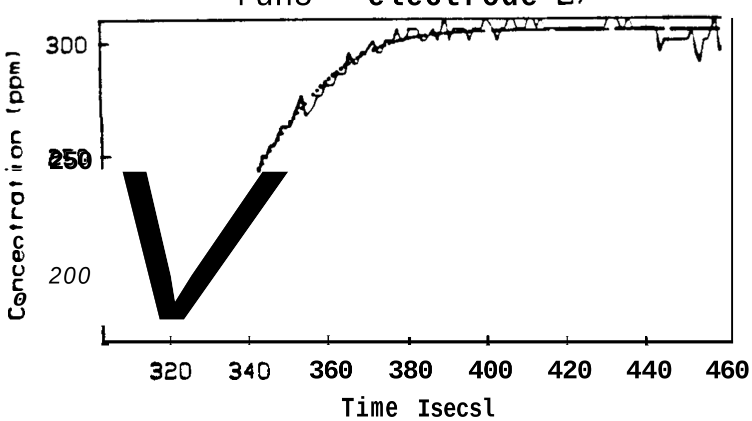

run5 - **electrode 26**



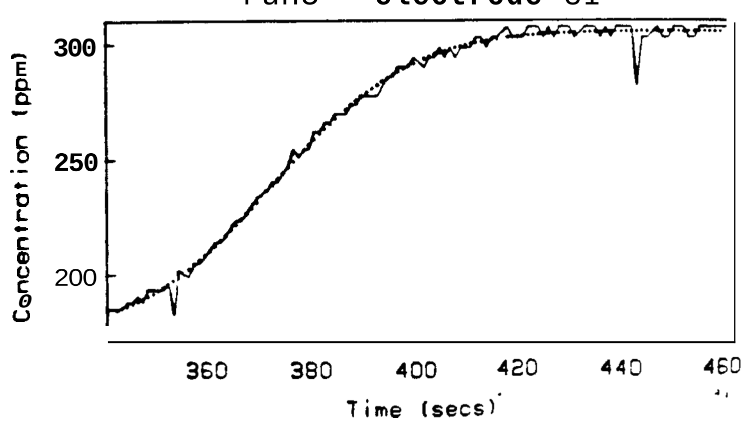

run5 - **electrode** 30



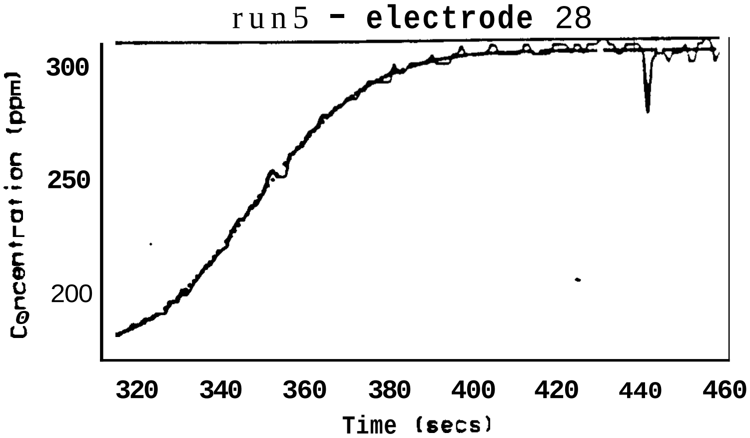

run5 - **electrode** 27



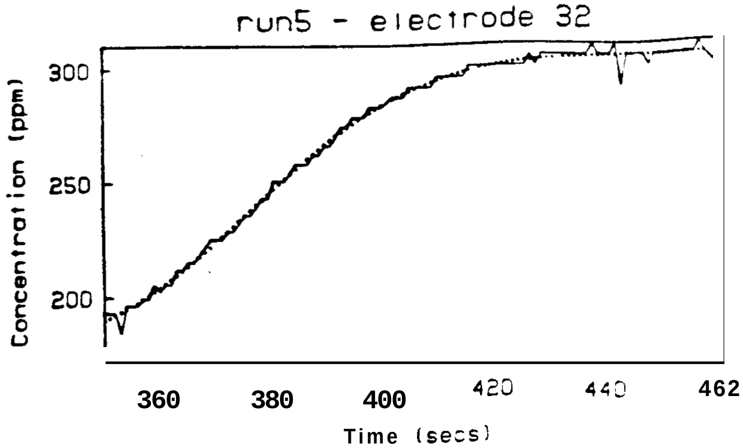

run5 - **electrode** 31




run5 - **electrode** 28




run5 - electrode 32



| Table 7. RUN 7 Results | | | | | |
|---|---|---|---|---|---|
| electrode | $C_0$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\overline{\eta}$ (cm$^2$/sec) | apparent $t_0$ (sec) |
| 1 | 179.2666 | 305.6806 | 0.5717 | 0.4679 | 71.7837 |
| 2 | 178.6063 | 305.4210 | 0.6423 | 0.5171 | 72.7662 |
| 3 | 176.7776 | 305.0984 | 0.7263 | 0.3560 | 72.4584 |
| 4 | 176.9432 | 305.0747 | 0.7554 | 0.4441 | 71.7963 |
| 6 | 177.2003 | 305.0239 | 0.7962 | 0.3357 | 72.1182 |
| 6 | 176.8742 | 305.0944 | 0.8025 | 0.5089 | 72.0861 |
| 7 | 176.5317 | 304.7535 | 0.8446 | 0.3708 | 73.2562 |
| 8 | 177.0942 | 305.0427 | 0.8334 | 0.4695 | 72.5221 |
| 9 | 176.9618 | 305.1009 | 0.8464 | 0.5463 | 72.3593 |
| 10 | 178.5818 | 308.8341 | 0.7717 | 2.1440 | 69.0034 |
| 11 | 177.0820 | 305.0543 | 0.8744 | 0.4760 | 72.6505 |
| 12 | 176.9861 | 305.1233 | 0.8728 | 0.4913 | 72.6992 |
| 13 | 176.9890 | 305.1339 | 0.8793 | 0.4733 | 72.4796 |
| 14 | 176.9812 | 305.1491 | 0.8763 | 0.5617 | 72.4105 |
| 15 | 177.0046 | 305.1351 | 0.8821 | 0.4752 | 73.2172 |
| 16 | 176.9860 | 305.1614 | 0.8812 | 0.5194 | 73.3541 |
| 17 | 177.0174 | 305.1481 | 0.8890 | 0.4531 | 72.5010 |
| 18 | 177.0549 | 305.1027 | 0.8926 | 0.5390 | 73.2215 |
| 19 | 177.0296 | 305.1550 | 0.8914 | 0.5489 | 73.3826 |
| 20 | 177.0435 | 305.2248 | 0.8864 | 0.4589 | 72.9996 |
| 21 | 177.0824 | 305.1236 | 0.8990 | 0.4089 | 73.0323 |
| 22 | 176.9820 | 305.3146 | 0.8868 | 0.4852 | 72.6011 |
| 23 | 177.0107 | 305.2252 | 0.8908 | 0.4432 | 72.4894 |
| 24 | 177.0221 | 305.2593 | 0.8900 | 0.3673 | 72.6674 |
| 25 | 177.0153 | 305.1995 | 0.8953 | 0.4526 | 72.8346 |
| 26 | 177.0257 | 305.2189 | 0.8937 | 0.4460 | 72.8886 |
| 27 | 177.0623 | 305.1226 | 0.9012 | 0.4210 | 72.5135 |
| 28 | 177.0006 | 305.2429 | 0.9031 | 0.4885 | 73.2226 |
| 29 | 177.0204 | 305.2684 | 0.9005 | 0.4380 | 73.1053 |
| 30 | 176.9823 | 305.3311 | 0.9062 | 0.4202 | 72.5762 |
| 31 | 177.0160 | 305.2626 | 0.9069 | 0.4229 | 73.0941 |
| 32 | 177.0289 | 305.2506 | 0.9049 | 0.4531 | 73.0907 |

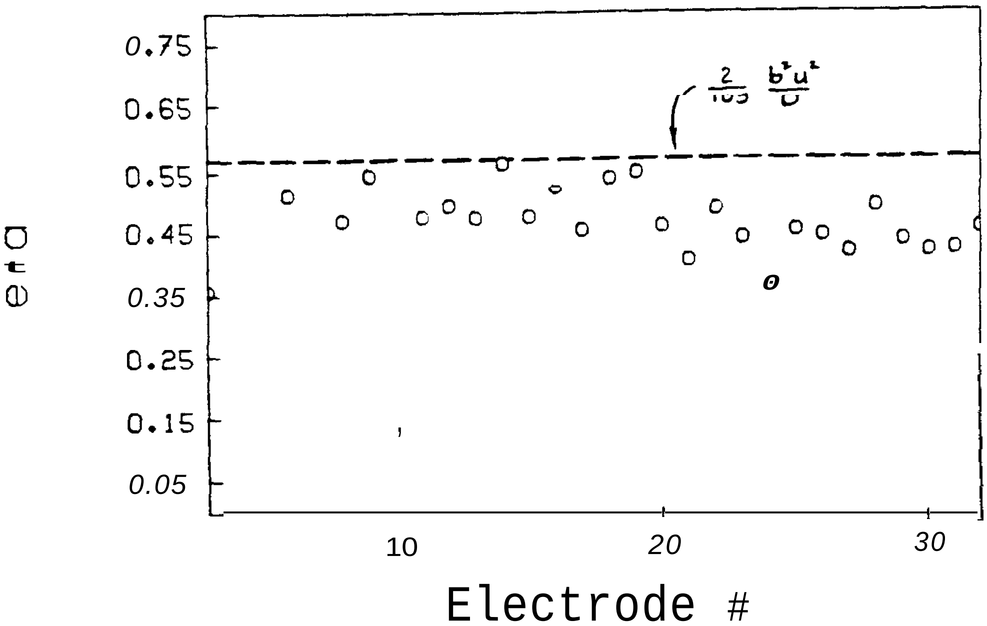
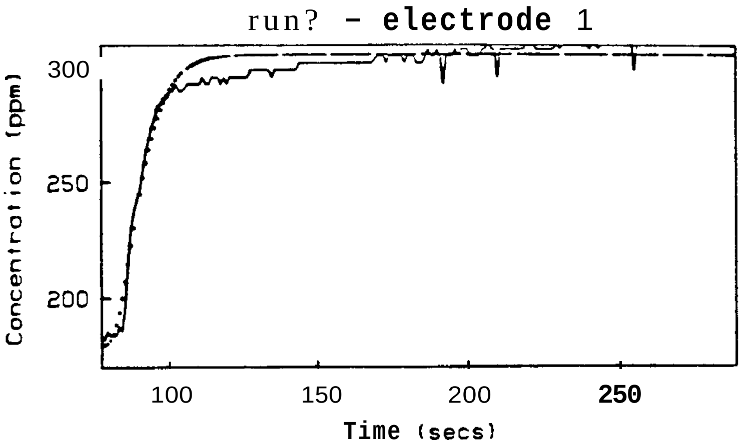
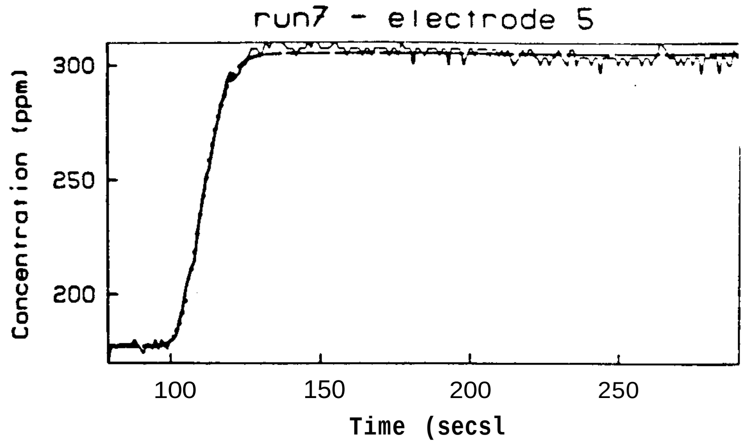
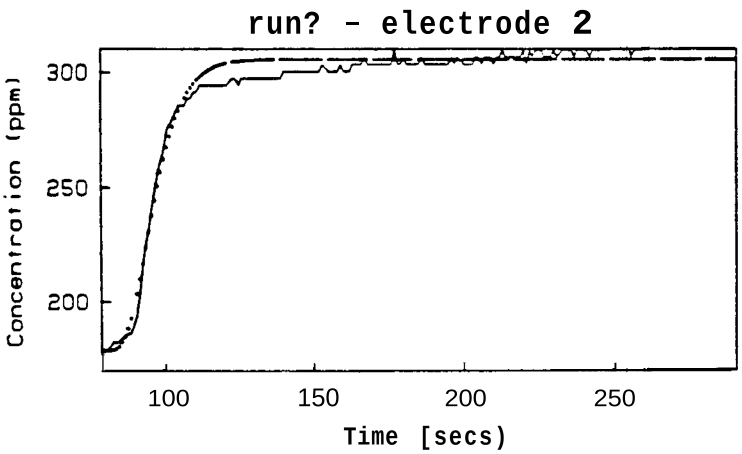
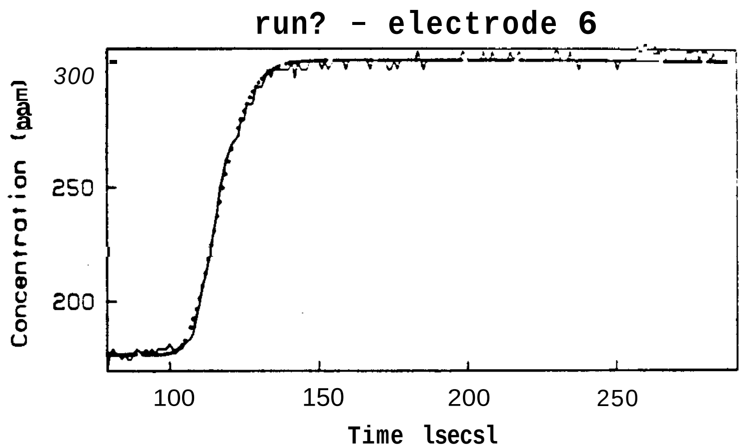


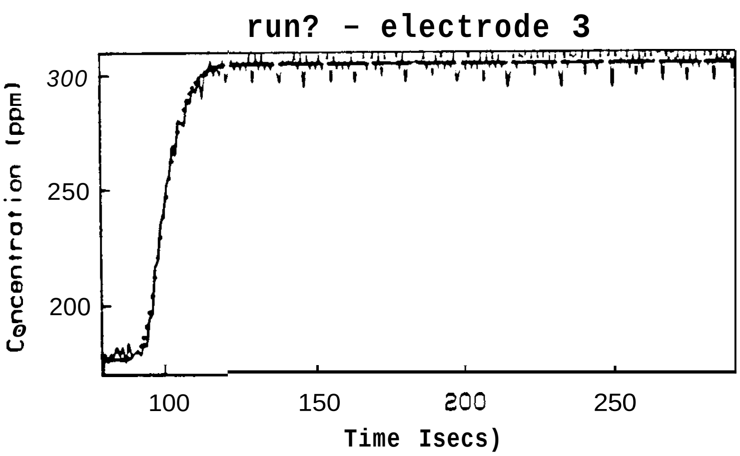

**Fig. 11.** Comparison **of estimated and calculated dispersivity.** Run **7.**

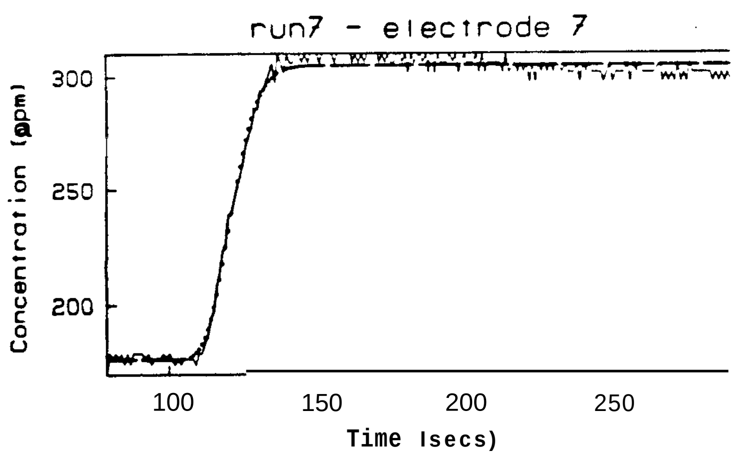

run? - electrode 1
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

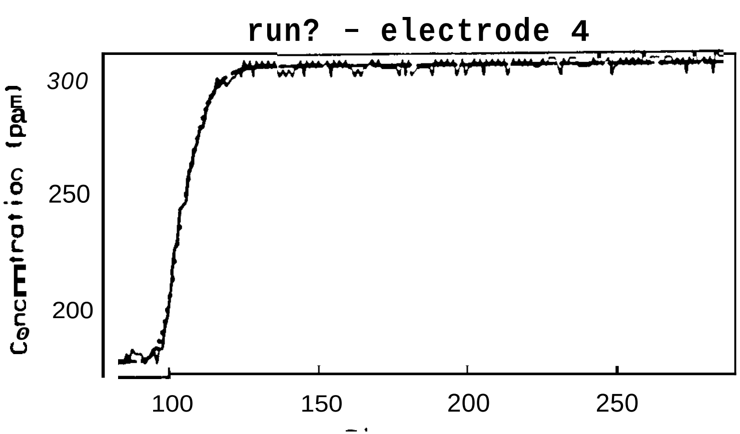

run7 - electrode 5
Concentration (ppm)
300
250
200
100
150
200
250
Time (secsl

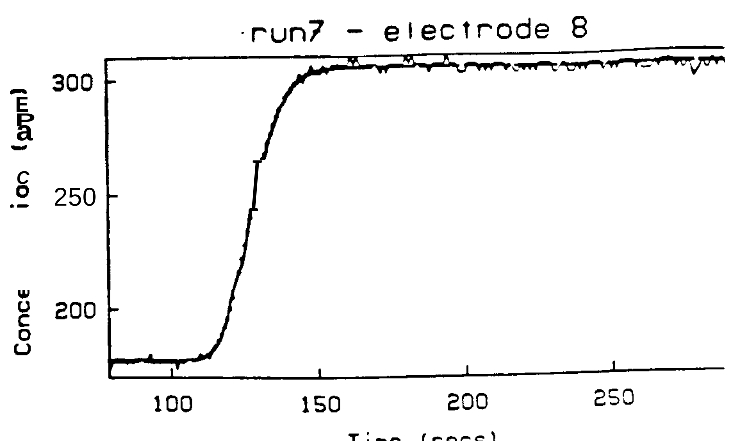

run? - electrode 2
Concentration (ppm)
300
250
200
100
150
200
250
Time [secs)

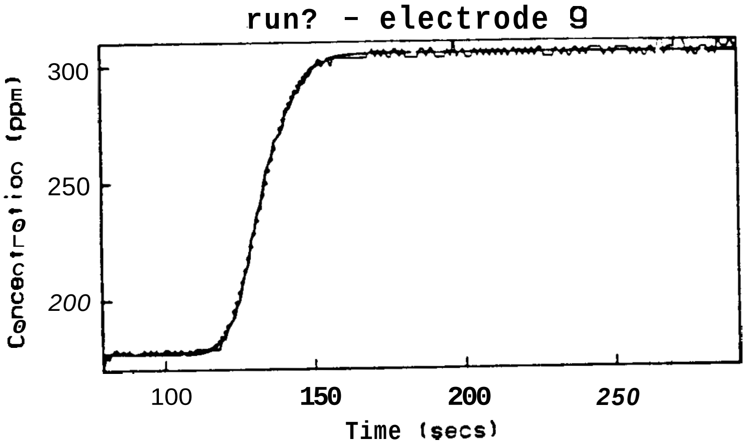

run? - electrode 6
Concentration (ppm)
300
250
200
100
150
200
250
Time lsecsl

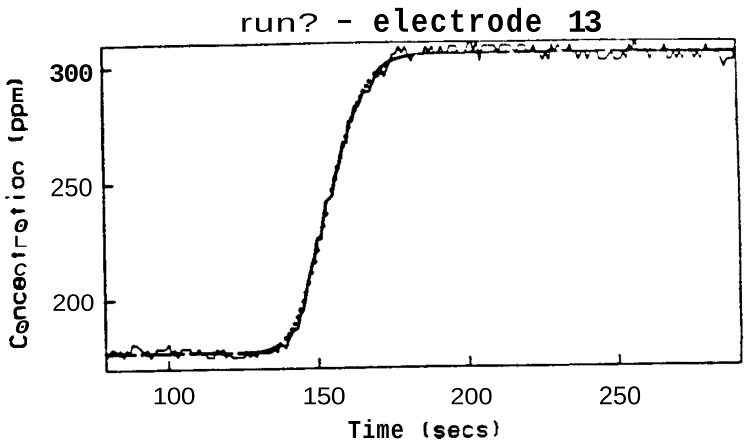

run? - electrode 3
Concentration (ppm)
300
250
200
100
150
200
250
Time Isecs)

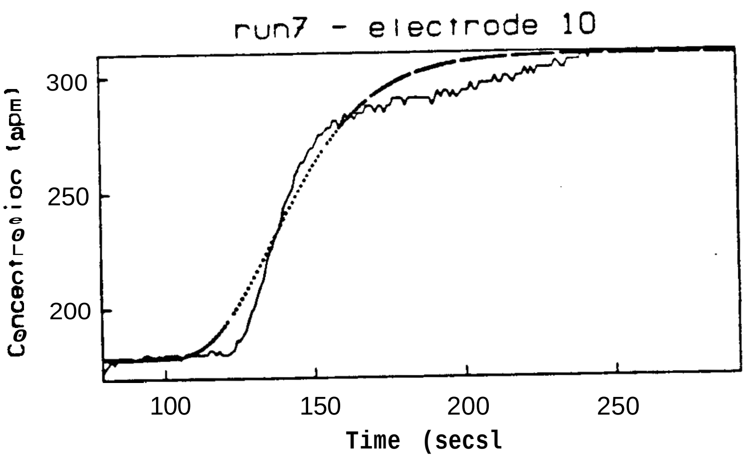

run7 - electrode 7
Concentration (ppm)
300
250
200
100
150
200
250
Time Isecs)

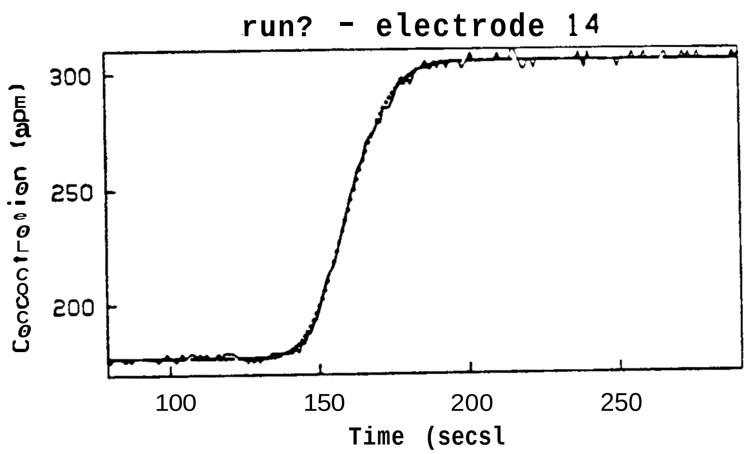

run? - electrode 4
300
250
200
100
150
200
250

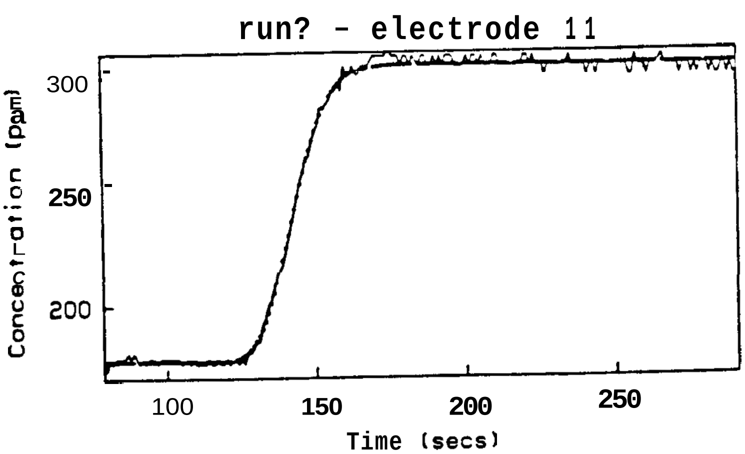

run7 - electrode 8
300
250
200
100
150
200
250

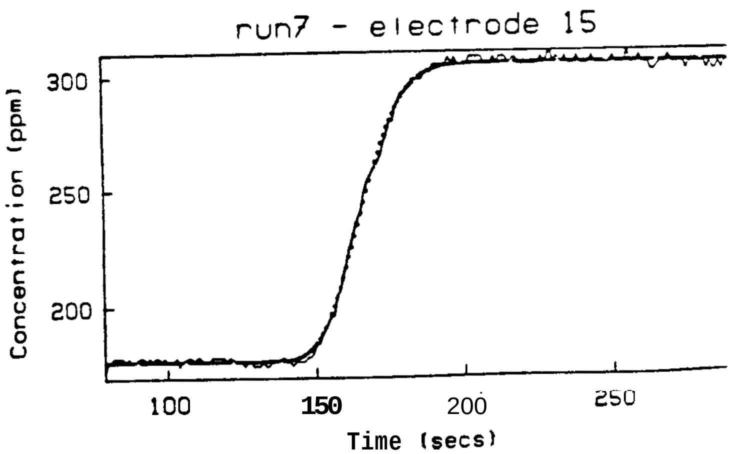

run? - electrode 9
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

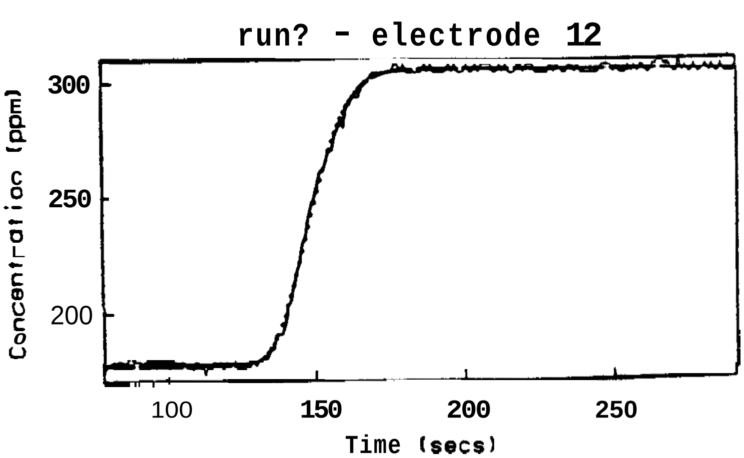

run? - electrode 13
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

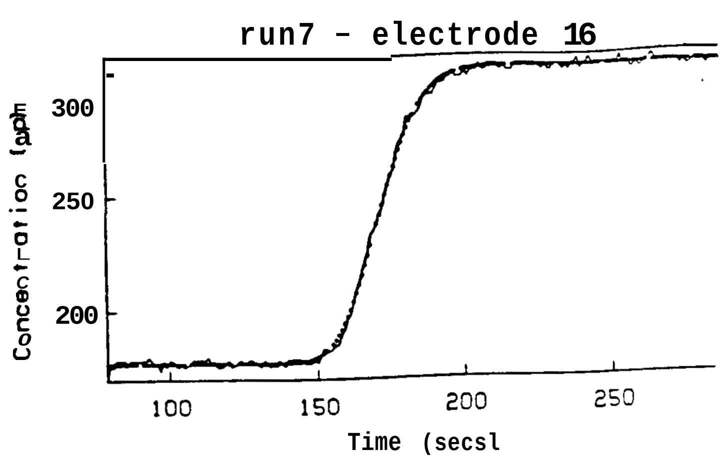

run7 - electrode 10
Concentration (ppm)
300
250
200
100
150
200
250
Time (secsl


run? - electrode 14
Concentration (ppm)
300
250
200
100
150
200
250
Time (secsl

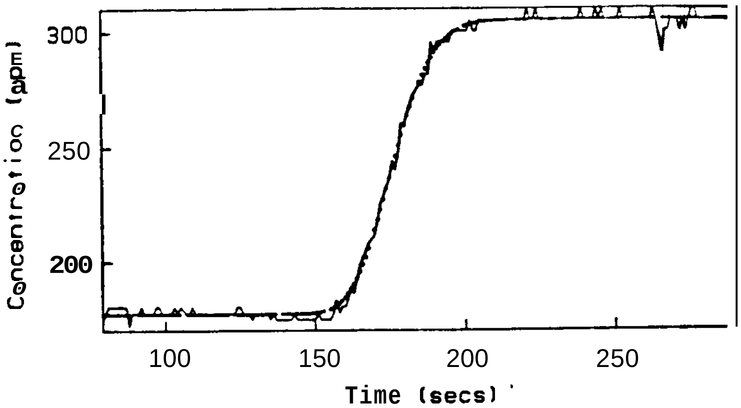

run? - electrode 11
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)


run7 - electrode 15
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)

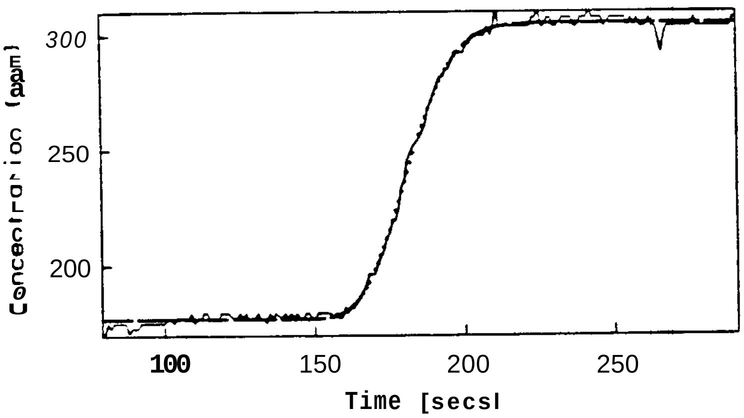

run? - electrode 12
Concentration (ppm)
300
250
200
100
150
200
250
Time (secs)


run7 - electrode 16
Concentration (ppm)
300
250
200
100
150
200
250
Time (secsl

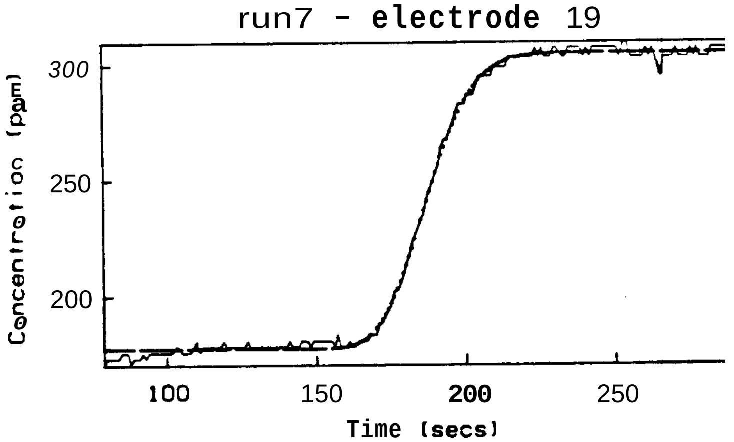
run7 - **electrode** 17



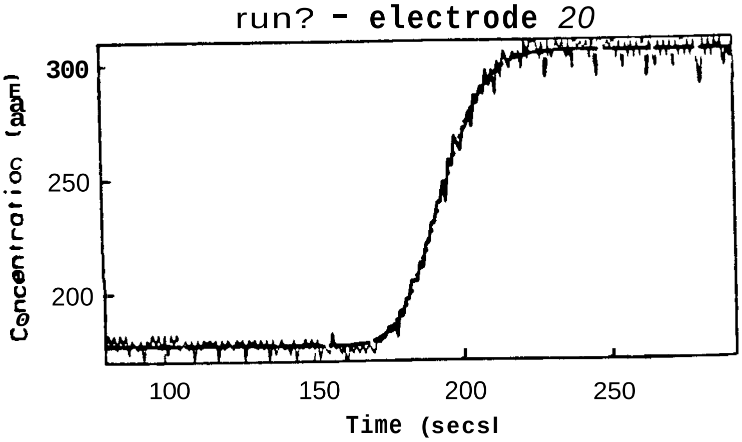

run7 - **electrode** 18



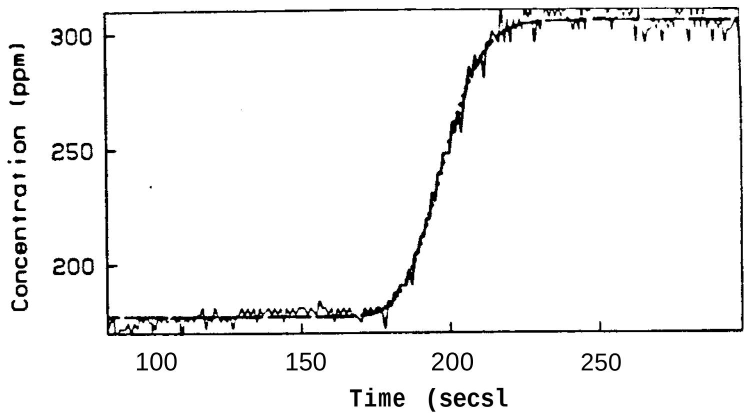

run7 - **electrode** 19



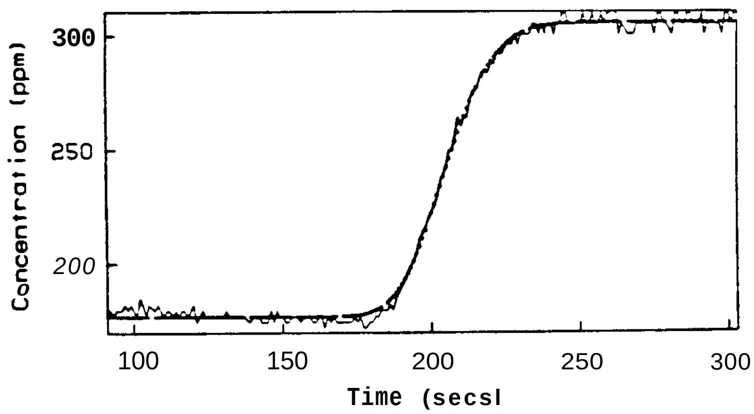

run? - **electrode** *20*



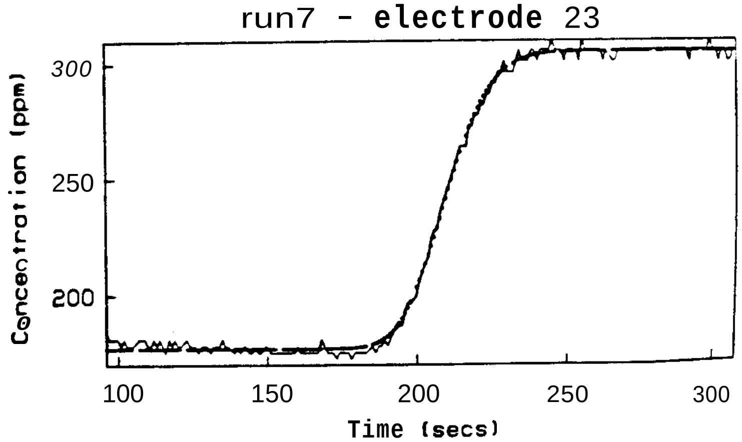

run? - **electrode** 21



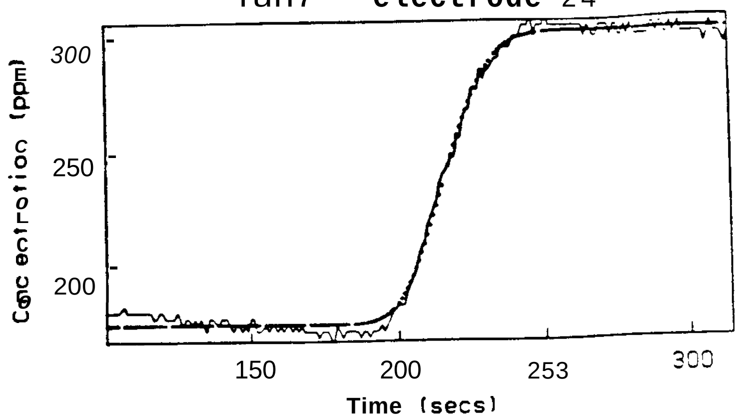

run7 - **electrode** 22



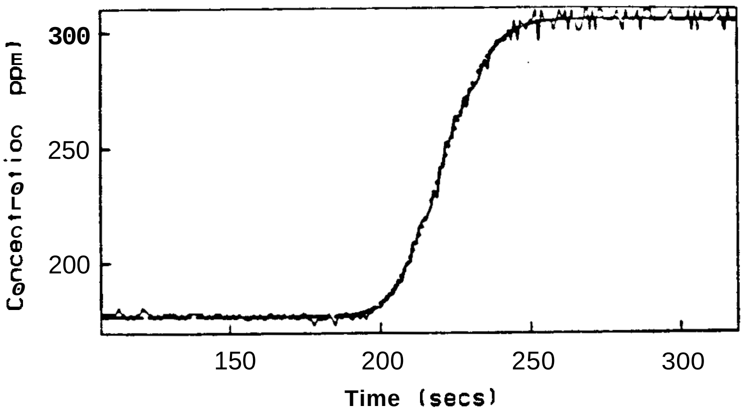

run7 - **electrode** 23



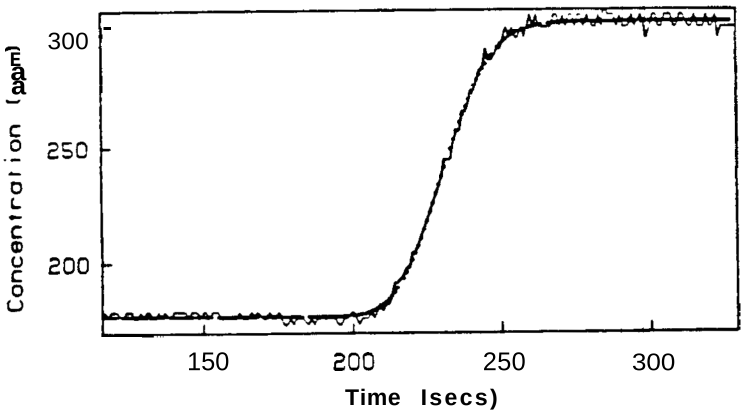

run7 - **electrode** 24



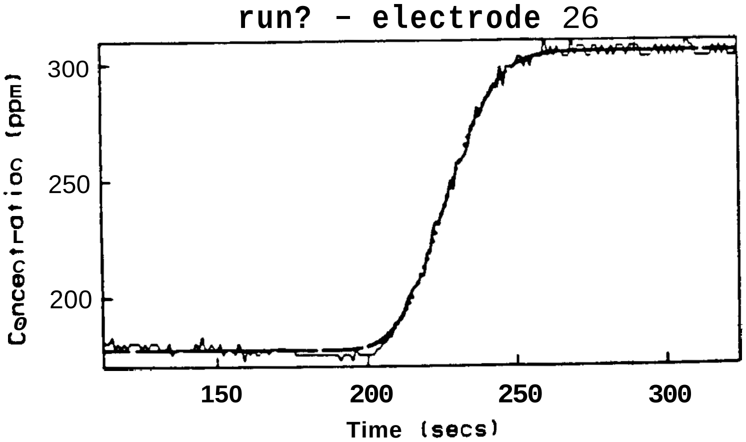
run7 - electrode 25



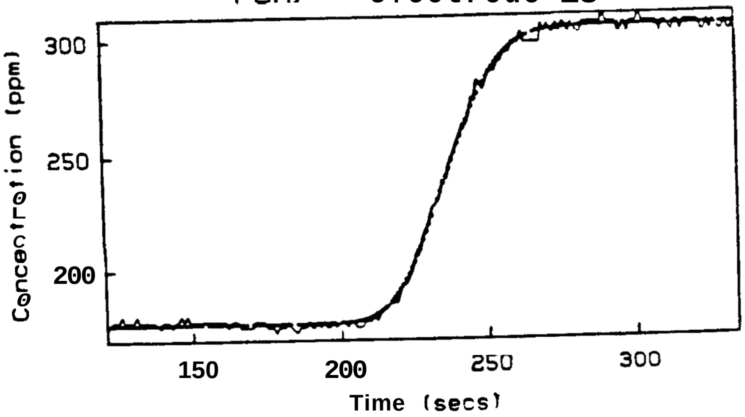

run? - electrode 27



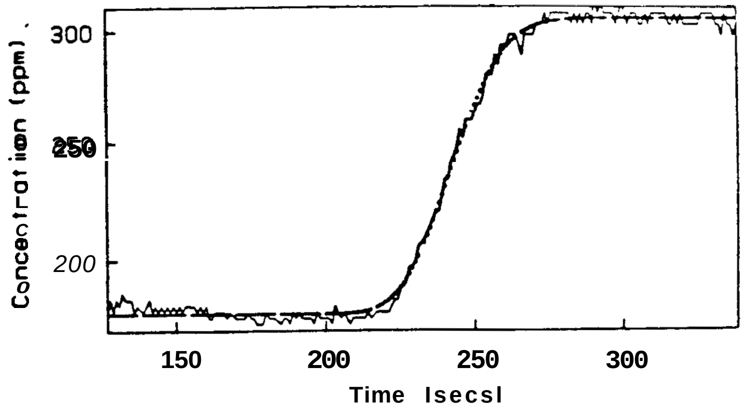

run? - electrode 26



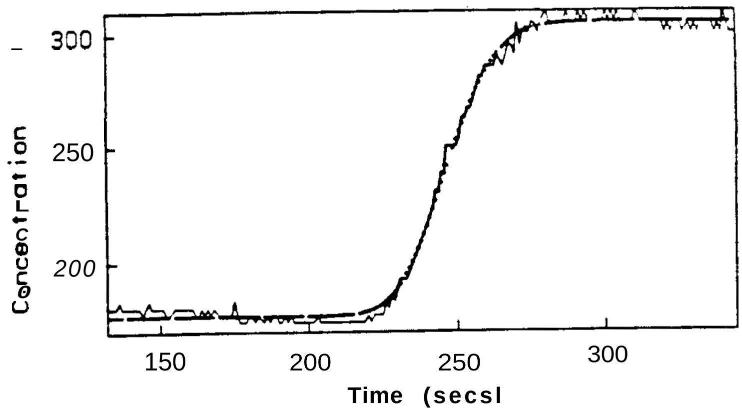

run7 - electrode 28



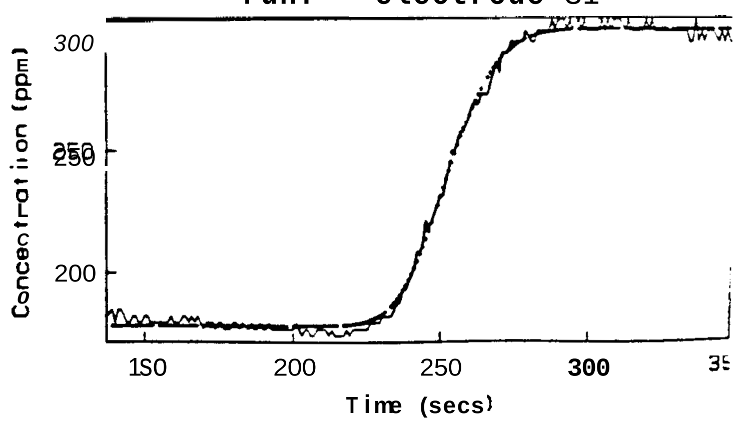

run? - electrode 29



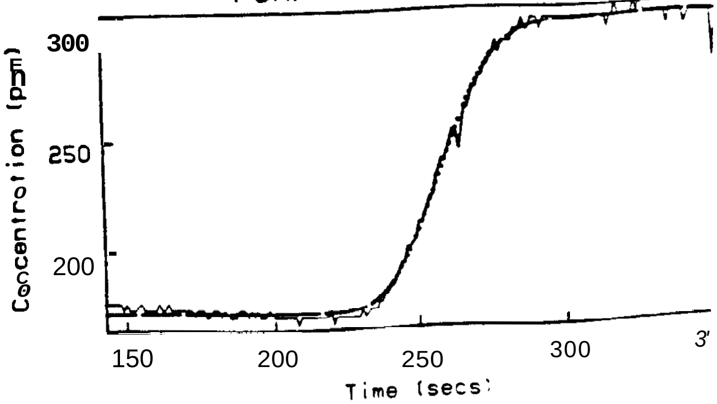

run? - electrode 30




run? - electrode 31




run7 - electrode 32



| Table 8. RUN 8 Results | | | | | |
|---|---|---|---|---|---|
| Electrode | $C_0$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\eta$ ($cm^2/sec$) | apparent $t_0$ (sec) |
| 2 | 170.1399 | 304.8965 | 0.5513 | 0.9233 | 59.9313 |
| 3 | 179.8520 | 306.0940 | 0.4889 | 0.4890 | 57.0895 |
| 4 | 176.8666 | 305.3446 | 0.4808 | 0.2899 | 57.8292 |
| 6 | 176.7606 | 305.5164 | 0.4695 | 0.2779 | 58.7547 |
| 6 | 176.9526 | 306.1938 | 0.4655 | 0.2536 | 57.2856 |
| 7 | 176.9783 | 305.4040 | 0.4734 | 0.1585 | 59.8243 |
| 8 | 176.8539 | 305.5481 | 0.4669 | 0.2108 | 59.7648 |
| 9 | 176.8430 | 305.6109 | 0.4658 | 0.2067 | 58.6273 |
| 10 | 176.8034 | 305.7105 | 0.4613 | 0.2355 | 59.5570 |
| 11 | 176.8416 | 305.5755 | 0.4622 | 0.2043 | 59.7669 |
| 12 | 176.9483 | 305.3648 | 0.4629 | 0.1351 | 59.0176 |
| 13 | 176.9379 | 305.4082 | 0.4610 | 0.1314 | 59.4217 |
| 14 | 176.8700 | 305.4972 | 0.4588 | 0.1966 | 59.5951 |
| 15 | 176.9528 | 305.3391 | 0.4595 | 0.1341 | 59.7998 |
| 16 | 176.9221 | 305.3843 | 0.4569 | 0.1555 | 59.6897 |
| 17 | 177.1410 | 305.1577 | 0.4545 | 0.2325 | 59.5215 |
| 18 | 176.9500 | 305.3545 | 0.4557 | 0.1753 | 59.9117 |
| 19 | 176.9141 | 305.4277 | 0.4545 | 0.1594 | 59.1586 |
| 20 | 176.9196 | 305.5346 | 0.4527 | 0.1775 | 69.0275 |
| 21 | 177.0144 | 305.3112 | 0.4528 | 0.1384 | 59.8606 |
| 22 | 176.8555 | 305.6125 | 0.4489 | 0.1951 | 59.4112 |
| 23 | 176.9429 | 305.4110 | 0.4499 | 0.1390 | 69.7394 |
| 24 | 176.9425 | 305.4284 | 0.4502 | 0.1712 | 69.6374 |
| 25 | 176.9841 | 305.4207 | 0.4494 | 0.1376 | 59.7931 |
| 26 | 176.9982 | 305.4667 | 0.4490 | 0.1397 | 69.2642 |
| 27 | 176.9812 | 305.6110 | 0.4498 | 0.1285 | 59.4932 |
| 28 | 177.0255 | 305.6131 | 0.4507 | 0.1225 | 59.8619 |
| 29 | 176.9814 | 306.3142 | 0.4496 | 0.1238 | 59.7666 |
| 30 | 177.0031 | 306.6916 | 0.4511 | 0.1401 | 59.5834 |
| 31 | 176.9886 | 308.1960 | 0.4515 | 0.1342 | 59.6492 |
| 32 | 176.9780 | 310.5608 | 0.4541 | 0.1365 | 69.5192 |

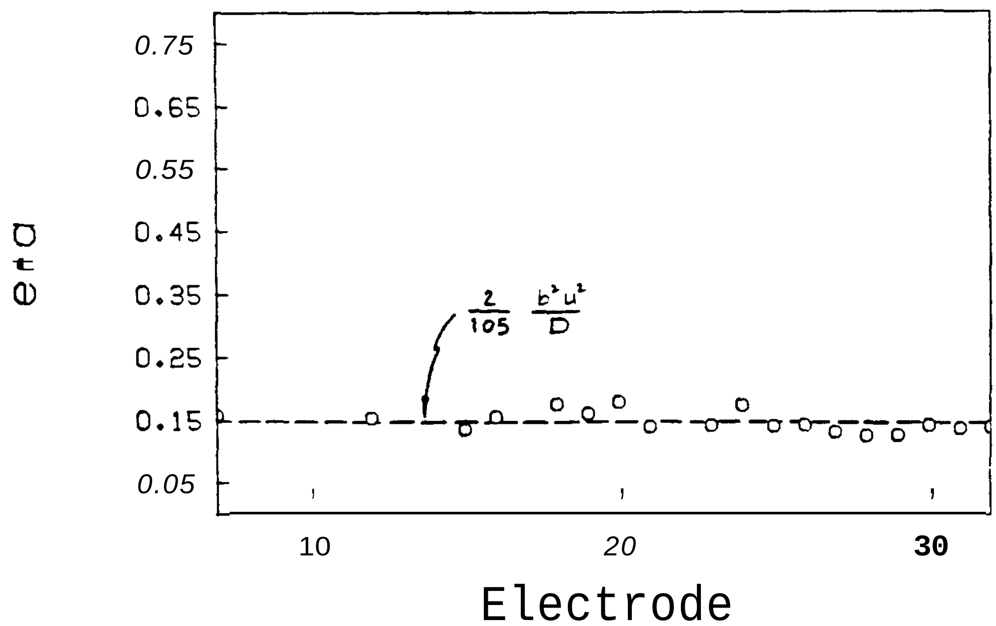
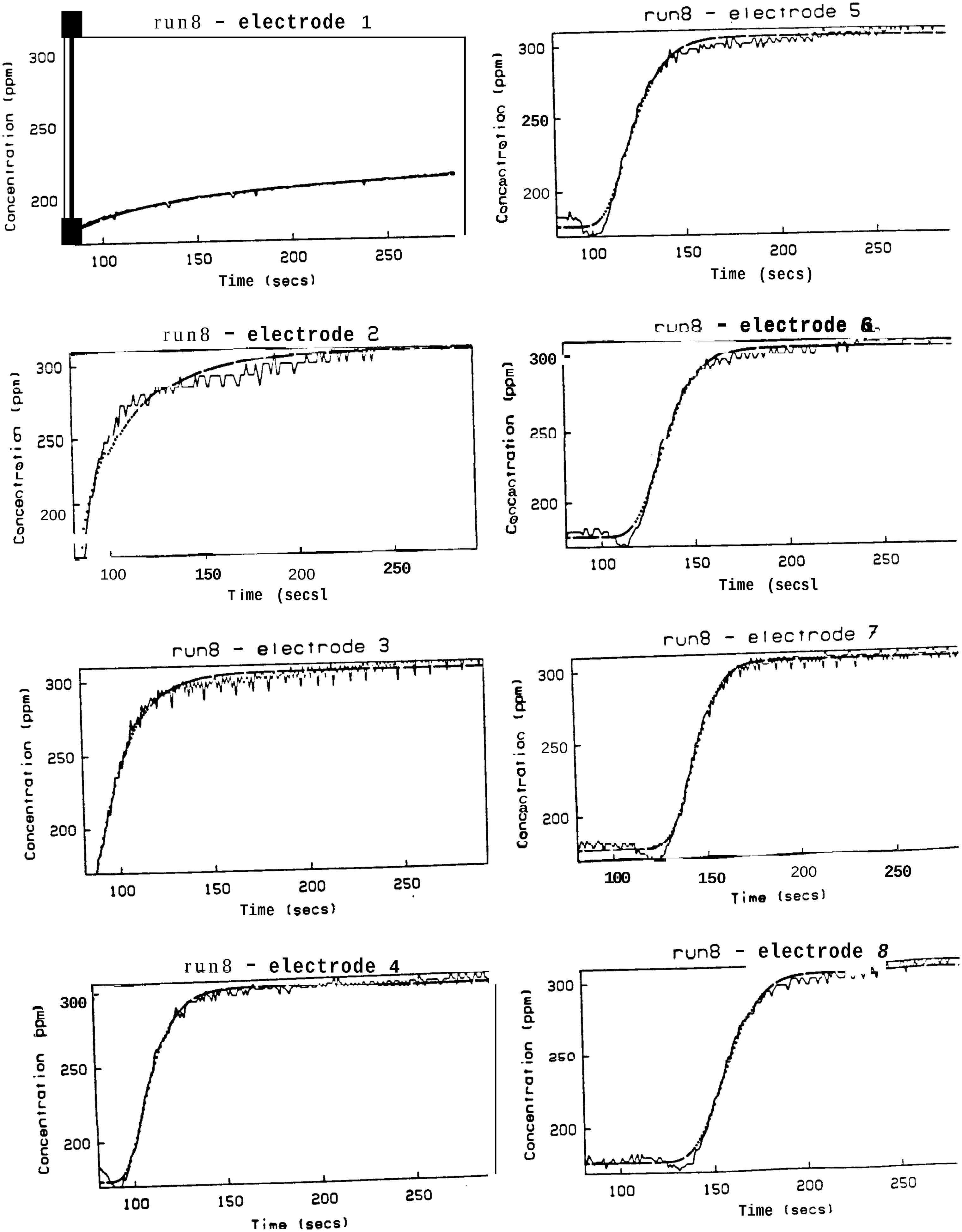
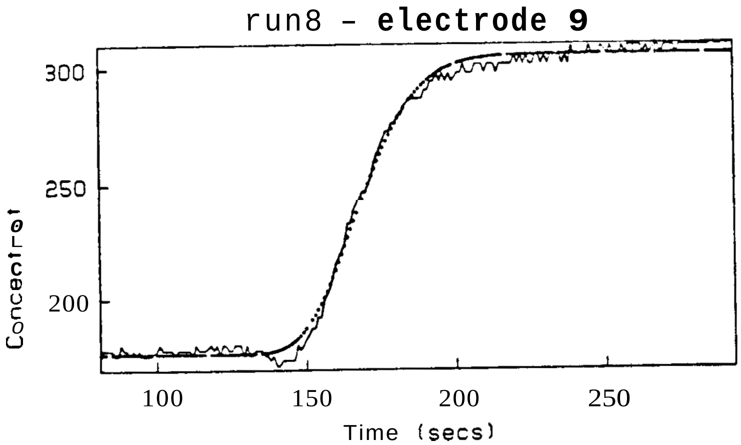
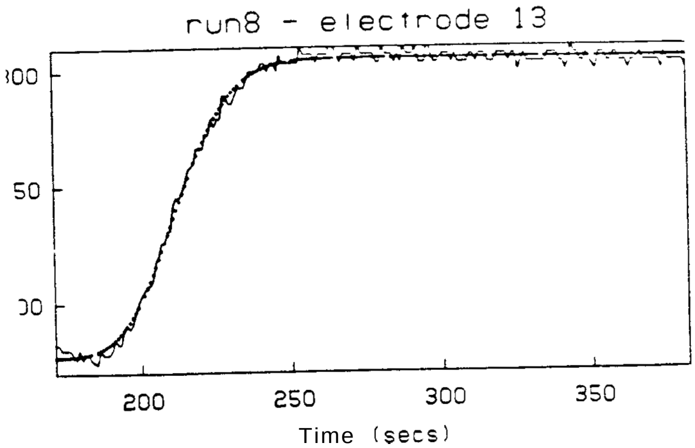
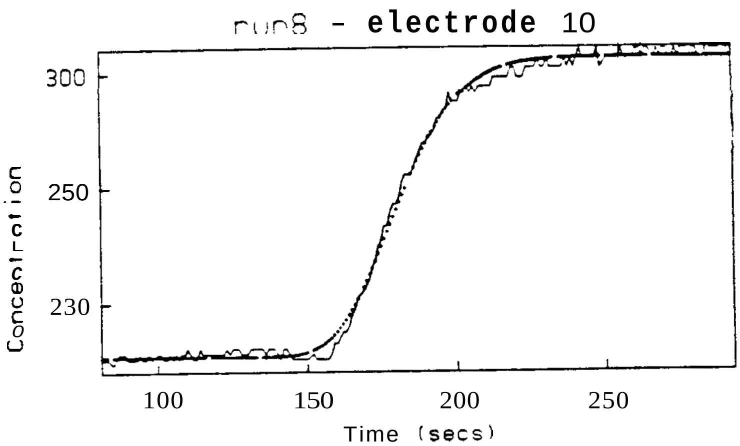


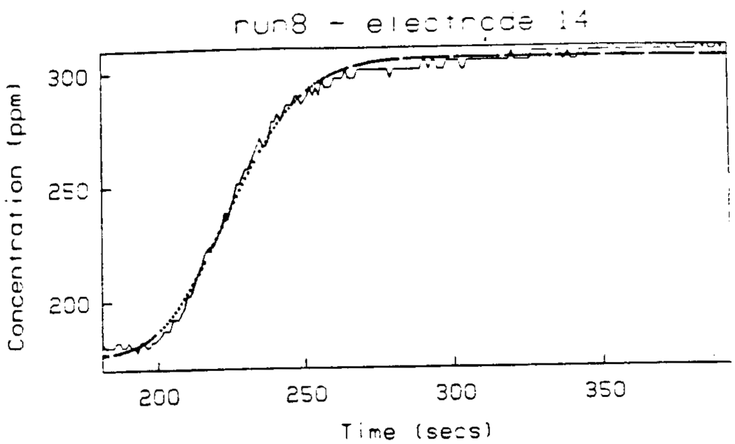

Fig. 12. Comparison of estimated and calculated dispersivity. Run 8.

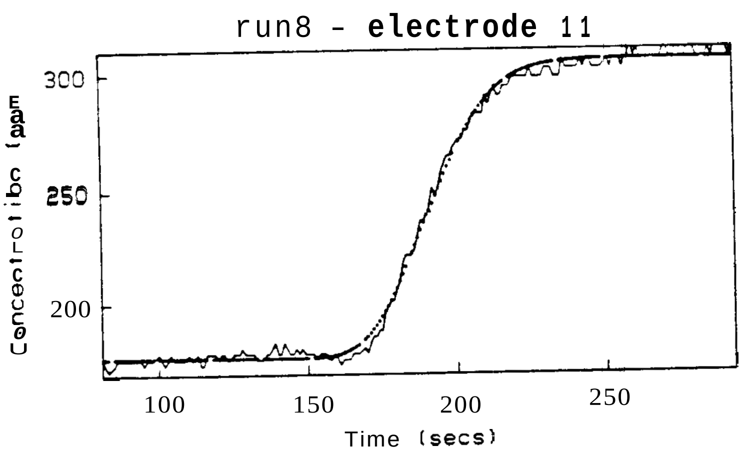

run8 - electrode 1
run8 - electrode 5
run8 - electrode 2
run8 - electrode 6
run8 - electrode 3
run8 - electrode 7
run8 - electrode 4
run8 - electrode 8
Concentration (ppm)
Time (secs)
300
250
200
100
150
200
250

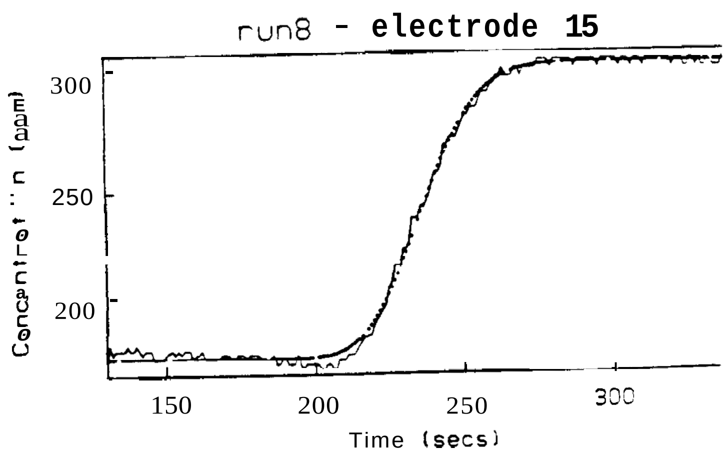

run8 - electrode 9
300
250
200
100
150
200
250
Time (secs)

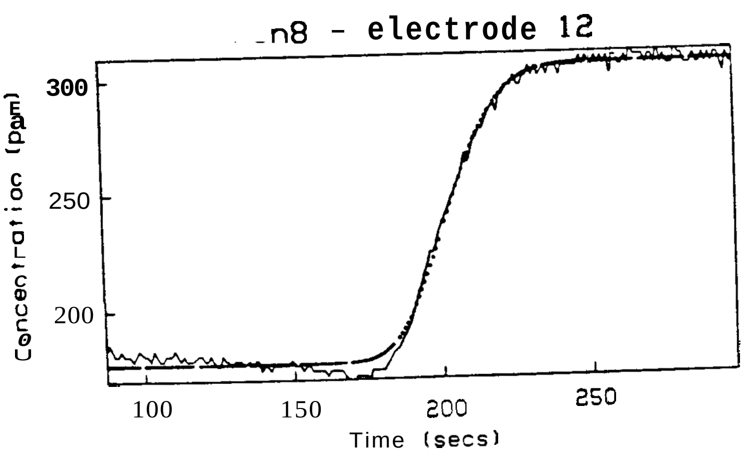

run8 - electrode 13
200
250
300
350
Time (secs)

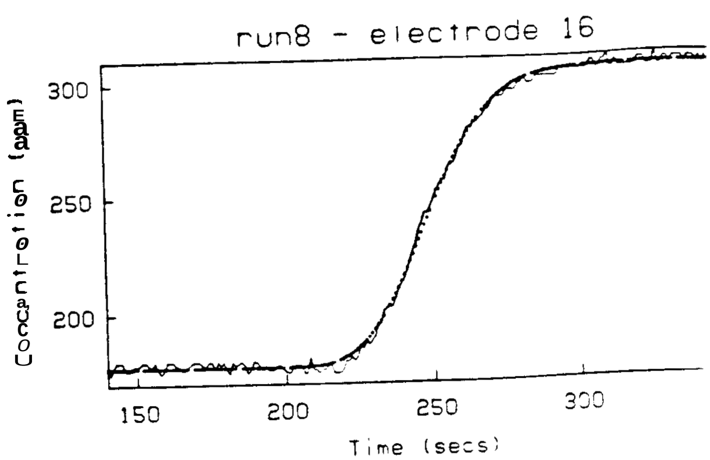

electrode 10
300
250
100
150
200
250
Time (secs)


Concentration (ppm)
300
200
200
250
300
350
Time (secs)

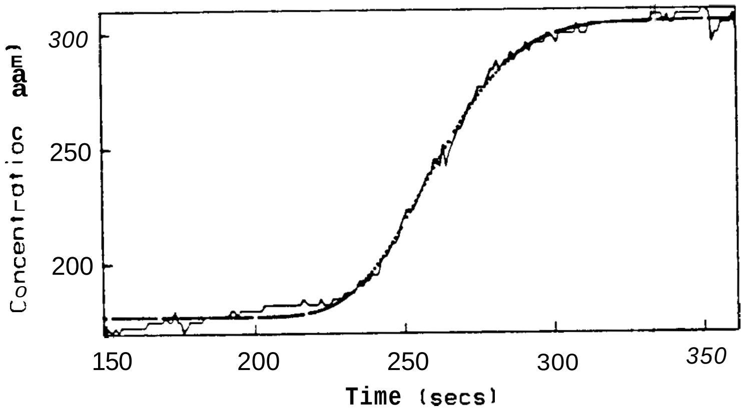

run8 - electrode 11
300
200
100
150
200
250
Time (secs)


run8 - electrode 15
300
250
200
150
200
250
300
Time (secs)

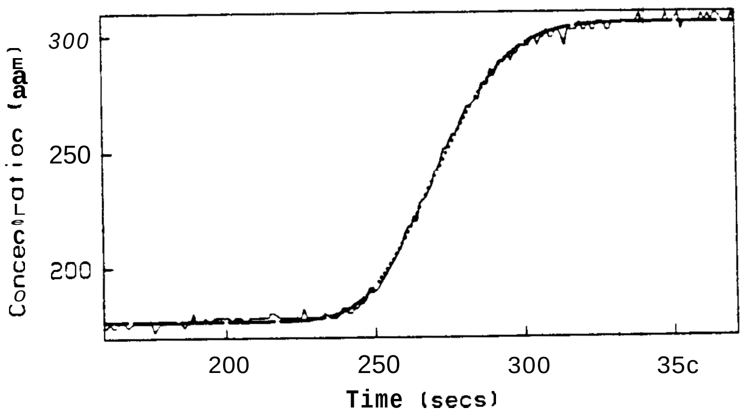

electrode 12
300
250
200
100
150
200
250
Time (secs)


run8 - electrode 16
300
250
200
150
200
250
300
Time (secs)

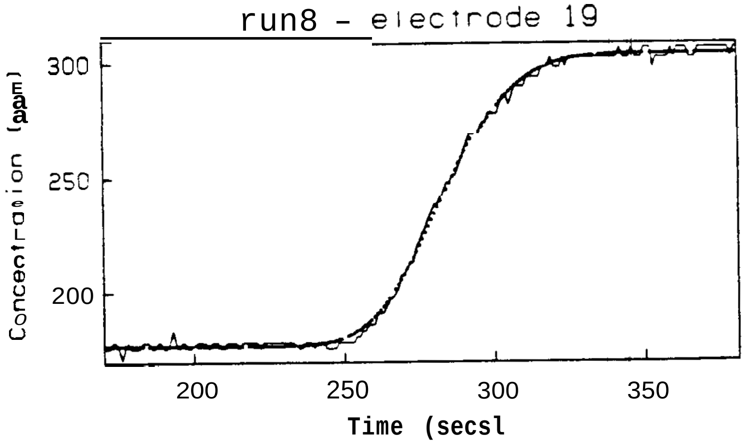
run8 - **electrode** 17



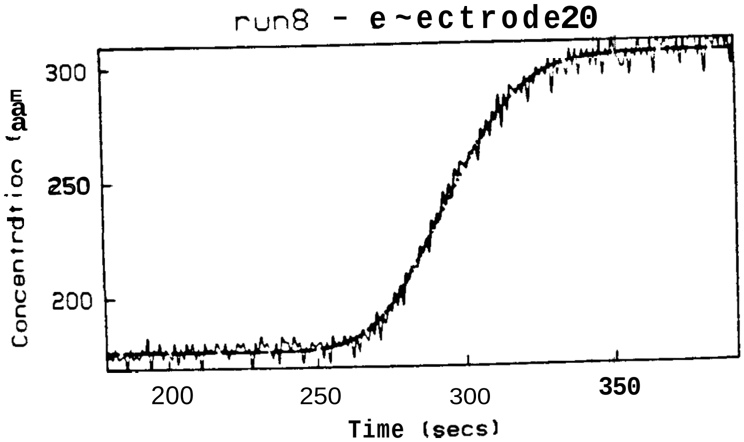

run8 - **electrode 18**



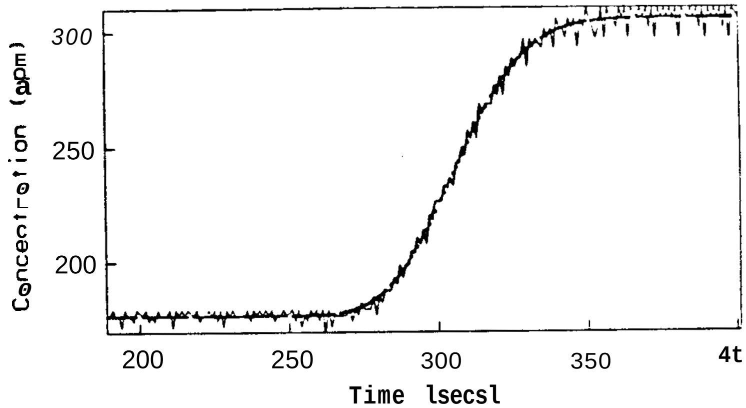

run8 - electrode 19



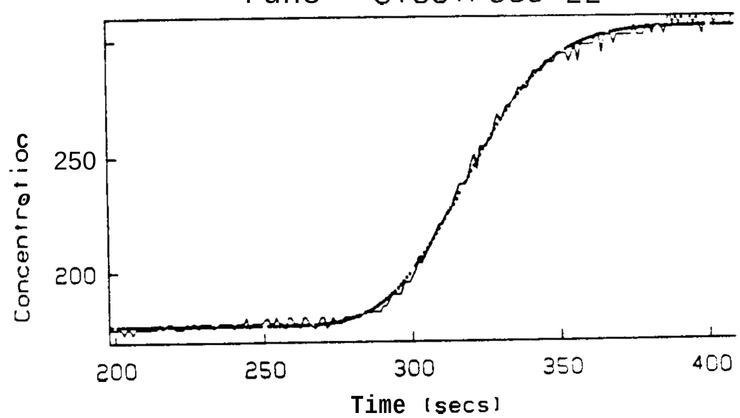

run8 - **e~ectrode20**



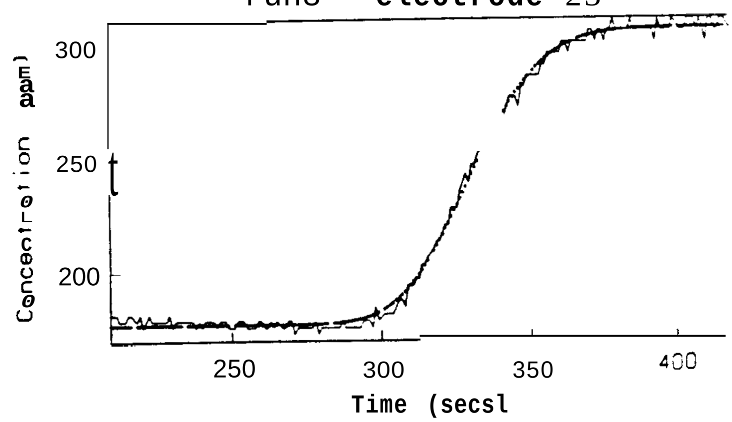

run8 - **electrode** 21



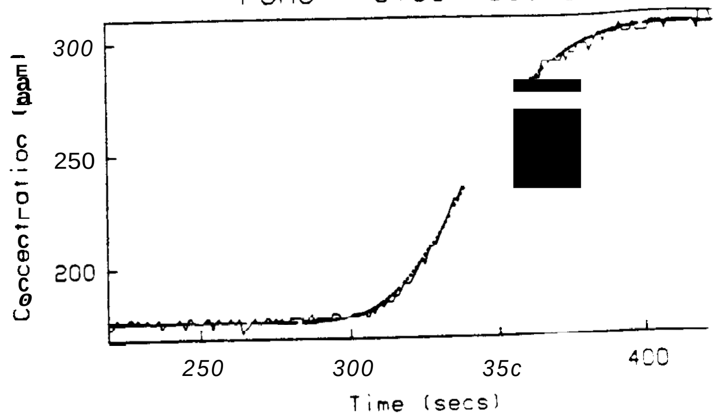

run8 - electrode 22



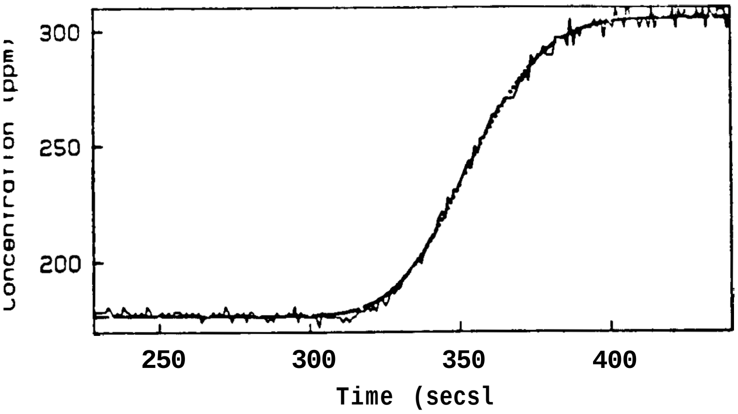

run8 - **electrode** 23



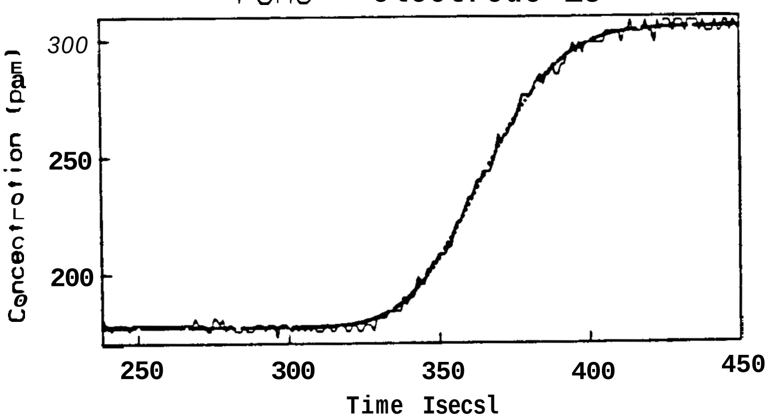

run8 - electrode 24



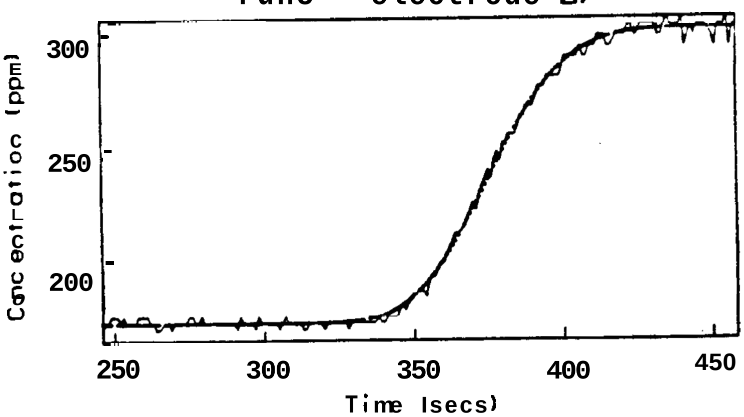
run8 - electrode 25



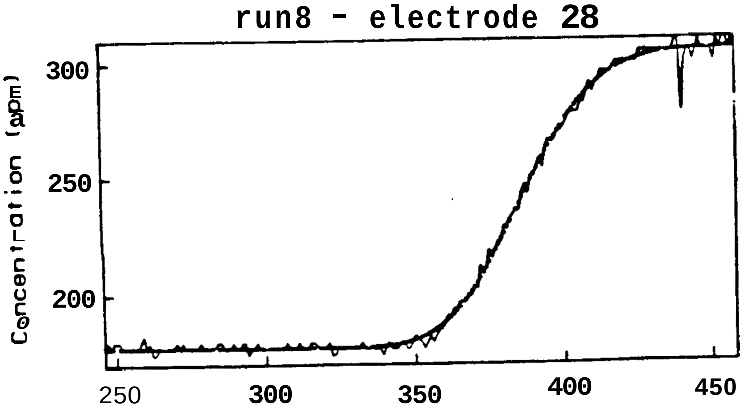

run8 - electrode 26



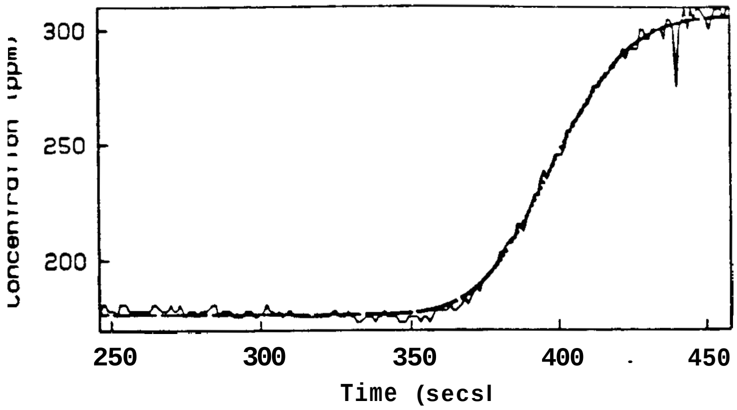

run8 - electrode 27



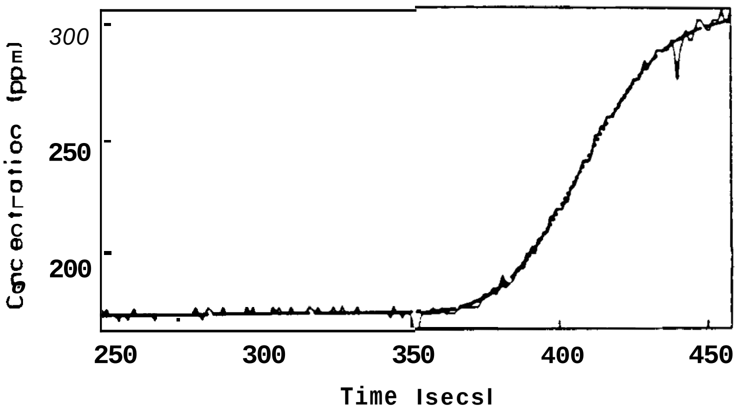

run8 - electrode 28



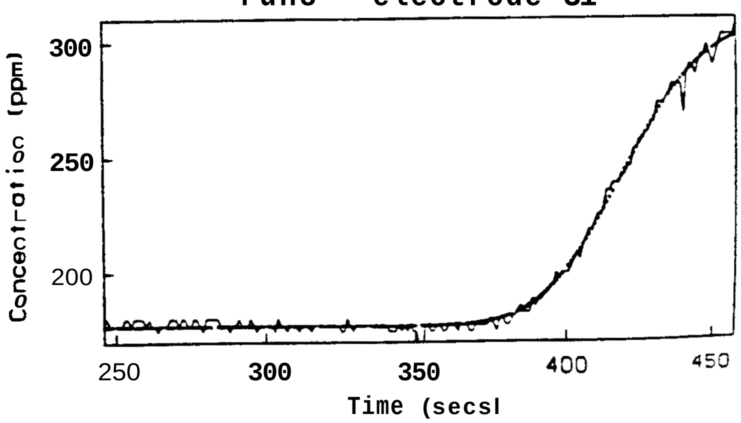

run8 - electrode 29



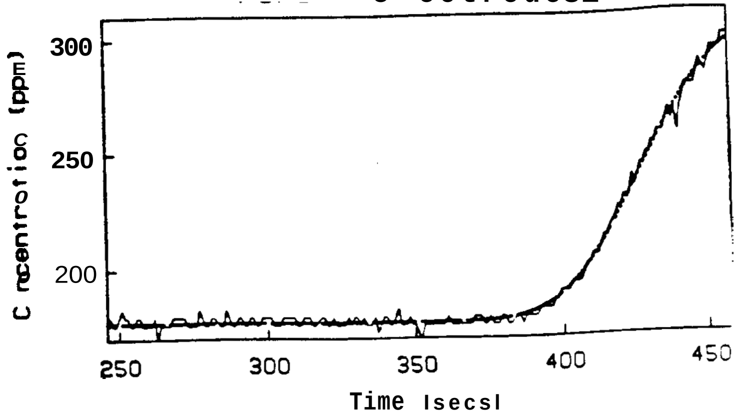

run8 - electrode 30




run8 - electrode 31




- e~ectrode32



| Table 9. RUN 9 Results | | | | | |
|---|---|---|---|---|---|
| Electrode | $C_0$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\eta$ ($cm^2/sec$) | apparent $t_0$ (sec) |
| 1 | 92.4754 | 291.9013 | 0.8247 | 11.5627 | 42.0653 |
| 2 | 131.5554 | 304.1395 | 0.7286 | 4.8659 | 41.5202 |
| 3 | 176.0719 | 304.6890 | 0.6112 | 1.1524 | 42.2768 |
| 4 | 177.1845 | 305.0923 | 0.5164 | 0.6782 | 40.4316 |
| 5 | 177.1948 | 305.0734 | 0.4881 | 0.3830 | 40.2042 |
| 6 | 176.9417 | 305.2280 | 0.4528 | 0.4245 | 40.1679 |
| 7 | 177.1461 | 305.0948 | 0.4477 | 0.2400 | 39.9412 |
| 8 | 176.9773 | 305.1975 | 0.4303 | 0.2683 | 40.4746 |
| 9 | 177.0871 | 305.1323 | 0.4284 | 0.2139 | 42.1549 |
| 10 | 176.8942 | 305.2768 | 0.4158 | 0.2379 | 42.3783 |
| 11 | 177.0735 | 305.1511 | 0.4137 | 0.1834 | 42.4097 |
| 12 | 177.1774 | 305.1143 | 0.4067 | 0.1622 | 42.4030 |
| 13 | 177.2131 | 305.1068 | 0.4007 | 0.1483 | 42.3131 |
| 14 | 177.1551 | 305.1408 | 0.3955 | 0.1653 | 42.1147 |
| 15 | 177.1852 | 305.1264 | 0.3930 | 0.1385 | 41.9828 |
| 16 | 177.0934 | 305.1743 | 0.3876 | 0.1490 | 41.9846 |
| 17 | 177.1607 | 305.1516 | 0.3871 | 0.1339 | 41.6244 |
| 18 | 177.1774 | 305.1328 | 0.3851 | 0.1163 | 42.1718 |
| 19 | 177.4015 | 305.0996 | 0.3849 | 0.1121 | 41.8483 |
| 20 | 177.2561 | 305.2126 | 0.3806 | 0.0932 | 41.5267 |
| 21 | 177.1605 | 305.2447 | 0.3809 | 0.0827 | 42.4186 |
| 22 | 177.0856 | 305.2866 | 0.3759 | 0.1076 | 42.0458 |
| 23 | 177.0395 | 305.4663 | 0.3730 | 0.1034 | 42.0218 |
| 24 | 177.0251 | 305.6692 | 0.3728 | 0.0868 | 42.1476 |
| 25 | 177.0109 | 306.2703 | 0.3708 | 0.0879 | 41.6299 |
| 26 | 177.0091 | 306.7268 | 0.3721 | 0.0894 | 42.2254 |
| 27 | 177.0063 | 308.3059 | 0.3732 | 0.0773 | 41.6680 |
| 28 | 177.0743 | 314.0203 | 0.3721 | 0.0861 | 41.9085 |
| 29 | 177.4232 | 320.7780 | 0.3729 | 0.0551 | 42.1727 |
| 30 | 177.1694 | 210.5363 | 0.3841 | 0.0251 | 41.7788 |

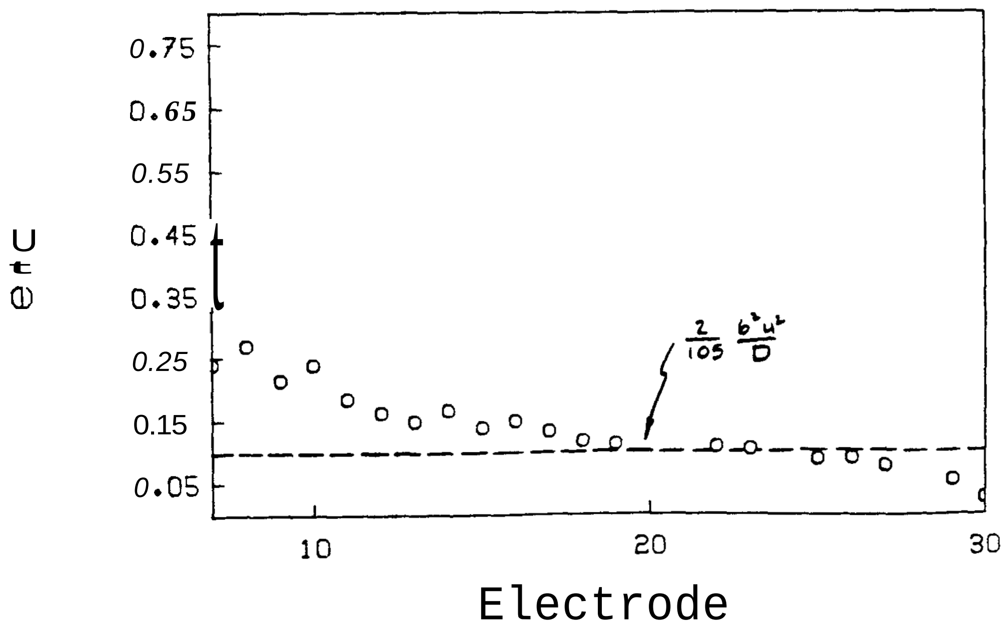
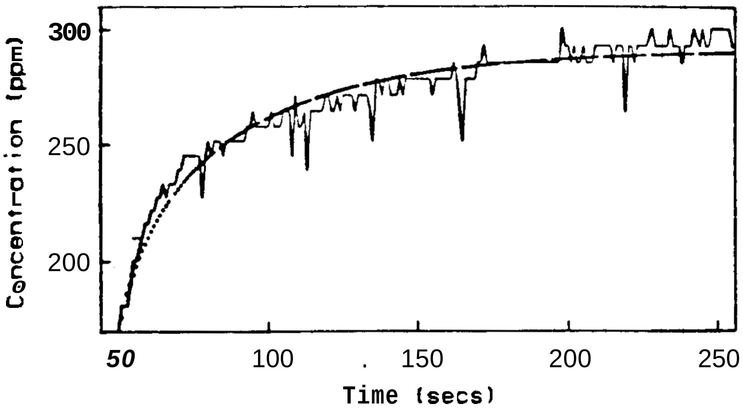
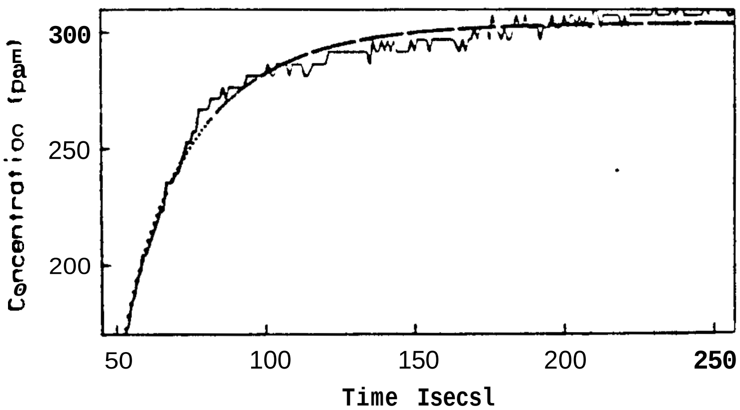



**Fig. 13.** Comparison **of estimated and** calculated **dispersivity.** Run 9.

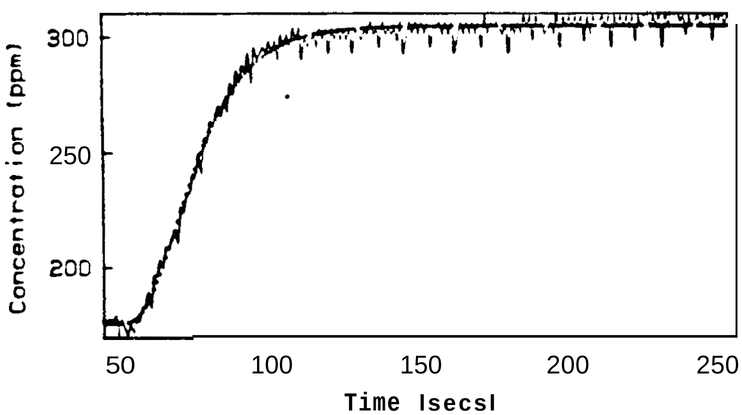
run9 - electrode 1



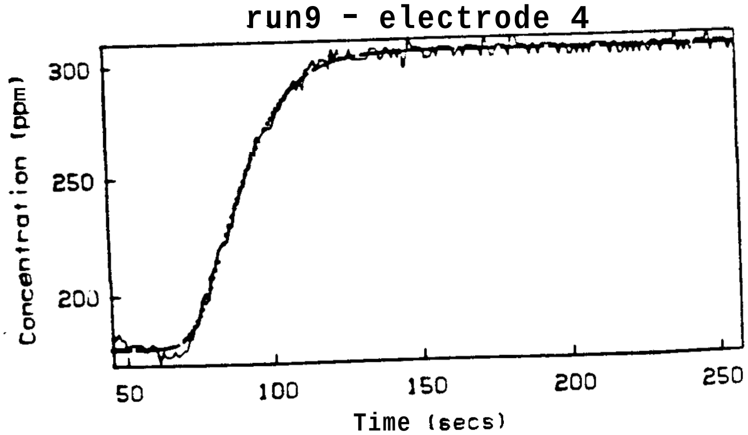

run9 - electrode 2



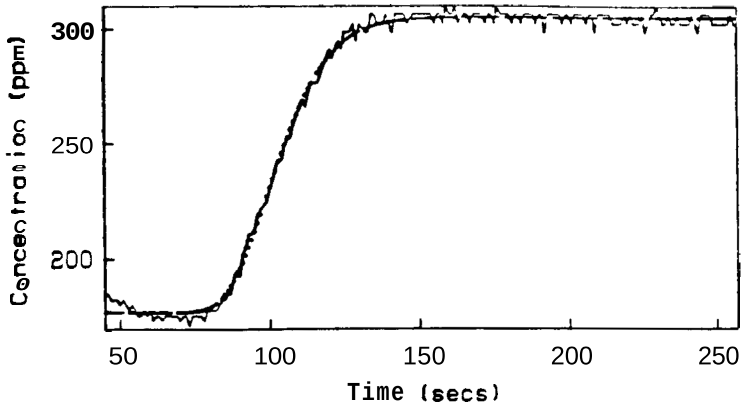

run9 - electrode 3



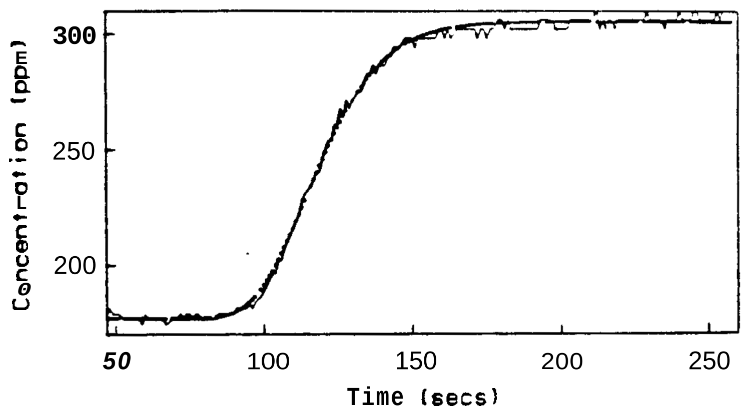

run9 - electrode 4



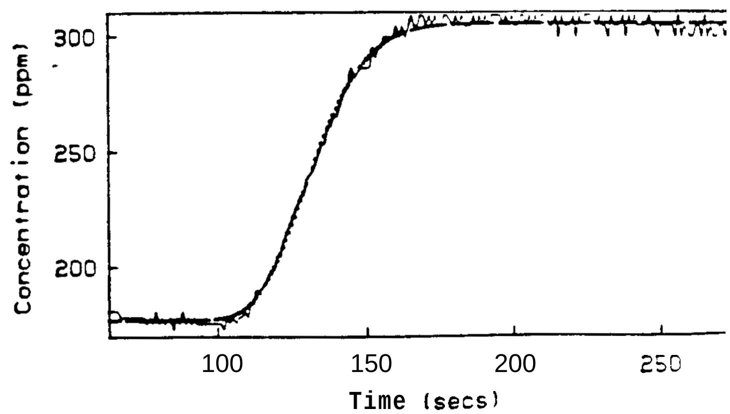

run9 - electrode 5



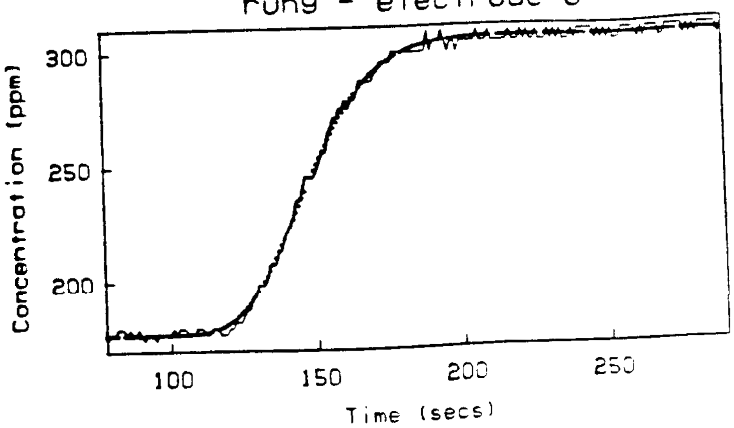

run9 - electrode 6



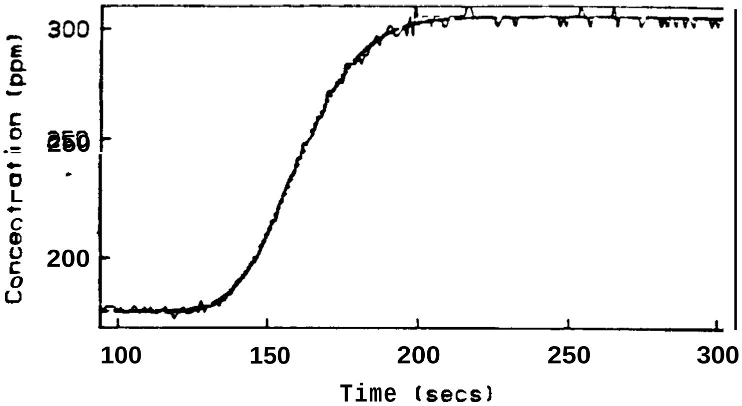

run9 - electrode ?



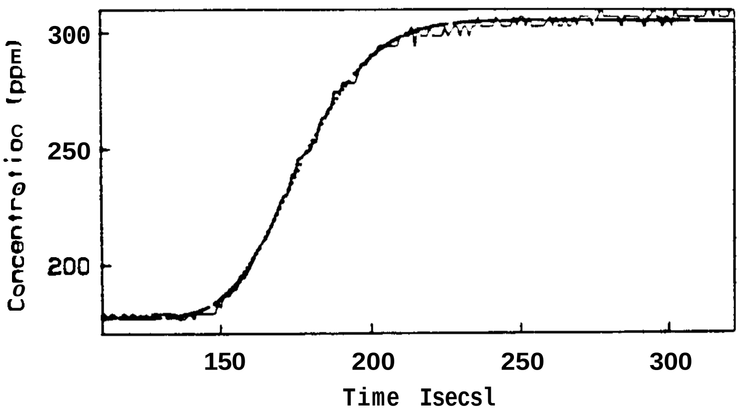

run9 - electrode 8



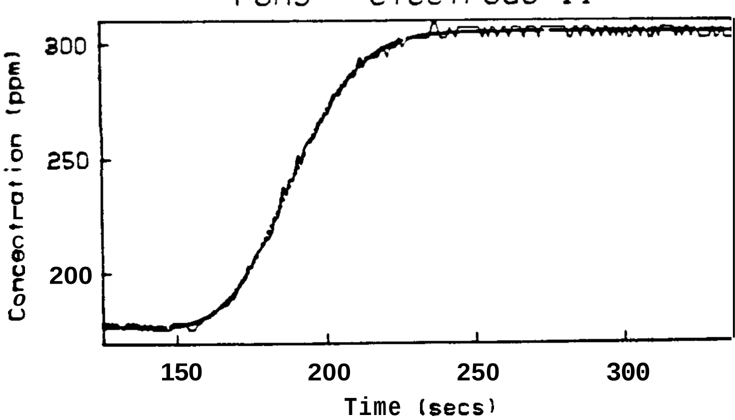
run9 - electrode 9



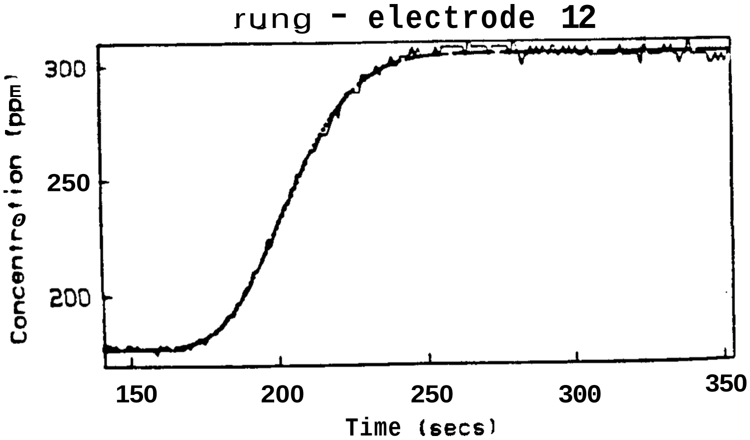

run9 - electrode 10



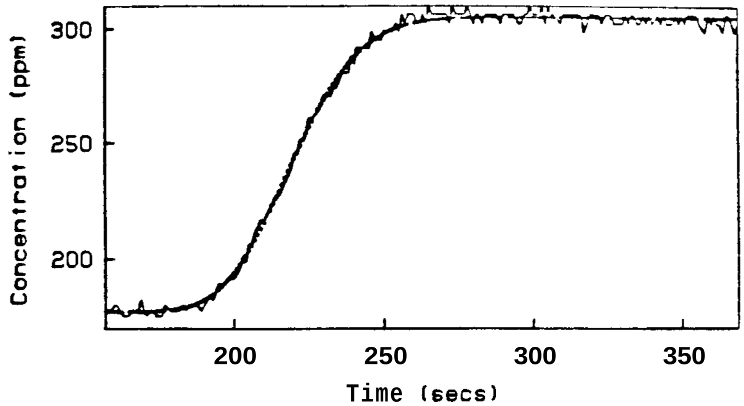

run9 - electrode 11



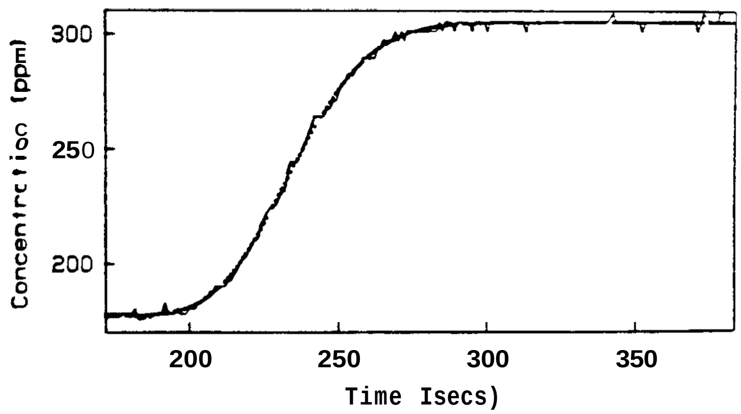

run9 - electrode 12



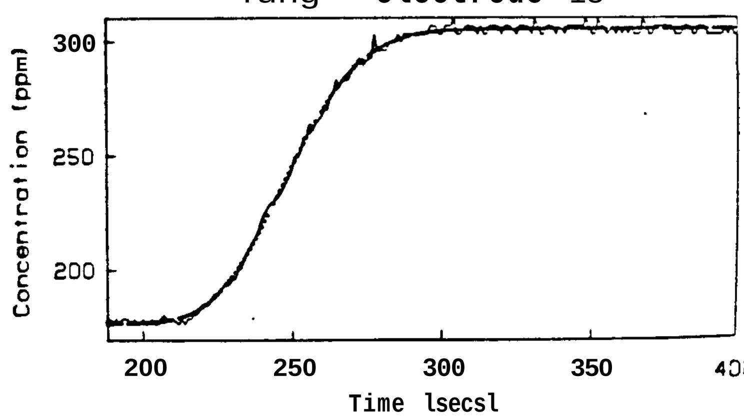

run9 - electrode 13



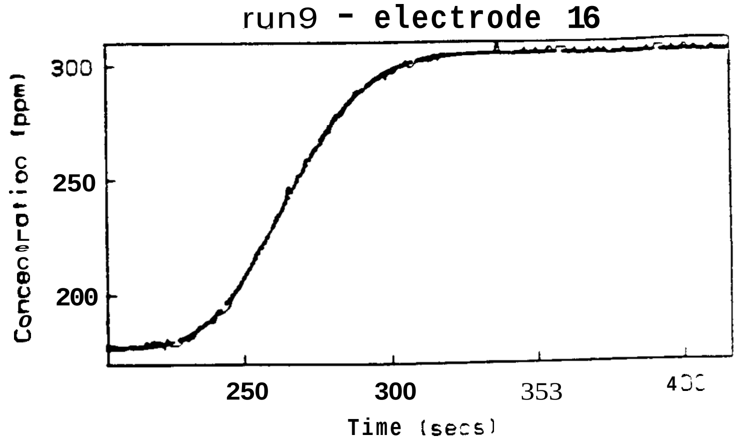

run9 - electrode 14



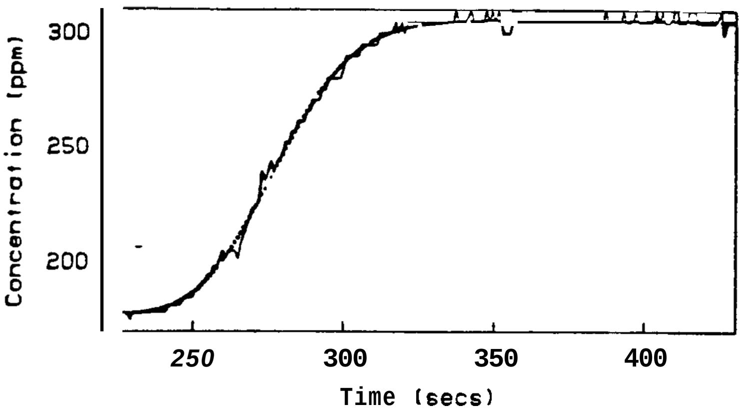

run9 - electrode 15



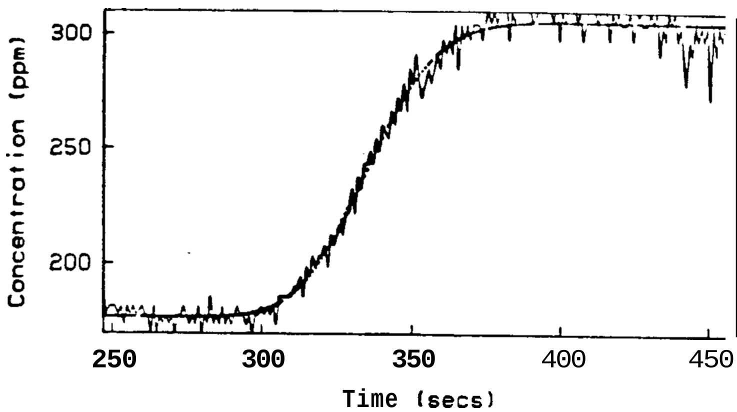

run9 - electrode 16



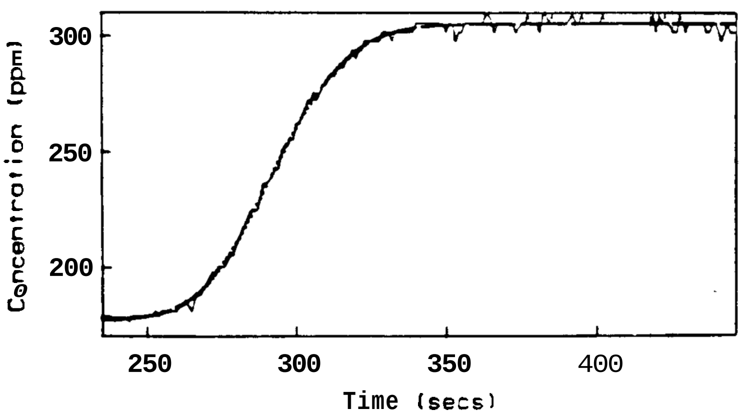
run9 - **electrode 17**



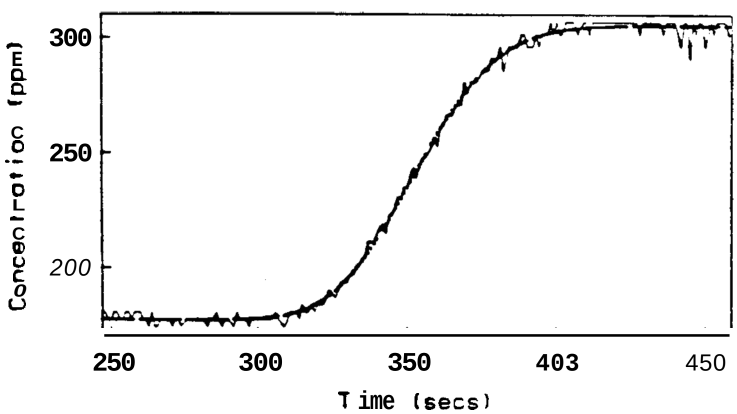

run9 - **electrode 21**



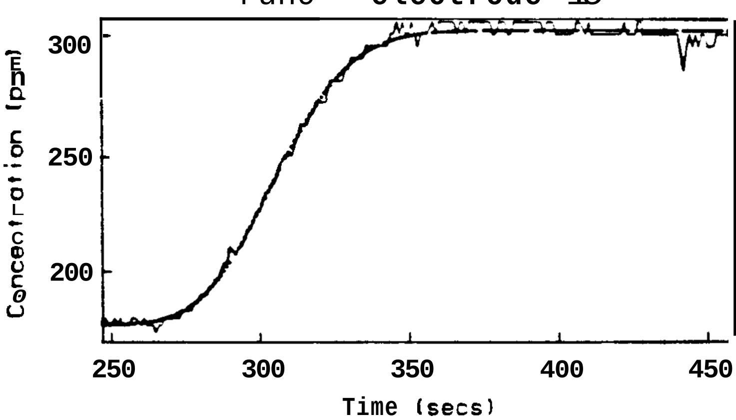

run9 - **electrode** 18



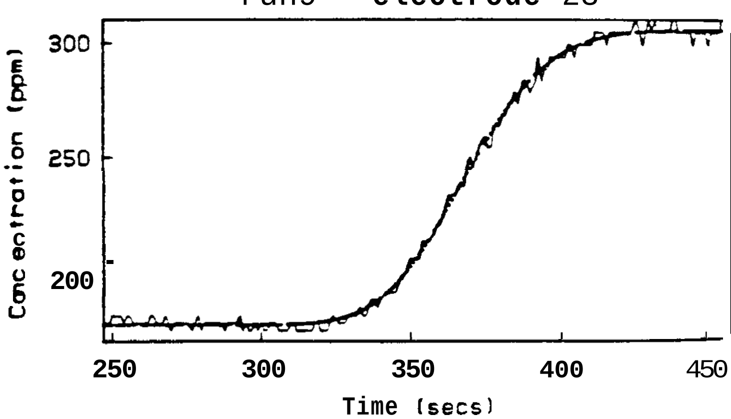

run9 - **electrode** 22



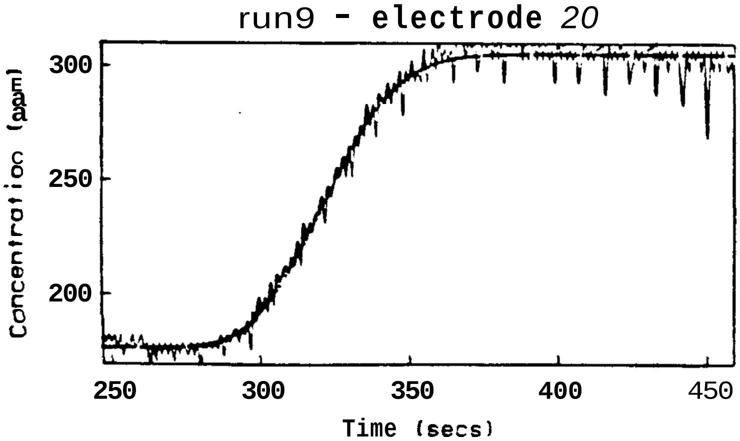

run9 - **electrode** 19



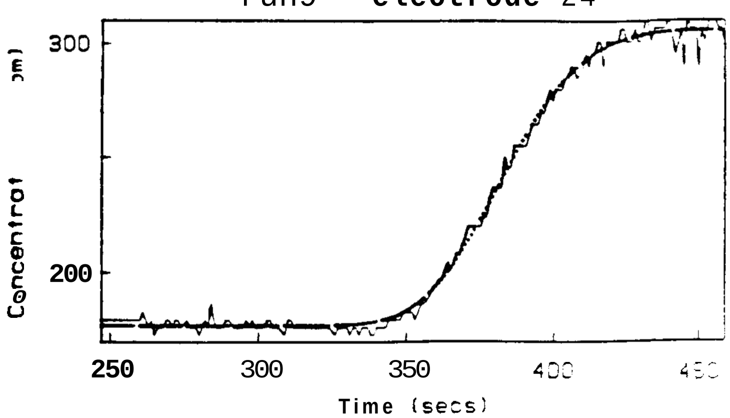

run9 - **electrode** 23



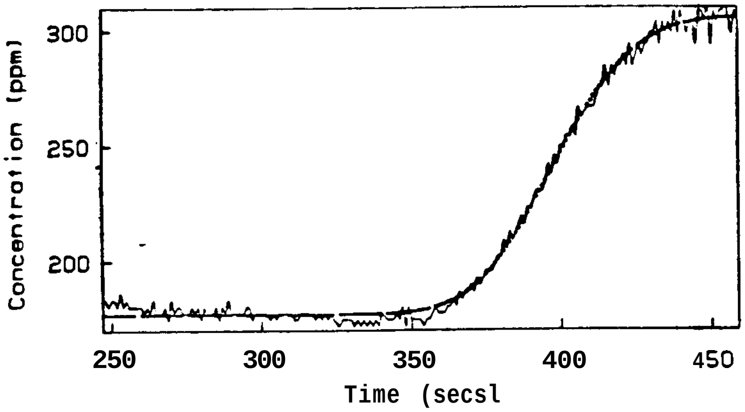

run9 - **electrode** *20*



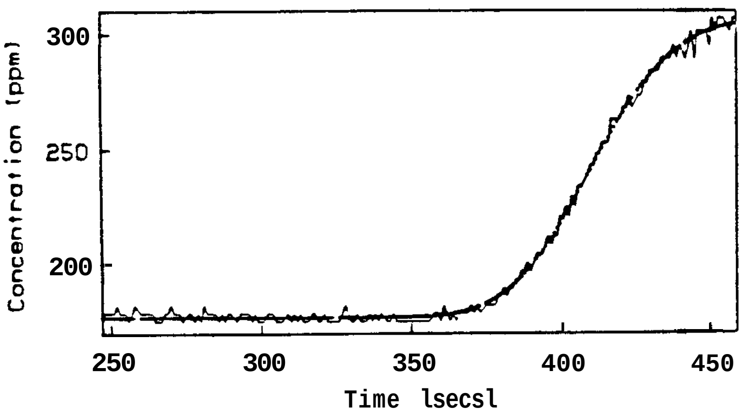

run9 - **electrode** 24



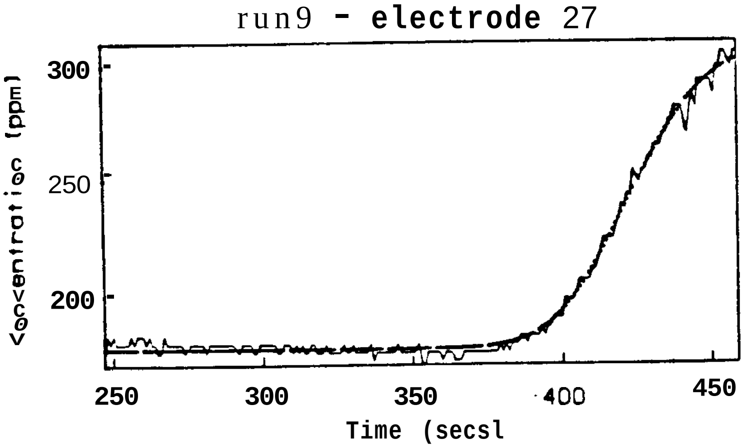
run9 - electrode 25



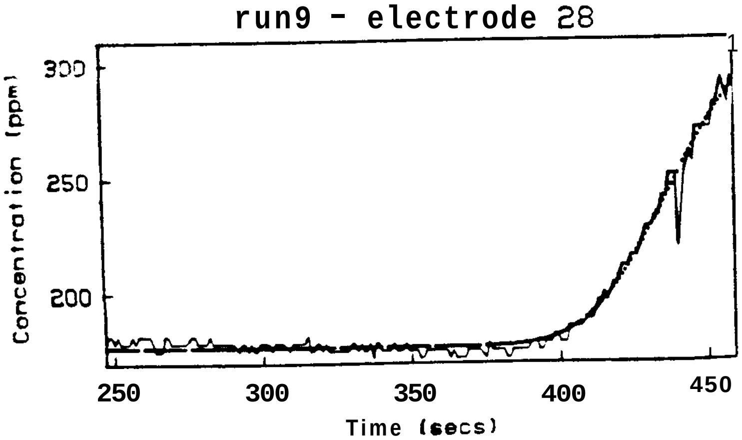

run9 - electrode 26



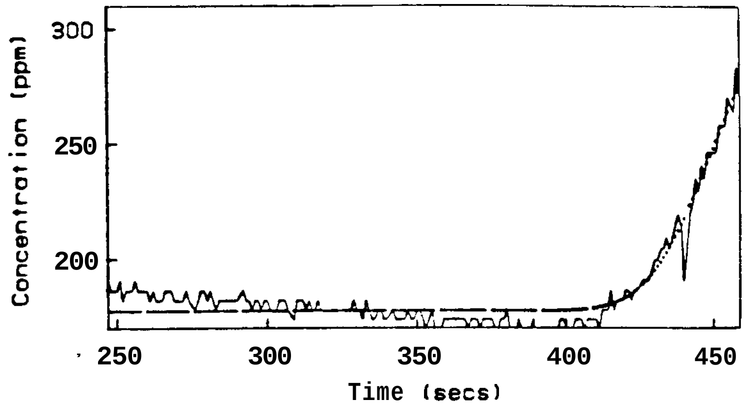

run9 - electrode 27



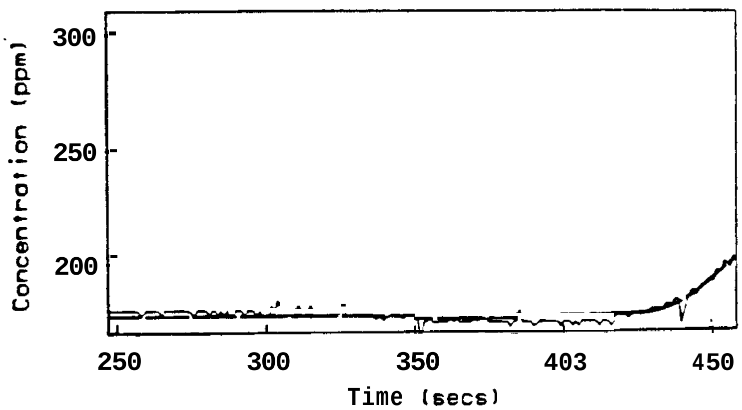

run9 - electrode 28




run9 - electrode 29




run9 - electrode 30



| Table 10. RUN 10 Results | | | | | |
|---|---|---|---|---|---|
| Electrode | $C_o$ (ppm) | $C_1$ (ppm) | u (cm/sec) | $\eta$ ($cm^2$/sec) | apparent $t_o$ (sec) |
| 1 | 155.5770 | 301.8529 | 0.7688 | 1.4091 | 42.1981 |
| 2 | 155.8977 | 305.9257 | 0.6398 | 2.1460 | 41.6750 |
| 3 | 174.2000 | 304.6744 | 0.5717 | 0.7228 | 42.0110 |
| 4 | 176.9750 | 305.1616 | 0.5180 | 0.4317 | 40.5803 |
| 5 | 177.1250 | 305.1054 | 0.5005 | 0.3067 | 40.7347 |
| 6 | 176.9881 | 305.1995 | 0.4778 | 0.3406 | 40.2335 |
| 7 | 177.1116 | 305.1659 | 0.4784 | 0.2061 | 41.7093 |
| 8 | 177.1743 | 305.3780 | 0.4620 | 0.2459 | 40.7168 |
| 9 | 177.0269 | 305.1664 | 0.4610 | 0.2013 | 42.4894 |
| 10 | 176.9113 | 305.3792 | 0.4508 | 0.2708 | 41.7489 |
| 11 | 177.0229 | 305.1925 | 0.4496 | 0.1863 | 42.1240 |
| 12 | 176.9876 | 305.2614 | 0.4431 | 0.1885 | 41.6820 |
| 13 | 177.0255 | 305.2101 | 0.4389 | 0.1720 | 41.6887 |
| 14 | 177.0176 | 305.2232 | 0.4346 | 0.1776 | 42.3983 |
| 15 | 177.0756 | 305.1718 | 0.4322 | 0.1668 | 41.6534 |
| 16 | 177.0307 | 305.2253 | 0.4272 | 0.1629 | 41.5687 |
| 17 | 177.0804 | 305.2016 | 0.4290 | 0.1566 | 41.6011 |
| 18 | 177.0782 | 305.1882 | 0.4240 | 0.1542 | 42.1210 |
| 19 | 177.0610 | 305.1971 | 0.4258 | 0.1273 | 42.1639 |
| 20 | 177.0654 | 305.3203 | 0.4211 | 0.1299 | 42.4330 |
| 21 | 177.2330 | 305.1388 | 0.4212 | 0.1184 | 42.4194 |
| 22 | 177.0714 | 305.2319 | 0.4188 | 0.1301 | 40.7729 |
| 23 | 177.0949 | 305.1991 | 0.4164 | 0.1221 | 41.9701 |
| 24 | 177.1479 | 305.1808 | 0.4158 | 0.1174 | 42.3275 |
| 25 | 177.0338 | 305.4143 | 0.4142 | 0.1258 | 41.8614 |
| 26 | 177.0397 | 305.4812 | 0.4150 | 0.1262 | 42.2972 |
| 27 | 177.0394 | 305.5894 | 0.4147 | 0.1160 | 41.7694 |
| 28 | 177.0389 | 305.8246 | 0.4147 | 0.1092 | 41.5514 |
| 29 | 176.9966 | 306.9598 | 0.4128 | 0.0969 | 41.6143 |
| 30 | 177.0095 | 308.2523 | 0.4127 | 0.1110 | 42.1783 |
| 31 | 176.9899 | 314.8080 | 0.4107 | 0.0983 | 41.9902 |
| 32 | 177.0404 | 316.6703 | 0.4156 | 0.0831 | 42.4431 |

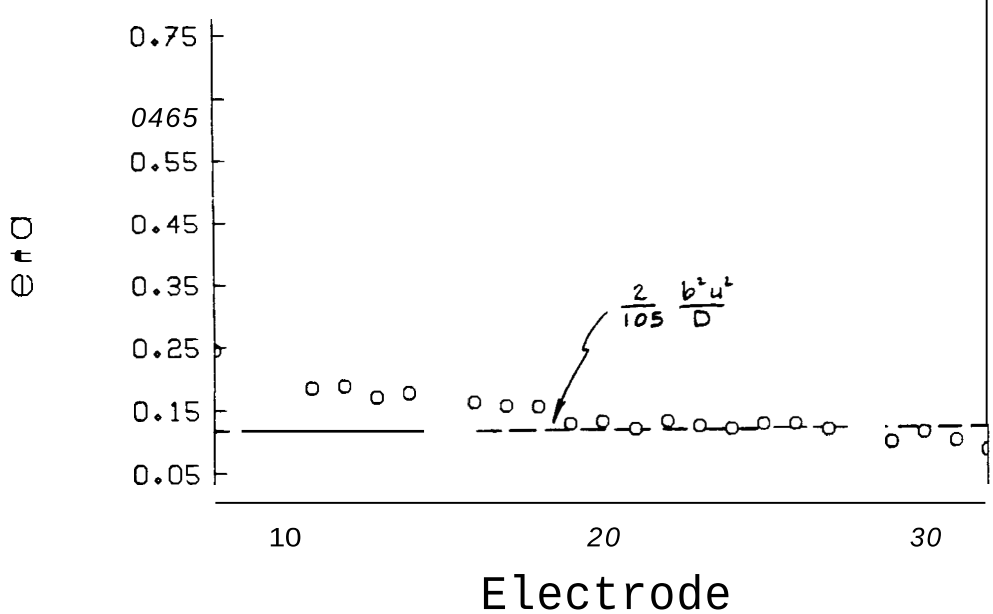
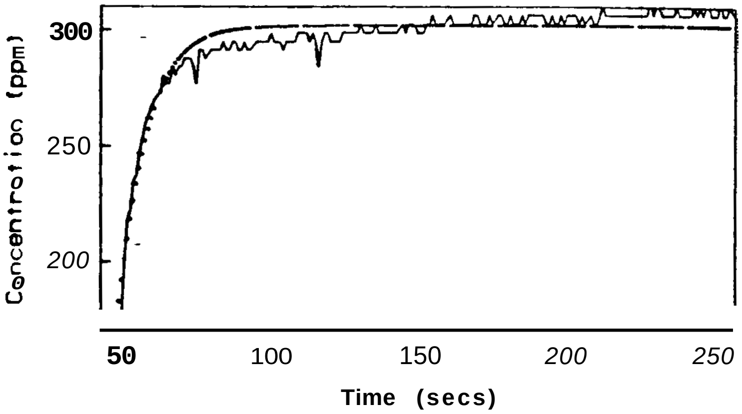
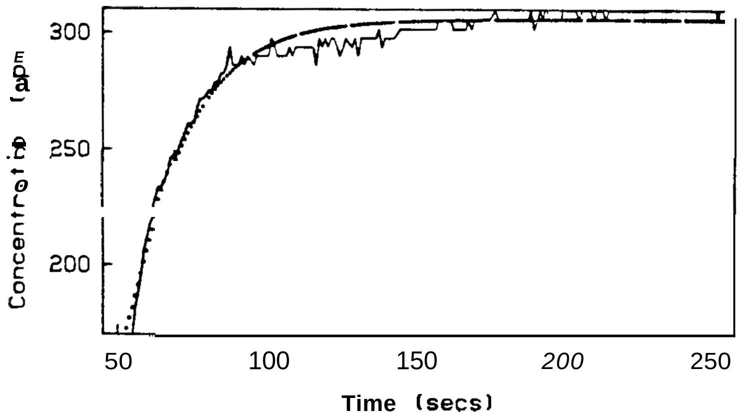



Fig. **14. Comparison of estimated and calculated dispersivity.** Run 10.

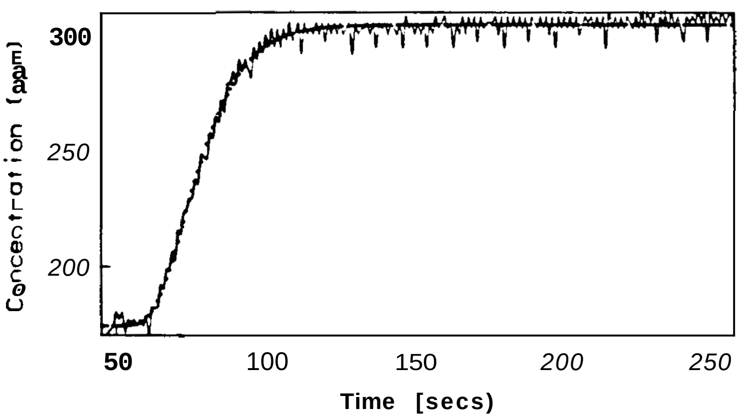
run10 - electrode 1



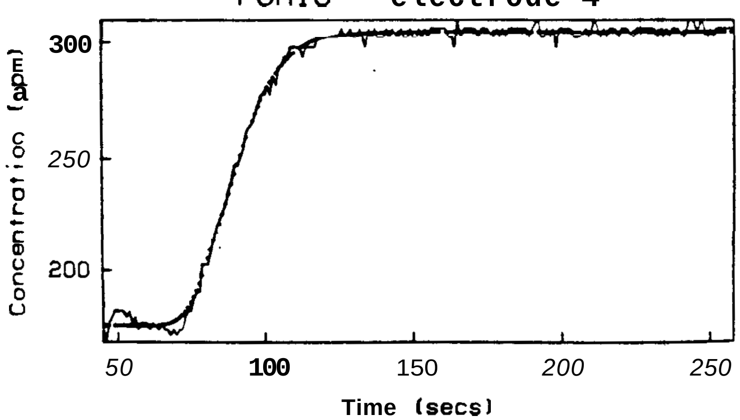

run10 - electrode 2



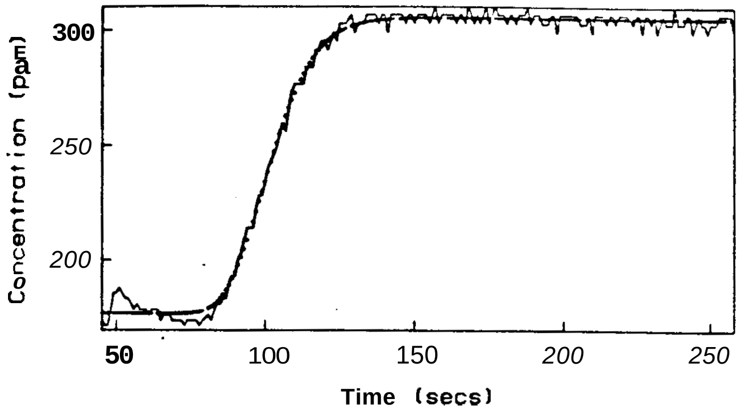

run10 - electrode 3



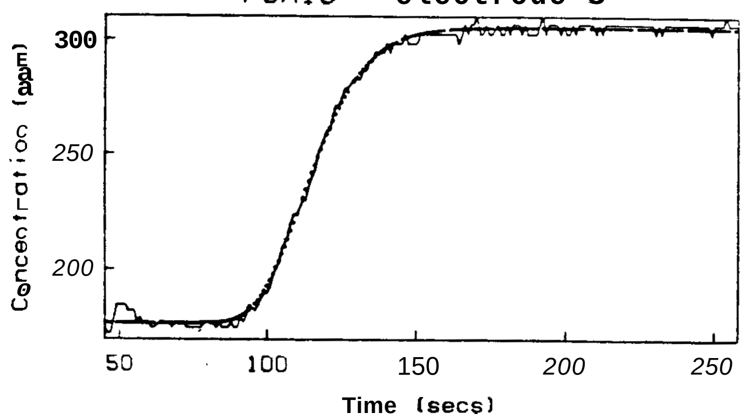

run10 - electrode 4



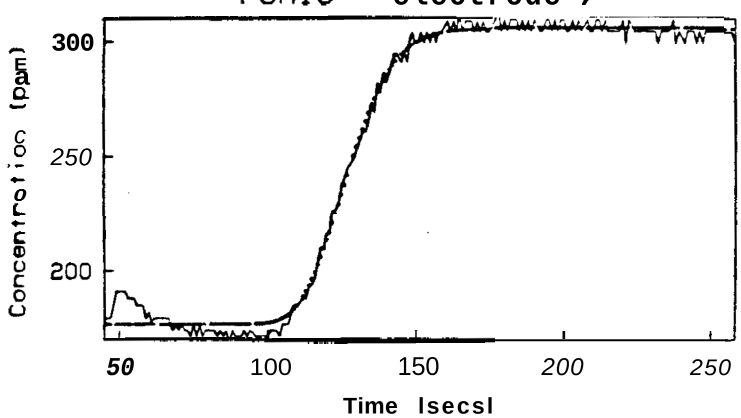

run10 - electrode 5



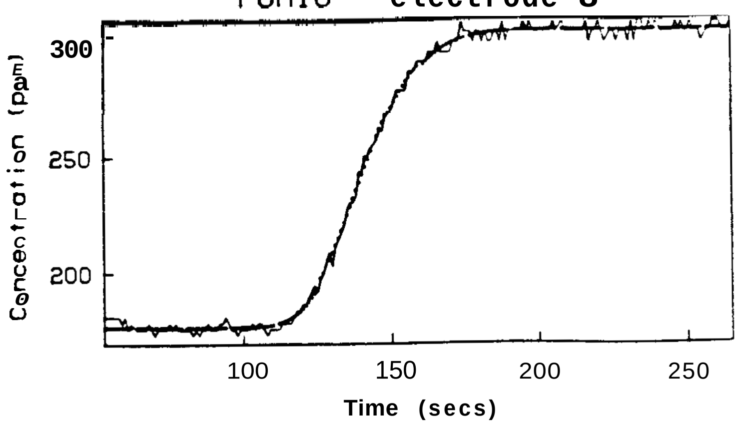

run10 - electrode 6

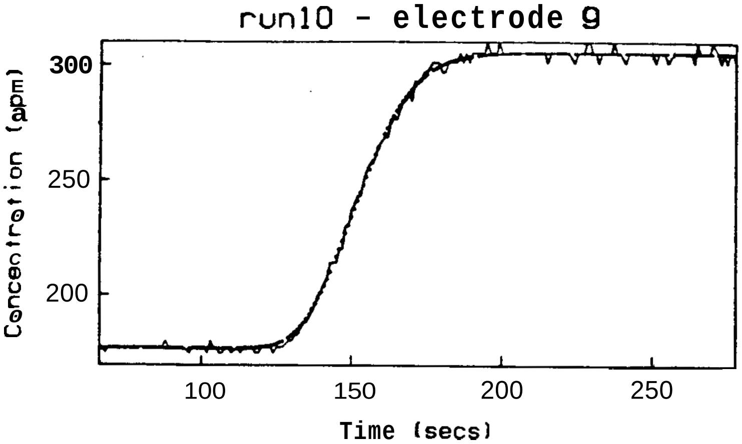


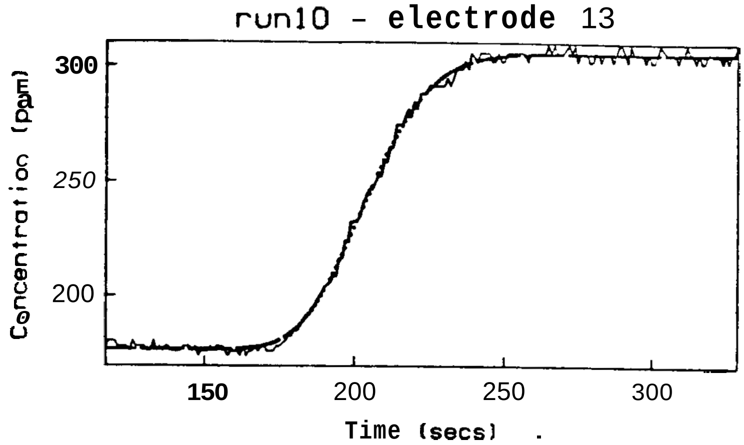

run10 - electrode 7

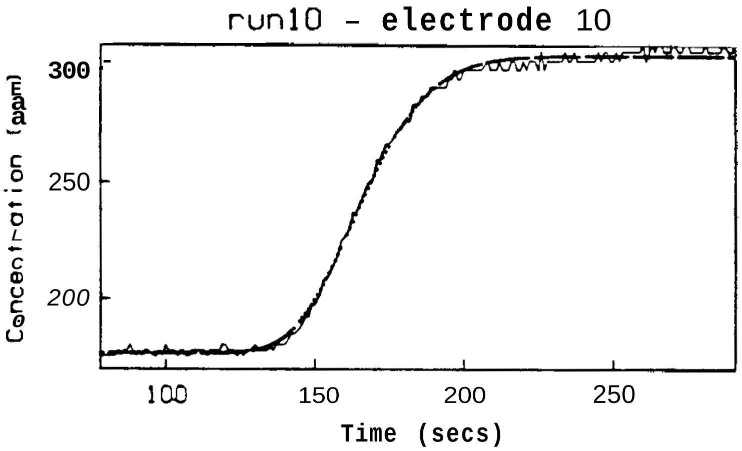


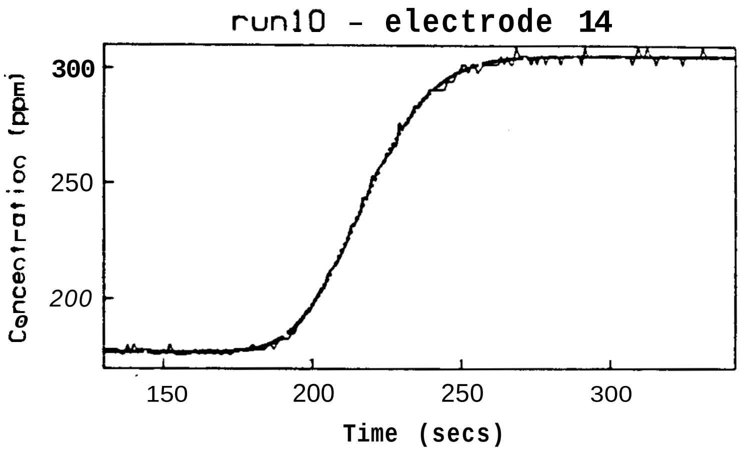

run10 - electrode 8

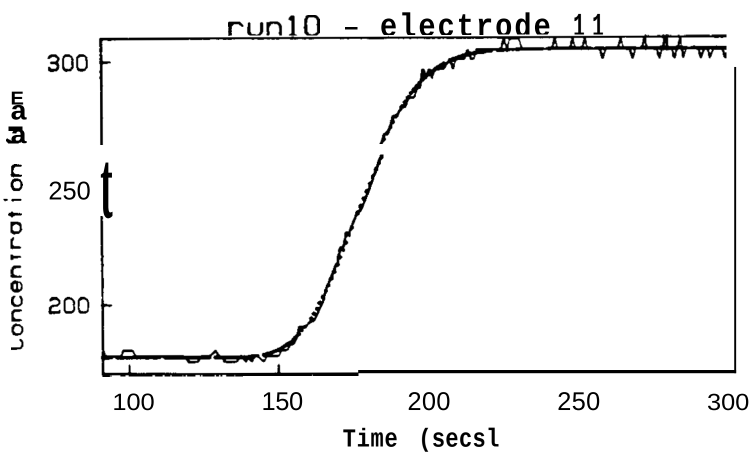


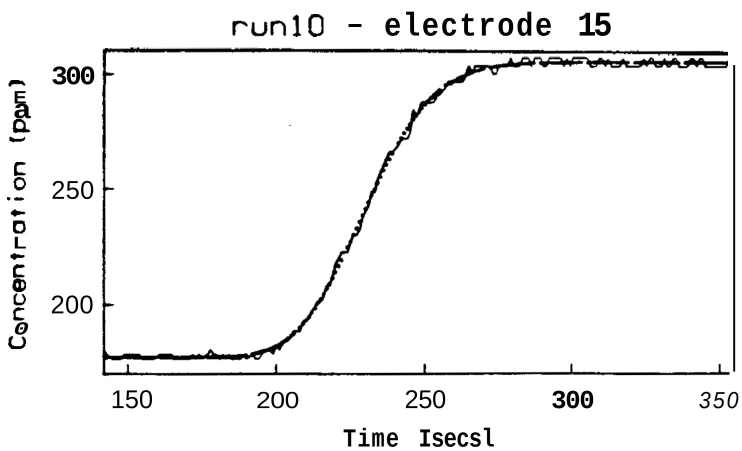

run10 - electrode 9
300
250
200
Concentration (ppm)
100
150
200
250
Time (secs)

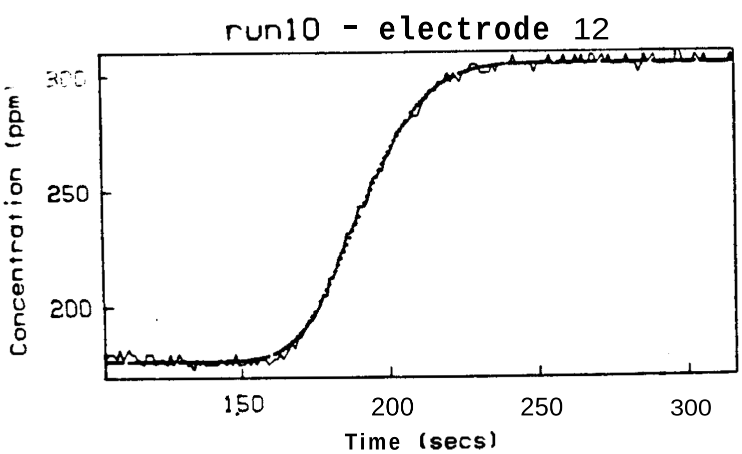

run10 - electrode 13
300
250
200
Concentration (ppm)
150
200
250
300
Time (secs)

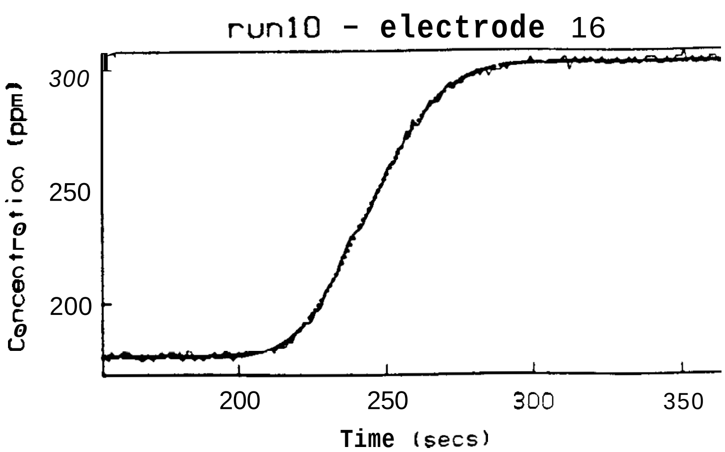

run10 - electrode 10
300
250
200
Concentration (ppm)
100
150
200
250
Time (secs)


run10 - electrode 14
300
250
200
Concentration (ppm)
150
200
250
300
Time (secs)

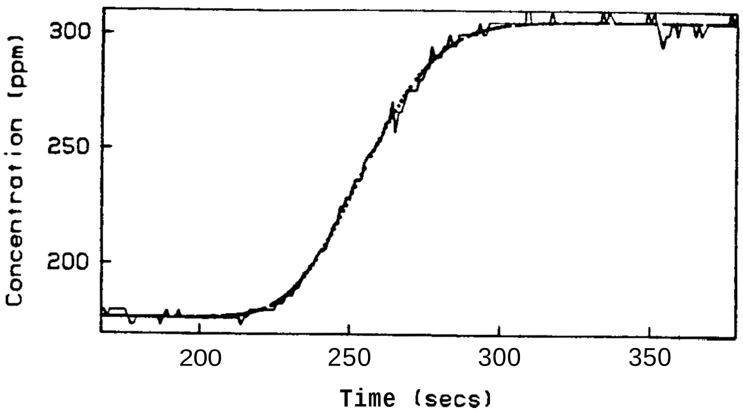

run10 - electrode 11
300
250
200
Concentration
100
150
200
250
300
Time (secs)


run10 - electrode 15
300
250
200
Concentration (ppm)
150
200
250
300
350
Time (secs)

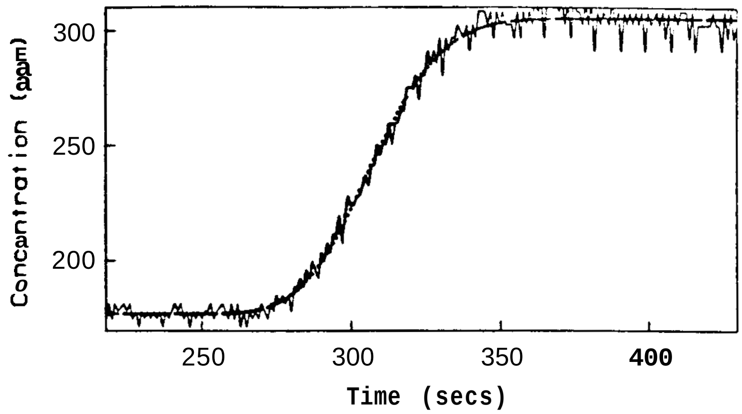

run10 - electrode 12
250
200
Concentration (ppm)
150
200
250
300
Time (secs)


run10 - electrode 16
300
250
200
Concentration (ppm)
200
250
300
350
Time (secs)

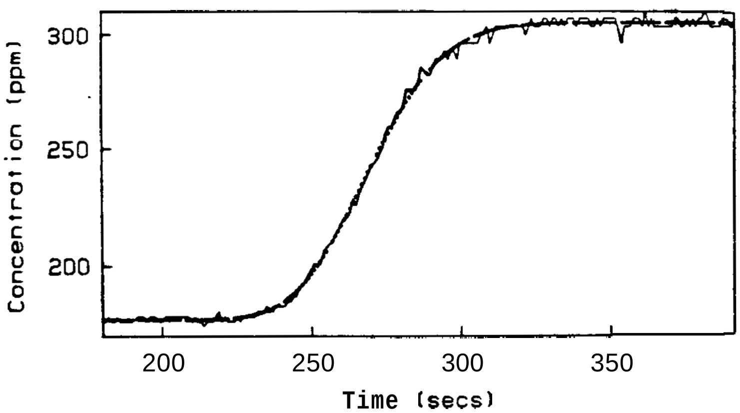
run10 - electrode 17



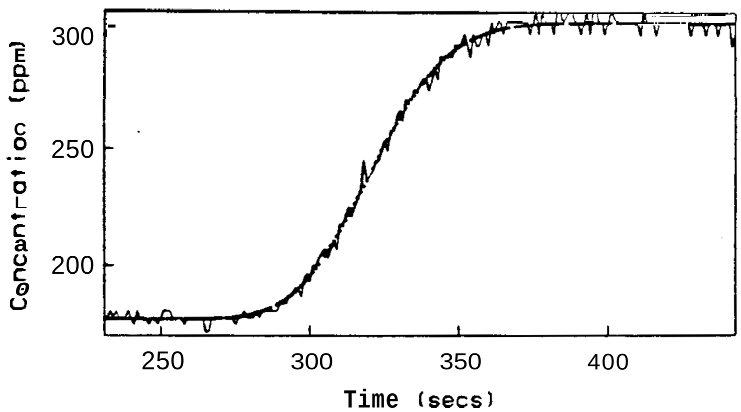

run10 - electrode 21



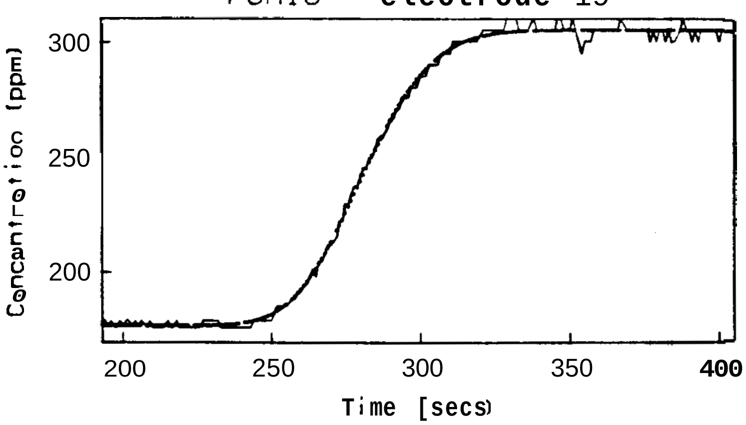

run10 - electrode 18



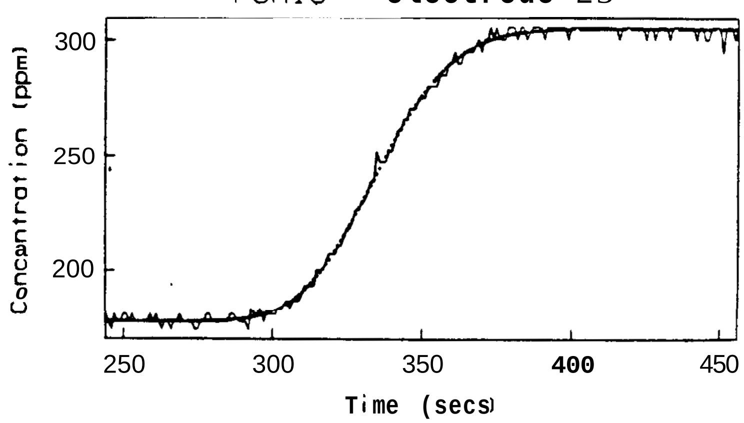

run10 - electrode 22



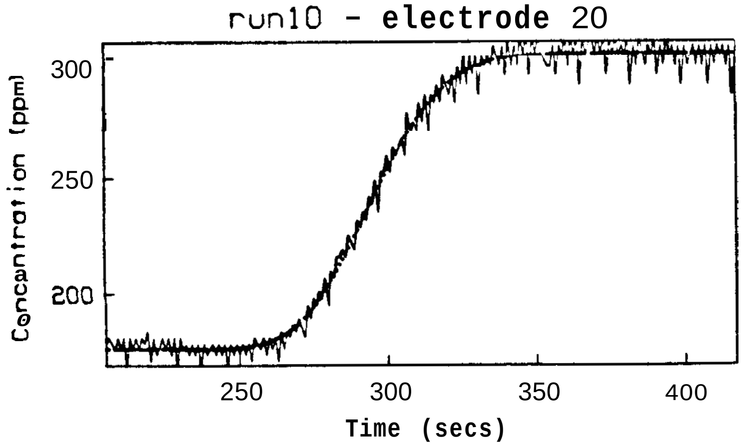

run10 - electrode 19



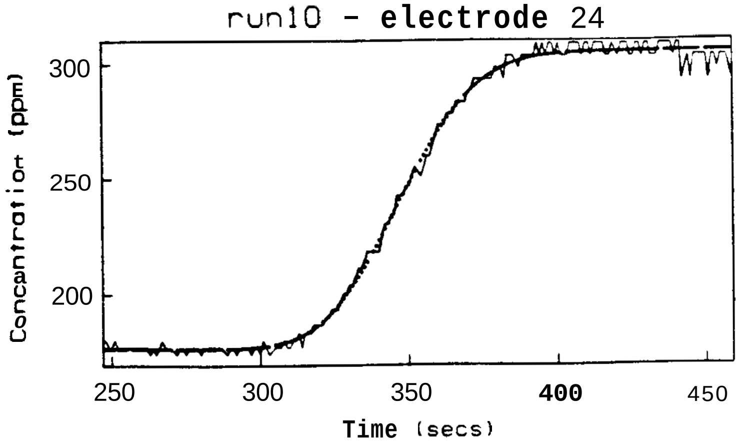

run10 - electrode 23



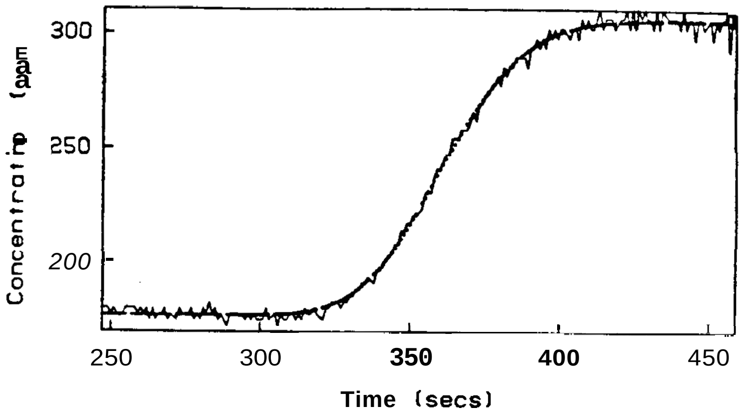

run10 - electrode 20



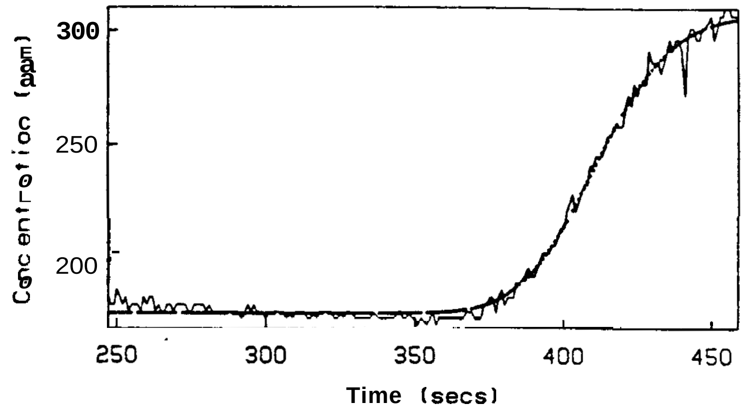

run10 - electrode 24



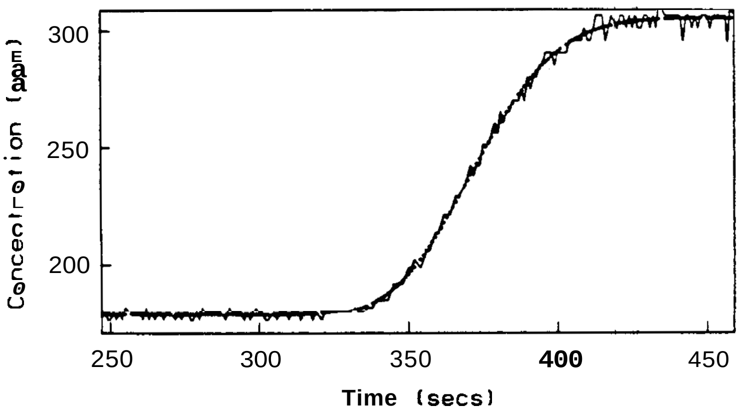
run10 - electrode 25



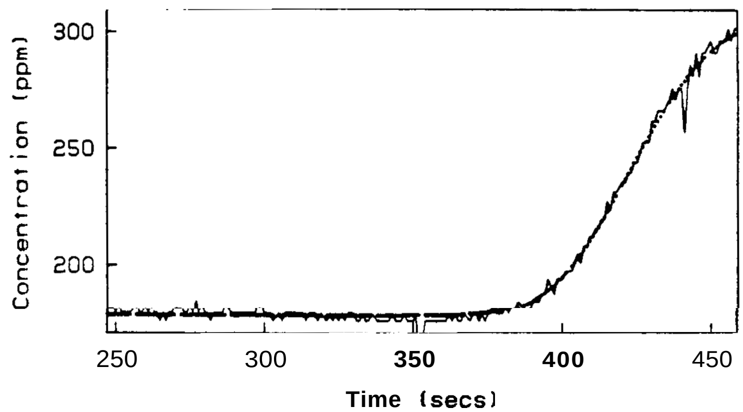

run10 - electrode 29



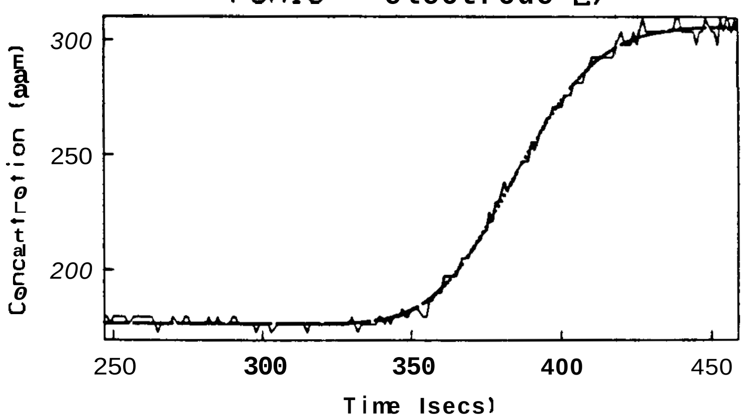

run10 - electrode 26



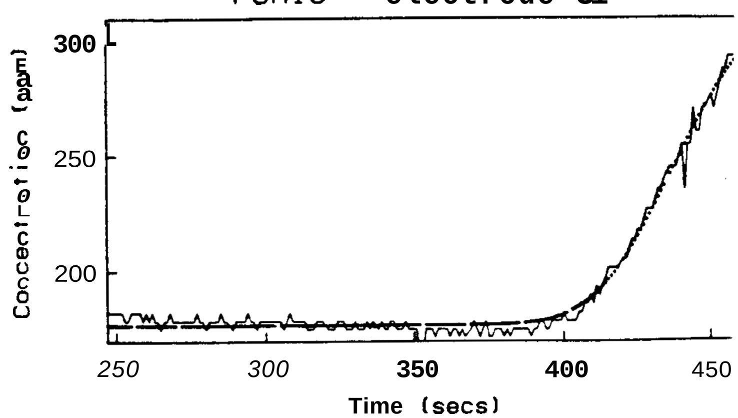

run10 - electrode 30



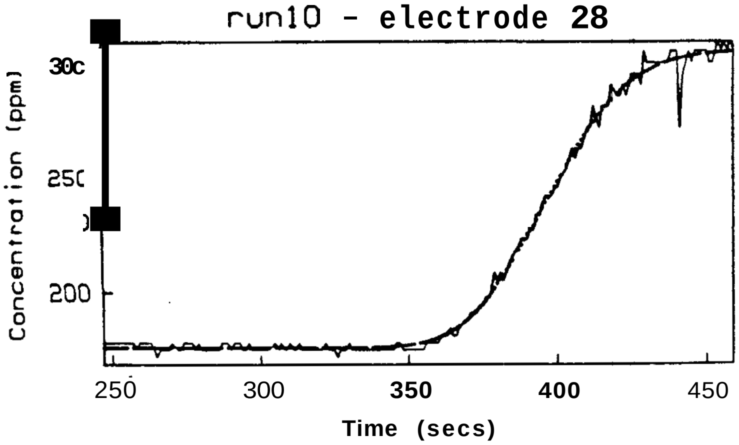

run10 - electrode 27



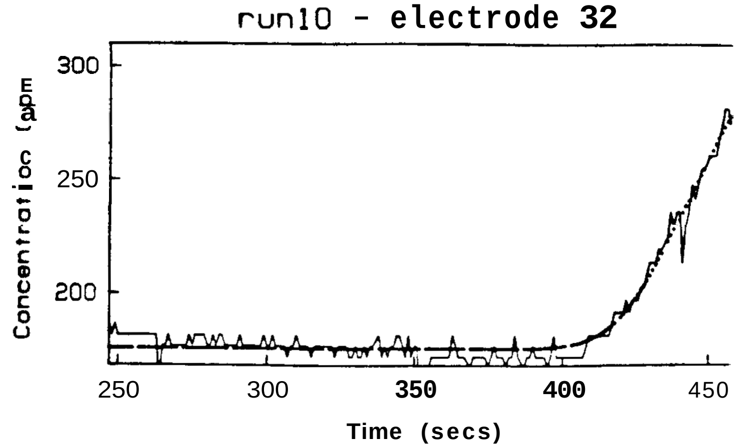

run10 - electrode 31




run10 - electrode 28




run10 - electrode 32



## Section 7: CONCLUSIONS

The approximation for Taylor Dispersion, Equation (3), is sufficiently accurate to be used in a fracture flow model for tracer test analysis.

The slight decline of dispersivity with distance suggests that further studies may find a stricter criterium for the non-dimensional time before tracer front equalization.

## Section 8: REFERENCES


1. Bear, Jacob. Dynamics of Fluids in Porous Media , American Elsevier, New York, NY. 1972.

2. Bear, Jacob. "Some Experiments in Dispersion", J. Geoph. Res. ,Volume 66, no. 8. pp. 2455-2467.

3. Carslaw, H.S. & Jaeger, J.C. Conduction of Feat in Solids , Oxford University Press, 2nd Ed., 1959.

4. Coats, K.H. "Dead End Pore Volume and Dispersion in Porous Media", Soc. Pet. Eng. Jour. , March 1964.

5. Davies, Cecil W. The Conductivity of Solutions , John Wiley & Sons, Inc., New York. 2nd Ed. 1933.

6. Fossum, M.P. Tracer Analysis in a Fractured Geothermal Reservoir , Masters Report, Stanford University, June 1982.

7. Fossum, M.P. and Horne, R.N. "Interpretation of Tracer Return Profiles at Wairakei Geothermal Field Using Fracture Analysis", Geoth. Res. Counc., Transaction 6 , 1982.

8. Horne, R.N. "Reservoir Engineering Aspects of Reinjection", Japan Geoth. Energy Assoc. Jour. , 19 , 23-30, (1982), (in Japanese).

9. Horne, R.N. and Rodriguez, F. "Dispersion in Tracer Flow in Fractured Geothermal Systems", Geoph. Res. Let. , Vol. 10, no. 4, 289-292, 1983.

10. Hull, L.C. and Koslow, K.N. "Dispersion in Fracture Networks", Proc. , Eighth Wkshp Geoth. Res. Engr., Stanford University, Dec. 1982.

11. Taylor, Sir Geoffrey, F.R.S. "Dispersion of Soluble Matter Flowing Slowly Through a Tube", Proc. Roy. Soc. , A 219, pp 186-203, 1953.

12. VARPRO, Computer Science Dept., Stanford University.

APPENDIX **A: MULTIPLEXER** BOARD

The multiplexer board contains circuitry which switches through the electrode array, allowing instantaneous voltage measurement at each location. Its design is described by the flow dagram of Fig. A.I.I.

The circuitry is capable of testing 128 electrodes, divided into 8 groups of 16. The computer selects a bank of 16 electrodes and the voltage across each is measured individually using a resistance bridge. The signal is amplified, converted to digital logic, and stored in OUTPUT.DAT for later processing.

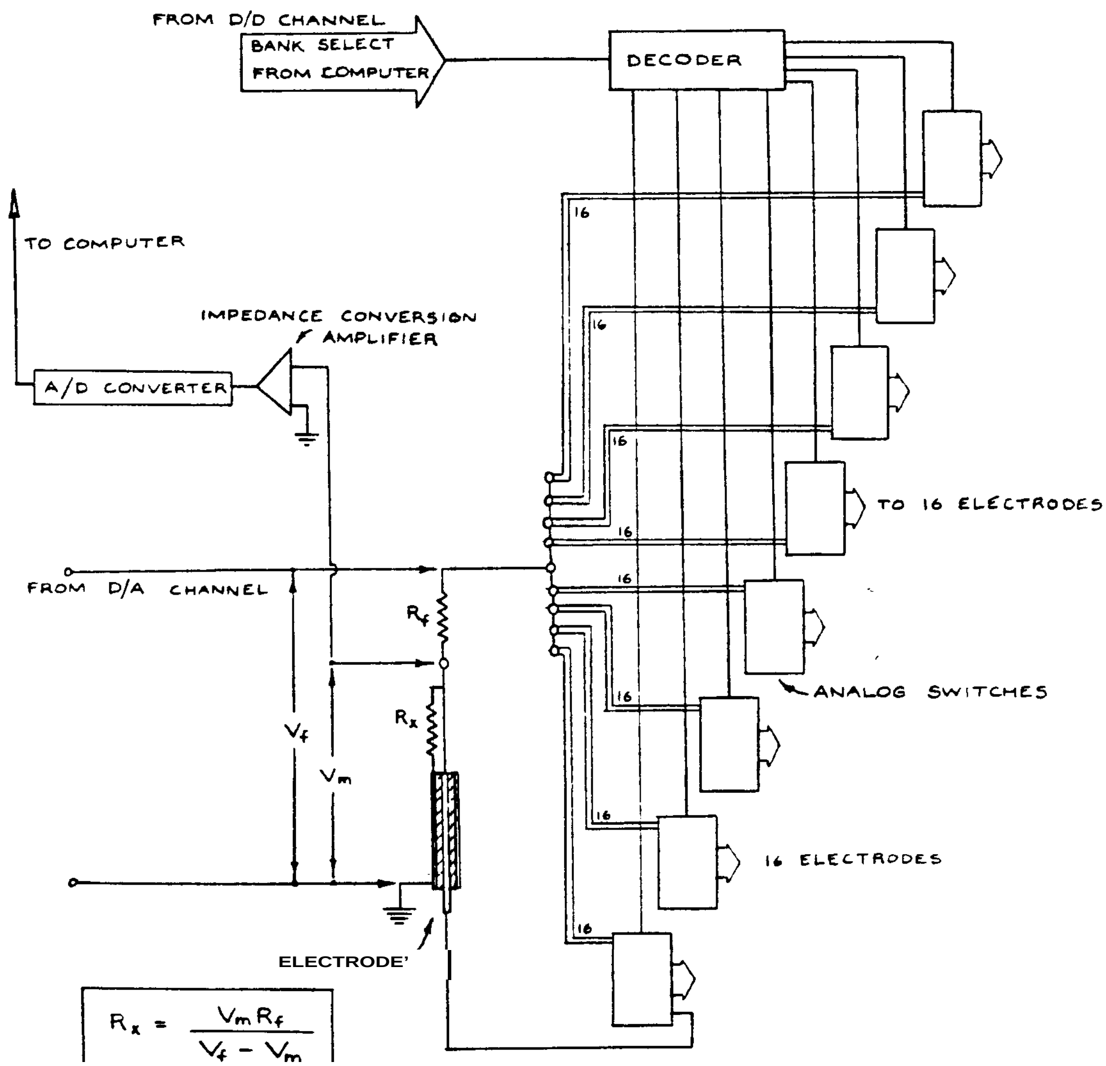



Fig. A. 1.1. Multiplexer Board Design

APPENDIX **B: COMPUTER SCANNING ALGORITHMS**

Figure B.1.1 shows a flow diagram describing the computer scanning algorithms used for data aquisition, transmission, curve fitting and plotting.

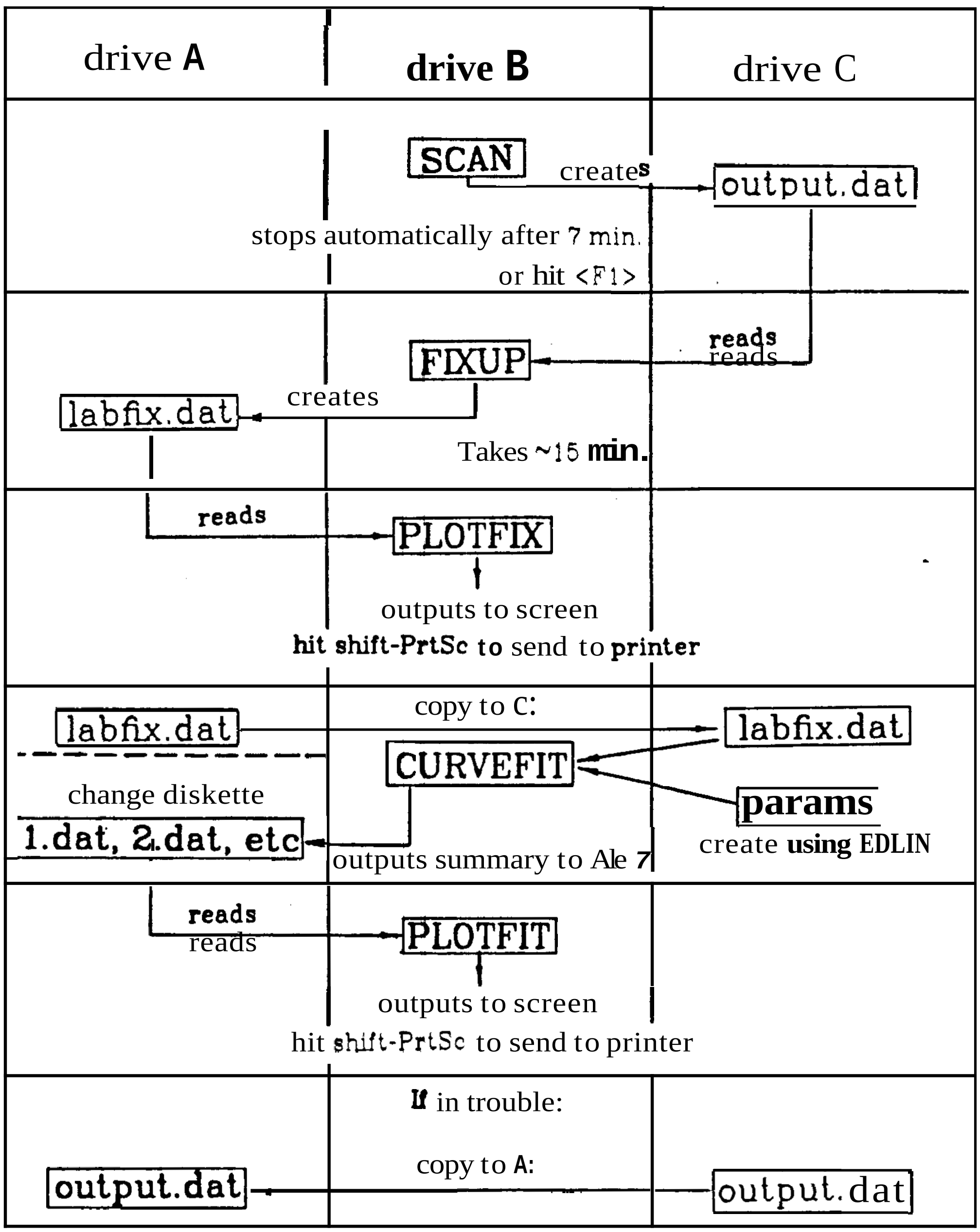



Fig. B. 1.1. Computer Scanning Algorithms

## Appendix C: COMPUTER PROGRAMS AND SUBROUTINES

### Directory of Lab Control Diskette

```
 Volume in drive B has no label
 Directory of  B:\

SCAN      BAS     3153   5-10-84  10:26a
INIT      BAS     4779   5-10-84  10:15a
CALTABLE  DAT      128   5-29-84  11:01p
INIT      EXE    27611   5-10-84  10:18a
CURVEFIT  EXE   117428   6-05-84   3:10p
PLOTFIT   EXE    26362   5-30-84   5:02p
TEST      BAS      128   3-01-84  11:44a
FLOTLAB   EXE    24632   5-04-84  11:42a
CALVOLT   BAS      640   5-09-84   5:13p
INIT2     BAS     3200   5-09-84   1:36p
SCAN2     BAS     3200   5-09-84   5:36p
SCAN2     EXE    24889   5-09-84   5:39p
FIXUF'    EXE    47668   5-11-84   2:36p
SCAN      EX€    24953   5-11-84  10:39a
DDTEST    BAS      128   3-02-84   3:50p
DACAL     BAS      512   3-06-84   2:20p
FIXUP     FOR     1926   5-11-84   2:34p
PARAMS             101   5-31-84  11:21a
FLOTFIX   BAS     2040   5-10-84   5:23p
FLOTFIT   BAS     3224   5-30-84   5:01p
FLOTFIX   EXE    24632   5-10-84   5:24p
GO                  35   5-32-84   9:16a
CAL       DAT     3064   5-28-84   7:44p
       23 File(s)      4096 bytes free
```

***SCAN***

```
10    DEF FNRCONV(X)=X+6*INT(X/10)
20 HRS=0
30 MIN=0
40 SEC=0
50 OUT &H719,23
60 OUT &H718,3
70 OUT &H718,129
80 OUT &H719,1
90 OUT &H718,57
100 OUT &H718,31
110 OUT &H718,0
120 OUT &H718,FNRCONV(SEC)
130 OUT &H719,2
140 OUT &H718,57
150 OUT &H718,16
160 OUT &H718,FNRCONV(MIN)
170 OUT &H718,FNRCONV(HRS)
180 OUT &H719,71
190 OUT &H719,9
200 OUT &H718,0
210 OUT &H718,0
220 OUT &H719,10
230 OUT &H718,0
240 OUT &H718,0
250 OUT &H719,39
260 DIM ELECVOLT(100),YPOLD(100),XPOLD(100)
270 FOR NUM=1 TO 100
280 XPOLD(NUM)=(NUM-1)*19+15: YPOLD(NUM)=130
290 NEXT NUM
300 SIGMAX=2!
310 ON KEY(1) GOSUB 1090
320 KEY OFF
330 OPEN "c:output.dat" FOR OUTPUT AS #1
335 SCAN=0
340 SCREEN 2
350 CLS
360 PRINTCOL=1
370 TIME0$=TIME$
380 OLDTIME$=TIME$
390 THS=0
400 NUMCYCLE=0
410 ADHIGH=0
420 ADLOW=0
430 ADDRESS=&H710
440 OUT &H71F,&H80
450 OUT &H714,128
460 VOLTAGE=1
```

```
470 VOLTS=5
480 DECIMAL=204.7*VOLTS
490 DECIMAL=INT(DECIMAL) : DAHIGH=INT(DECIMAL/256): DALOW=DECIMAL-256*DAHIGH
500 FOR CYCLE=1 TO 2
510 REM alternate voltage
520 REM IF CYCLE=1 THEN DAHIGH=&H7: DALOW=&HFF ELSE DAHIGH=&H8: DALOW=0
530 OUT &H711,DAHIGH
540 UUT &H710,DALOW
550 FOR CIRCUIT=1 TO 2
560 OUT &H71D,CIRCUIT
590 PRINTCOL=1+(CIRCUIT-1)*20
600 FOR ELECTRODE=0 TO 15
610 OUT &H715,ELECTRODE
620 OUT &H716,0
630 IF ELECTRODE=0 THEN GOTO 720
640 ADSIG=256*ADHIGH+ADLOW
650 IF ADSIG>32767 THEN ADSIG=ADSIG-65536!
660 ADVOLT=VOLTAGE*ADSIG/204.8
6'70 ELECNUM=(CIRCUIT-1)*16+ELECTRODE
680 ELECVOLT(ELECNUM)=ADVOLT
690 FRINT #1, USING "&_ ###_ ###_ ##.####";OLDTIME$,THS,ELECNUM,ADVOLT
700 REM LOCATE ELECTRODE,PRINTCOL
710 REM FRINT USING "&_ ###_ ###_ ##.####";OLDTIME$,THS,(CIRCUIT-1)*16+ELECTROD
:ADVOLT
720 IF INF(&H714)<128 THEN GOTO 720
730 ADLOW=INP(&H715)
740 ADHIGH=INP(&H716)
750 NEXT ELECTRODE
760 ADSIG=256*ADHIGH+ADLOW
770 IF ADSIG>32767 THEN ADSIG=ADSIG-65536!
780 ADVOLT=VOLTAGE*ADSIG/204.8
790 ELECNUM=(CIRCUIT-1)*16+16
800 ELECVOLT'(ELECNUM)=ADVOLT
810 FRINT #1, USING "&_ ###_ ###_ ##.####";OLDTIME$,THS,ELECNUM,ADVOLT
820 OLDTIME$=TIME$
830 OUT &H719,163
840 OUT &H713,17
850 THS=INF(&H718)
860 THS=THS-6*INT(THS/16)
870 NEXT CIRCUIT
875 OUT &H71D,0
880 NUMCYCLE=NUMCYCLE+1
890 OLDTIME$=TIME$
900 KEY(1) ON
910 MIN=1:MAX=32:YBASE=130
920 FOR NUM=1 TO 32
930 SIGNAL=ELECVOLT(NUM)
940 YP=YBASE-60*SIGNAL/SIGMAX
950 XP=(NUM-MIN)*19+15
960 IF NUM<MAX THEN LINE(XPOLD(NUM),YPOLD(NUM))-(XPOLD(NUM+1),YPOLD(NUM+1)),0
 LINE  (XP,YP)-(XPOLD(NUM+1),YPOLD(NUM+1)),1
970 IF NUM>MIN THEN LINE (XPOLD(NUM),YPOLD(NUM))-(XPOLD(NUM-1),YPOLD(NUM-1)),0:
 LINE (XP,YP) - (XPOLD(NUM-1),YPOLD(NUM-1)),1
```

```
980 XPOLD(NUM)=XP:YPOLD(NUM)=YP
990 REM IF NUM<17 THEN LOCATE NUM,1 ELSE LOCATE NUM-16,40
1000 REM PRINT NUM-1,SIGNAL
1010 NEXT NUM
1020 LINE(15,YBASE-60)-(624,YBASE-60)
1030 LINE (15,YBASE)-(624,YBASE)
1040 LINE (15,YBASE)-(15,YBASE-60)
1050 LINE (624,YBASE)-(624,YBASE-60)
1055 SCAN=SCAN+1 : IF SCAN>440 THEN GOTO 1090
1060 KEY(1) STOP
1070 NEXT CYCLE
1080 GOTO 500
1090 KEY(1) STOP
1100 CLOSE 1
1110 PRINT "start time",TIME0$,"finish time",OLDTIME$,"No. of cycles",NUMCYCL
1120 STOP
```

**FIXUP**

```
      COMMON /A/VOLT(32,500)
      INTEGER ELEC,ELECN,HR,MIN,SEC,THS,RECNO
      DIMENSION TIME(500),CTBL(20),VTBL(20,32)
      CALBRT=0
      IF (CALBRT.EQ.0) GO TO 200
C  Next section skipped if no calibration desired
      OPEN (3,FILE='cal.dat',STATUS='OLD',ACCESS='SEQUENTIAL')
      READ (3,103) NPTS
 103  FORMAT(I2)
      DO 10 ELEC=1,32
      DO 11 N=1,NPTS
      READ(3,102) NDUMMY,VTBL(N,ELEC)
 102  FORMAT (I4,11X,F6.4)
      CTBL(N)=NDUMMY
 11   CONTINUE
 10   CONTINUE
      END FILE 3
 200  OPEN (1,FILE='c:output.dat',status='old',access='direct',
     & recl=24,form='formatted')
      READ(1,100,REC=1) HR,MIN,SEC,THS,ELEC,SIG
 100  FORMAT (I2,1X,I2,1X,I2,2X,I2,2X,I2,2X,F6.4)
```

```
      TZERO=3600.*HR+60.*MIN+SEC
      DO 1 ELEC=1,32
      RECNO=ELEC
      1=1
 3    READ (1,100,REC=RECNO,END=2) HR,MIN,SEC,THS,ELECN,SIG
      IF (ELEC.EQ.1) TIME(I)=3600.*HR+60.*MIN+SEC-TZERO
      IF (CALBRT.NE.O)CONCN=CONVRT(SIG,NFTS,ELEC,CTBL,VTBL(1,ELEC))
      VOLT(ELEC,I)=SIG
C     write (*,'(f7.1,1x,f10.4)') time(i),concn
      I=I+1
      RECNO=RECNO+32
      GO TO 3
 2    CONTINUE
      WRITE (*,'(I4)') ELEC
 1    CONTINUE
      END FILE 1
      OPEN (2,FILE='a:labfix.dat',status='new',access='sequential')
      DO 4 ELEC=1,32
      DO 5 II=1,I
      WRITE (2,101) TIME(II),VOLT(ELEC,II)
 101  FORMAT (F7.1,1X,F10.4)
 5    CONTINUE
 4    CONTINUE
      END FILE 2
      STOP
      END
      FUNCTION CONVRT(SIG,NPTS,ELEC,CTBL,V)
      DIMENSION CTBL(20),V(20)
      NFTS1=NPTS-1
      IF (SIG.GT.V(1)) GO TO 1
      DO 2 I=1,NPTS1
      IF (SIG.GT.V(I+1)) GO TO 3
 2    CONT1NUE
      SLOPE=1./(V(NPTS)-V(NPTS-1))/(CTBL(NPTS)-CTBL(NPTS-1))
      CONVRT=CTBL(NFTS)+SLOPE*(V(NPTS)-SIG)
      RETURN
 3    SLOPE=(CTBL(I+1)-CTBL(I))/(V(I+1)-V(I))
      CONVRT=CTBL(I+1)+SLOPE*(SIG-V(I+1))
      RETURN
 1    SLOPE=(CTBL(2)-CTBL(1))/(V(2)-V(1))
      CONVRT=CTBL(1)+SLOPE*(SIG-V(1))
      RETURN
      END
```

**PLOTFIT**

```
10 DIM X(500),Y(500)
11 DIM FILE$(32)
12 DATA "1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16"
17","18","19","20","21","22","23","24","25","26","27","28","29","30","31","32"
13 FOR ELEC%=1 TO 32
14 READ FILE$(ELEC%)
15 NEXT ELEC%
20 INPUT "Experiment number";NUMEXP$
30 INPUT "Range of electrodes";MINELEC%,MAXELEC%
40 FOR ELEC%=MINELEC% TO MAXELEC%
50 FILEN$=FILE$(ELEC%)+".dat"
60 TITLE$="Run - "+NUMEXP$+" ; Electrode "+FILE$(ELEC%)
70 OPEN FILEN$ FOR INPUT AS 1
80 INPUT #1,NUMPTS
90 XLOG%=0
100 YLOG%=0
110 PT%=0
120 FOR N=1 TO NUMPTS
130 IF EOF(1) THEN GOTO 180
140 INPUT #1,X(N),Y(N)
150 IF XLOG% THEN X(N)=LOG(X(N))/LOG(10!)
160 IF YLOG% THEN Y(N)=LOG(Y(N))/LOG(10!)
170 NEXT N
180 PRINT "number of data points is ",N
190 N1=N-1
200 YMIN=150!
210 YMAX=320!
220 XMAX=X(N-5)
230 XMIN=X(1)
240 IF XLOG% THEN XMAXA=10!^XMAX: XMINA=10!^XMIN ELSE XMAXA=XMAX: XMINA=XMIN
250 IF YLOG% THEN YMAXA=10!^YMAX: YMINA=10!^YMIN ELSE YMAXA=YMAX: YMINA=YMIN
260 PRINT "minimum & maximum x value",XMINA,XMAXA
270 PRINT "minimum & maximum y value",YMINA,YMAXA
280 XSCRA=9!
290 YSCRA=6!
300 XSCR=XSCRA-.5
310 YSCR=YSCRA-.5
320 PRINT "screen size is",XSCR,"wide, by",YSCR,"high"
330 XDIV=50!
340 YDIV=50!
350 IF XLOG% THEN XDIV=LOG(XDIV)/LOG(10!)
360 IF YLOG% THEN YDIV=LOG(YDIV)/LOG(10!)
370 DEF FNIX(P)=INT((P/XSCRA)*639!)
380 DEF FNIY(P)=199-INT((P/YSCRA)*199!)
390 XWID=XMAX-XMIN
400 YWID=YMAX-YMIN
410 SCREEN 2,0,0
420 KEY OFF
470 CLS
480 KEY OFF
490 KPASS=1
510 LINE (FNIX(1!),FNIY(1!))-(FNIX(1!),FNIY(YSCR))
520 LINE (FNIX(1!),FNIY(1!))-(FNIX(XSCR),FNIY(1!))
```

```
525 LOCATE 1,20,0 : PRINT TITLE$
530 NX=INT(XWID/XDIV)
540 NY=INT(YWID/YDIV)
550 FOR I=0 TO NX
560 PSET (FNIX(1!+I*(XSCR-1!)*XDIV/XWID),FNIY(1!))
570 LINE -STEP(0,199-FNIY(.125))
580 NEXT
590 FOR I=0 TO NY
600 PSET (FNIX(1!),FNIY(1!+I*(YSCR-1!)*YDIV/YWID))
610 LINE -STEP(-FNIX(.125),0)
620 NEXT
630 FOR I=0 TO NX
640 XVAL=XMIN+I*XDIV
650 XPOS=((XVAL-XMIN)/XWID)*(XSCR-1!)+.85
660 IX=INT(80!*XPOS/XSCRA)+1
670 IY=25-INT(12.5/YSCRA)
680 IF IX>80 OR IX<1 OR IY>25 OR IY<1 GOTO 720
690 LOCATE IY,IX
700 IF XLOG% THEN XVAL=10!^XVAL
710 PRINT XVAL
720 NEXT
730  FOR I=0 TO NY
740 YVAL=YMIN+I*YDIV
750 YPOS=((YVAL-YMIN)/YWID)*(YSCR-1!)+1!
760 IY=25-INT(25*YPOS/YSCRA)
770 IX=INT(.4*80!/XSCRA)
780 IF YMAX>1000 THEN IX=IX-1
790 IF YMAX>10000 THEN IX=IX-1
800 IF IX>80 OR IX<1 OR IY>25 OR IY<1 GOTO 840
810 LOCATE IY,IX
820 IF YLOG% THEN YVAL=10!^YVAL
830 PRINT YVAL
840 NEXT
850 LOCATE 1,1
860 IF KPASS=0 THEN KPASS=1:  OUT &H3DD,&H20: GOTO 510
880 KPASS=0
890 XP=FNIX(1!+(XSCR-1!)*(X(1)-XMIN)/XWID): YP=FNIY(1!+(YSCR-1!)*(Y(1)-YMIN)/YWI
D)
900 IF PT% THEN CIRCLE (XP,YP),FNIX(.03): PAINT(XP,YP) ELSE PSET(XP,YP)
910 FOR I=2 TO N1
920 XP=FNIX(1!+(XSCR-1!)*(X(I)-XMIN)/XWID): YP=FNIY(1!+(YSCR-1!)*(Y(I)-YMIN)/YWI
D)
930 IF PT% THEN CIRCLE (XP,YP),FNIX(.03): PAINT (XP,YP) ELSE LINE -(XP,YP)
940 NEXT
950 IF KPASS>0 THEN GOTO 1060
960 KPASS=1
970 BEEP
980 PT%=1
1000 INPUT #1,NUMPTS
1010 FOR N=1 TO NUMPTS
1020 IF EOF(1) GOTO 890
1030 INPUT #1,X(N),Y(N)
1040 NEXT N
1050 GOTO 890
1060 BEEP
1070 CLOSE 1
1080 NEXT ELEC%
1090 SCREEN 0,0,0
1100 SYSTEM
```

**PLOTFIX**

```
20 DIM X(1000) ,Y(1000)
25 INFUT "ymin,ymax,ydiv";YMIN,YMAX,YDIV
30 CHOICE=!:)
35 OFEN "a:labfix.dat" FOR INFUT AS 1
40 I=0
45 XOLD=-1!
50 IF EOF(1) THEN GOTO 120
60 INFUT #1,X(I),Y(I)
90 IF X(I)<XOLD THEN GOTO 120
95 XOLD=X(I)
96 I=I+1
100 IF EOF(1) THEN GOTO 120
110 GOTO 50
120 N=I-1
125 CHOICE=CHOICE+1
126 FT%=0
140 XMAX=X(N)
150 XMIN=0
160 XSCRA=9!
170 YSCRA=6!
180 XSCR=XSCRA-.5
190 YSCR=YSCRA-.5
200 XDIV=60
220 IF XLOG% THEN XDIV=LOG(XDIV)/LOG(10!)
230 IF YLOG% THEN YDIV=LOG(YDIV)/LOG(10!)
240 DEF FNIX(F)=INT((F/XSCRA)*639!)
250 DEF FNIY(F)=199-INT((F/YSCRA)*199!)
260 XWID=XMAX-XMIN
270 YWID=YMAX-YMIN
280 SCREEN 2,0,0
290 KEY OFF
300 CLS
310 CLS
320 KEY OFF
325 LOCATE 1,40,0 : FRINT "electrode",CHOICE
330 KFASS=1
340 LINE (FNIX(1!),FNIY(1!))-(FNIX(1!),FNIY(YSCR))
350 LINE (FNIX(1!),FNIY(1!))-(FNIX(XSCR),FNIY(1!))
360 NX=INT(XWID/XDIV)
370 NY=INT(YWID/YDIV)
380 FOR I=0 TO NX
390 FSET (FNIX(1!+I*(XSCR-1!)*XDIV/XWID),FNIY(1!))
400 LINE -STEF(0,199-FNIY(.125))
410 NEXT I
420 FOR I=0 TO NY
430 FSET (FNIX(1!),FNIY(1!+I*(YSCR-1!)*YDIV/YWID))
440 LINE -STEF(-FNIX(.125),0)
450 NEXT I
```

```
460 FOR I=0 TO NX
470 XVAL=XMIN+I*XDIV
480 XPOS=((XVAL-XMIN)/XWID)*(XSCR-1!)+.85
490 IX=INT(80!*XPOS/XSCRA)+1
500 IY=25-INT(12.5/YSCRA)
510 IF IX>80 OR IX<1 OR IY>25 OR IY<1 GOTO 550
520 LOCATE IY,IX
540 PRINT XVAL
550 NEXT I
560  FOR I=0 TO NY
570 YVAL=YMIN+I*YDIV
580 YPOS=((YVAL-YMIN)/YWID)*(YSCR-1!)+1!
590 IY=25-INT(25*YPOS/YSCRA)
600 IX=INT(.6*80!/XSCRA)-2
610 IF IX>80 OR IX<1 OR IY>25 OR IY<1 GOTO 650
620 LOCATE IY,IX
640 PRINT YVAL
650 NEXT I
660 LOCATE 1,1
670 XP=FNIX(1!+(XSCR-1!)*(X(1)-XMIN)/XWID): YP=FNIY(1!+(YSCR-1!)*(Y(1)-YMIN)/YW
D)
680 IF PT% THEN CIRCLE (XP,YP),FNIX(.03): PAINT(XP,YP) ELSE PSET(XP,YP)
690 FOR I=2 TO N
700 XP=FNIX(1!+(XSCR-1!)*(X(I)-XMIN)/XWID): YP=FNIY(1!+(YSCR-1!)*(Y(I)-YMIN)/YW
D)
710 IF PT% THEN CIRCLE (XP,YP),FNIX(.03): PAINT (XP,YP) ELSE LINE -(XP,YP)
720 NEXT I
730 BEEP
735 IF CHOICE < 32 THEN GOTO 40
3000 CLOSE 1: STOP
```

```
C ******************************************************************
C
C                 PROGRAM BEGINS -Gilardi experiment analysis
C
C ******************************************************************
C ******************************************************************
C
C
      IMPLICIT REAL*8(A-B,D-H,O-Z)
C
C
C      SET DIMENSIONS FOR VARFRO.  BE CAREFUL WHEN SETTING THE
C      DIMENSIONS FOR THE INCIDENCE MATRIX INC.  SEE NOTE.
C
      DIMENSION Y(500),T(500),ALF(3),BETA(2),W(500),A(500,7),
     *INC(12,8),NELTRY(32),NGO(32)
      COMMON XPOS,TZERO
      COMMON /DUMMY/C(500,6)
      EXTERNAL ADA
      CHARACTER*6 CFILE(32)
      PATA CFILE/'1.dat','2.dat','3.dat','4.dat','5.dat',
     @ '6.dat','7.dat','8.dat','9.dat',
     1 '10.dat','11.dat','12.dat','13.dat','14.dat','15.dat',
     @ '16.dat','17.dat','18.dat','19.dat',
     2 '20.dat','21.dat','22.dat','23.dat','24.dat','25.dat',
     @ '26.dat','27.dat','28.dat','29.dat',
     3 '30.dat','31.dat','32.dat'/
C
C         SET PARAMETERS FOR VARPRO.
C
C
       IPLOT=0
      NMAX = 500
      IPRINT=50
C
C
C         READ DATA SEQUENTIAL ORDERING AND
C         PROFER FORMATTING ARE IMPORTANT.
C
C
C
C       NL IS THE NUMBER OF NONLINEAR FARAMETERS
C
C
      READ (5,311) NL
 311  FORMAT (I1)
      WRITE(6,12) NL
 12   FORMAT (1H0,10X,'NUMBER OF NONLINEAR FARAMETERS'//(I3))
C
C
C       L IS THE NUMBER OF LINEAR FARAMETERS
C
C
      L=2
C
```

```
C
C        ESTIMATES OF THE NONLINEAR FAHAMETERS
C
C
      POSO=9.8425
      READ (5,310) VELOC,DIFFS,TZERO,DCO,DC1
 310  FORMAT (F10.4)
      READ (5,312) NTRIES, (NELTRY(K),K=1,NTRIES)
 312  FORMAT (33I2)
      DO 314 K=1,32
 314  NGO(K)=1.
      DO 313 K=1,NTRIES
 313  NGO(NELTRY(K))=0.
      WRITE (6,20) VELOC,DIFFS,TZERO
 20   FORMAT (/,'0 Mean Velocity(cm/sec)   Diffusivity (cm2/sec)·
     # '  Time zero (sec)',/,(5X,F9.5,18X,F13.3,10X,F10.3))
C
C
      LPP2=L+NL+2
C
C
C
C        N IS THE NUMBER OF OBSERVATIONS
C        IV IS THE NUMBER OF INDEFENDENT VARIABLES T
C
C
      IV=1
C
C
C
C
C     T IS THE INDEFENDENT VARIABLE
C     Y IS THE N-VECTOR OF OBSERVATIONS
C
C
C
      WRITE (7,323)
 323  FORMAT(2X,'Elec       Co          Cinj        Velocity   ',
     1 '      Disp         To',/,'                    ppm          ppm     ',
     2 '   cm/sec       cm2/sec        sec')
      DO 200 NELEC=1,32
      N=0
      TOLD=0.
 203  N=N+1
      READ (4,301,END=202) T(N),Y(N)
 301  FORMAT (2F10.4)
      IF (TOLD.GT.T(N)) GO TO 202
      TOLD=T(N)
      GO TO 203
 202  N=N-5
      IF (NGO(NELEC).GT.0) GO TO 200
      WRITE (6,320) NELEC
 320  FORMAT (//,2X,'Electrode number ',I3,'*****************')
      XPOS=POSO+5.08*(NELEC-1)
      ALF(1)=VELOC
      ALF(2)=DIFFS
      ALF(3)=TZERO
```

```
C
C       W(I) ARE THE WEIGHTING PARAMETERS
C
C
          DO 1 I=1,N
 1         W(I)=1.0
C
           IMINP=IDINT(TZERO+XPOS/VELOC)-100
           IMIN=IDINT(TZERO)
           IF (IMIN.LT.IMINP) IMIN=IMINP
        IF (IMIN.GT.N-200) IMIN=N-200
           N=N-IMIN
           IF (N.GT.200) N=200
           DO 201 1=1,N
           T(I)=T(IMIN+I)
           Y(I)=Y(IMIN+I)
C          WRITE (6,*) T(I),Y(I)
 201       CONTINUE
           CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR,10)
C
        V0=BETA(1)
        V1=BETA(1)+2.*BETA(2)
        BPAR=(V1-V0)/DLOG(DC0/DC1)
        APAR=V1+BPAR*DLOG(DC1)
        DO 211 I=1,N
 211    Y(I)=DEXP((APAR-Y(I))/BPAR)
           CALL VARPRO(L,NL,N,NMAX,LPP2,IV,T,Y,W,ADA,A,
     *IPRINT,ALF,BETA,IERR,100)
C
        LP1=L+1
        CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,1)
        DO 8 I=1,N
        C(I,LP1)=0.
        DO 9 J=1,L
        C(I,J)=BETA(J)*A(I,J)
 9      C(I,LP1)=C(I,LP1)+C(I,J)
 8      CONTINUE
        OPEN (1,FILE=CFILE(NELEC),STATUS='NEW',ACCESS='SEQUENTIAL')
        WRITE (1,305) N
        DO 205 I=1,N
 205    WRITE (1,14) T(I),Y(I)
        WRITE (1,305) N
        DO 206 I=1,N
 206    WRITE (1,14) T(I),C(I,LP1)
 14     FORMAT (1X,8F10.4)
 305    FORMAT (I6)
C
        END FILE 1
        DCMID=BETA(1)+BETA(2)
        DO 400 I1=1,N
        IF (C(II,LP1).GT.SNGL(DCMID)) GO TO 401
        TMID=T(II)
 400    CONTINUE
 401    TOCALC=TMID-XPOS/ALF(1)
        WRITE (7,302) NELEC,BETA(1),BETA(1)+2.*BETA(2),ALF(1),ALF(2)
     1  ,TOCALC
 302    FORMAT (2X,I4,5F12.4)
 200    CONTINUE
         STOP
         END
```

```
C ****************************************************************
C ****************************************************************
C
C                    SUBROUTINES
C
C ****************************************************************
C ****************************************************************
C
C
C
C
      SUBROUTINE ADA (LP,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,ISEL)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION ALF(NL),A(NMAX,LPP2),T(NMAX),INC(12,8)
      COMMON XPOS,TZERO
C
      PI=4.*ATAN(1.)
      TPI=2./DSQRT(PI)
      L=LP-1
      IF (NL.LT.3) ALF(3)=TZERO
C
C
C      THE INCIDENCE MATRIX INC(NL,L+1) IS FORMED BY SETTING
C      INC(K,J)=1 IF THE NONLINEAR PARAMETER ALF(K) APPEARS
C      IN THE J-TH FUNCTION PHI(J). (THE PROGRAM SETS ALL OTHER
C      INC(K,J) TO ZERO.)
C
C
      IF(ISEL.EQ.2) GO TO 90
      IF(ISEL.EQ.3)  GO TO 165
C
      INC(1,1)=0
      INC(2,1)=0
      INC(1,2)=1
      INC(2,2)=1
      IF (NL.EQ.3) INC(3,2)=1
C
C
C
C      THE VECTOR-SAMPLED FUNCTIONS PHI(J) ARE STORED IN
C      THE FIRST N ROWS AND FIRST L COLUMNS OF THE MATRIX
C      B(I,J). B(I,J) CONTAINS PHI(J,ALF;T(I),I,...N;
C      J=1,L. THE CONSTANT FUNCTIONS PHI WHICH DO NOT
C      DEPEND UPON ANY NONLINEAR PARAMETERS ALF MUST
C      APPEAR FIRST.
C
C
C
C
C
C
```

```
 90         DO 81 I=1,N
            IF (ISEL.EQ.1) A(I,1)=1.0
            A(I,2)=0.
            TTI=T(I)-ALF(3)
            IF (TTI.LE.0.) GO TO 81
            TFUNC=DSQRT(ALF(2)*TTI)
            A(I,2)=DERFC((XPOS-ALF(1)*TTI)/(2.*TFUNC))
      IF (XPOS*ALF(1)/ALF(2).GT.20.) A(I,2)=A(I,2)+DEXP(ALF
     1      (1)*XPOS/ALF(2))*DERFC((XPOS+ALF(1)*TTI)/(2.*TFUNC))
C
 81         CONTINUE
C
C
C
       IF (ISEL.EQ.2) GO TO 200
C
C
C
C
C
C
C
 165      DO 170 I=1,N
C
          A(I,4)=0.
          A(I,5)=0.
          A(I,6)=0.
          TTI=T(I)-ALF(3)
          IF (TTI.LE.0.) GO TO 170
          E1=DEXP(-((XPOS-ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E2=DEXP(-((XPOS+ALF(1)*TTI)**2)/(4.*ALF(2)*TTI))
          E3=0.
         IF (ALF(1)*XPOS/ALF(2).LT.20.) E3=DEXP(ALF(1)*XPOS/ALF(2))
          ERF1=DERFC((XPOS+ALF(1)*TTI)/(2.*DSQRT(ALF(2)*TTI)))
          FUNC=2.*DSQRT(TTI*ALF(2)**3)
          A(I,4)=-TPI*E1*(-TTI/(2.*DSQRT(ALF(2)*TTI)))
          IF (E3.GT.0.) A(I,4)=A(I,4)+(XPOS/ALF
     1       (2))*E3*ERF1+E3*(-TPI*E2)*TTI/(2.*DSQRT(ALF(2)*TTI))
          A(I,5)=TPI*E1*(XPOS-ALF(1)*TTI)/FUNC
          IF (E3.GT.0.) A(I,5)=A(I,5)-E3*ERF1*ALF(1)*XPOS/
     1     ALF(2)**2+E3*TPI*E2*(XPOS+ALF(1)*TTI)/FUNC
          IF (NL.LT.3) GO TO 170
          A(I,6)=-TPI*E1*(XPOS/(2.*DSQRT(ALF(2)*TTI**3))+ALF(1)/
     1       FUNC)
         IF (E3.GT.0.) A(I,6)=A(I,6)+E3*(-TPI*E2)
     2       *(XPOS/(2.*DSQRT(ALF(1)*TTI**3))-ALF(1)/FUNC)
C
C
 170              CONTINUE
```

```
      FUNCTION DERFC(Y)
      DOUBLE PRECISION    DERFC,Y
      DIMENSION           P(5),Q(4),P1(9),Q1(8),P2(6),Q2(5)
      DOUBLE FRECISION    P,Q,P1,Q1,P2,Q2,XMIN,XLARGE,SQRPI,X,
     *                    RES,XSQ,XNUM,XDEN,XI,XBIG
      INTEGER             ISW,I
      DATA                P(1)/113.8641541510502D0/,
     *                    P(2)/377.4852376853020D0/,
     *                    P(3)/3209.377589138469D0/,
     *                    P(4)/.1857777061846032D0/,
     *                    P(5)/3.161123743870566D0/
      DATA                Q(1)/244.0246379344442D0/,
     *                    Q(2)/1282.616526077372D0/,
     *                    Q(3)/2844.236833439171D0/,
     *                    Q(4)/23.60129095234412D0/
      DATA                P1(1)/8.883149794388376D0/,
     *                    P1(2)/66.11919063714163D0/,
     *                    P1(3)/298.6351381974001D0/,
     *                    P1(4)/881.9522212417691D0/,
     *                    P1(5)/1712.047612634071D0/,
     *                    P1(6)/2051.078377826071D0/,
     *                    P1(7)/1230.339354797997D0/,
     *                    P1(8)/2.153115354744038D-8/,
     *                    P1(9)/.5641884969886701D0/
      DATA                Q1(1)/117.6939508913125D0/,
     *                    Q1(2)/537.1811018620099D0/,
     *                    Q1(3)/1621.389574566690D0/,
     *                    Q1(4)/3290.799235733460D0/,
     *                    Q1(5)/4362.619090143247D0/,
     *                    Q1(6)/3439.367674143722D0/,
     *                    Q1(7)/1230.339354803749D0/,
     *                    Q1(8)/15.74492611070983D0/
      DATA                P2(1)/-3.603448999498044D-01/,
     *                    P2(2)/-1.257817261112292D-01/,
     *                    P2(3)/-1.608378514874228D-02/,
     *                    P2(4)/-6.587491615298378D-04/,
     *                    P2(5)/-1.631538713730210D-02/,
     *                    P2(6)/-3.053266349612323D-01/
      DATA                Q2(1)/1.872952849923460D0/,
     *                    Q2(2)/5.279051029514284D-01/,
     *                    Q2(3)/6.051834131244132D-02/,
     *                    Q2(4)/2.335204976268692D-03/,
     *                    Q2(5)/2.568520192289822D0/
      DATA                XMIN/1.0D-10/,XLARGE/6.375D0/
      DATA                XBIG/13.3D0/
      DATA                SQRPI/.5641895835477563D0/
```

```
      X = Y
      ISW = 1
      IF (X.GE.0.0D0) GO TO 5
      ISW = -1
      X = -X
    5 IF (X.LT..477D0) GO TO 10
      IF (X.LE.4.0D0) GO TO 30
      IF (ISW .GT. 0) GO TO 40
      IF (X.LT.XLARGE) GO TO 45
      RES = 2.0D0
      GO TO 70
   10 IF (X.LT.XMIN) GO TO 20
      XSQ = X*X
      XNUM = P(4)*XSQ+P(5)
      XDEN = XSQ+Q(4)
      DO 15 I = 1,3
         XNUM = XNUM*XSQ+P(I)
         XUEN = XDEN*XSQ+Q(I)
   15 CONTINUE
      RES = X*XNUM/XDEN
      GO TO 25
   20 RES = X*P(3)/Q(3)
   25 IF (ISW.EQ.-1) RES = -RES
      RES = 1.0D0-RES
      GO TO 70
   30 XSQ = X*X
      XNUM = P1(8)*X+P1(9)
      XDEN = X+Q1(8)
      DO 35 I=1,7
         XNUM = XNUM*X+P1(I)
         XDEN = XDEN*X+Q1(I)
   35 CUNTINUE
      RES = XNUM/XDEN
      GO TO 60
   40 IF (X.GT.XBIG) GO TO 65
   45 XSQ = X*X
      XI = 1.0D0/XSQ
      XNUM= P2(5)*XI+P2(6)
      XDEN = XI+Q2(5)
      DO 50 I = 1,4
         XNUM = XNUM*XI+P2(I)
         XDEN = XDEN*XI+Q2(I)
   50 CONTINUE
      RES = (SQRPI+XI*XNUM/XDEN)/X
   60 RES = RES*DEXP(-XSQ)
      IF (ISW.EQ.-1) RES = 2.0D0-RES
      GO TO 70
   65 RES = 0.0D0
   70 DERFC = RES
      RETURN
      END
```

```
      SUBROUTINE VARPRO (L, NL, N, NMAX, LPP2, IV, T, Y, W, ADA, A,
     X IPRINT, ALF, BETA, IERR, ITMAX)
      DOUBLE FRECISION A(NMAX, LPP2), BETA(L), ALF(NL), T(NMAX, IV),
     2 W(N), Y(N), ACUM,EPS1, GNSTEP, NU, FRJRES, R, RNEW, XNORM,
     2 AS,BS,S
      INTEGER B1, OUTPUT
      LOGICAL SKIF
      EXTERNAL ADA
      DATA EPS1 /1.D-6/, OUTFUT /6/
      IEHR = 1
      ITER = 0
      LP1 = L + 1
      B1 = L + 2
      LNL2 = L + NL + 2
      NLFI = NL + 1
      SKIP = .FALSE.
      MODIT = IFRINT
      IF (IPRINT .LE. 0) MODIT = ITMAX + 2
      NU = 0.
      NU = 1.
    5 CALL DF'A (L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF, ADA, IERR,
     X IFRINT, A, BETA, A(1, LP1), R)
      GNSTEF = 1.0
      ITERIN = 0
      IF (ITER .GT. 0) GO TO 10
         IF (NL .EQ. 0) GO TO 90
         IF (IERH .NE. 1) GO TO 99
         IF (IPRINT .LE. 0) GO TO 10
         WRITE (OUTPUT, 207) ITERIN, R
         WRITE (OUTFUT, 200) NU
   10 CALL ORFAC1(NLP1, NMAX, N, L, IFRINT, A(1, B1), FRJRES, IERR)
      IF (IERR .LT. 0) GO TO 99
      IERR = 2
      IF (NU .EQ. 0.) GO TO 30
   25    CALL ORFAC2(NLP1, NMAX, NU, A(1, B1))
   30    CALL BACSUE (NMAX, NL, A(1, EI), A(1, LNL2))
         DO 35 K = 1, NL
   35       A(K, B1)  = ALF(K) + A(K, LNL2)
   40    CALL DPA (L, NL, N, NMAX, LPP2, IV, T, Y, W, A(1, B1), ADA,
     X   IERR, IFRINT, A, BETA, A(1, LP1), RNEW)
         IF (IERR .NE. 2) GO TO 99
         ITER = ITER + 1
         ITERIN = ITERIN + 1
         SKIP = MOD(ITER, MODIT) .NE. 0
         IF (SKIP) GO TO 45
            WRITE (OUTPUT, 203) ITER
            WRITE (OUTPUT, 216) (A(K, B1), K = 1, NL)
            WRITE (OUTFUT, 207) ITEKIN, RNEW
   45    IF (ITER .LT. ITMAX) GO TO 50
            IERR = -1
            CALL VARERR (IPRINT, IERR, 1)
            GO TO 95
   50    IF (RNEW - R .LT. EPS1*(R + 1.DO)) GO TO 75
            IF (NU .NE. 0.) GO TO 60
            GNSTEP = 0.5*GNSTEP
            IF (GNSTEF' .LT. EPS1) GO TO 95
            DO 55 K = 1, NL
```

```
55            A(K, B1) = ALF(K) + GNSTEP*A(K, LNL2)
          GO TO 40
60        NU = 1.5*NU
          IF (.NOT. SKIP) WRITE (OUTPUT, 206) NU
          IF (NU .LE. 100.) GO TO 65
             IERR = -z
             CALL VARERR (IPRINT, IERR, 1)
             GO TO 95
65      DO 70 K = 1, NL
          KSUB = LP1 + K
          DO 70 J = K, NLP1
             JSUB = LP1 + J
             ISUB = NLP1 + J
70           A(K, JSUB) = A(ISUB, KSUB)
        GO TO 25
75 R = RNEW
   DO 80 K = 1, NL
80    ALF(K) = A(K, B1)
   ACUM = GNSTEP*XNORM(NL, A(1, LNL2))/XNORM(NL, ALF)
   IF (ITERIN .EQ. 1) NU = 0.5*NU
   IF (SKIP) GO TO 85
      WRITE (OUTPUT, 200) NU
      WRITE (OUTPUT, 208) ACUM
85 IERR = 3
   IF (PRJRES .GT. EPS1*(R + 1.D0)) GO TO 5
YO IERR = ITER
95 IF (NL .GT. 0) CALL DPA(L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF,
  X ADA, 4, IPRINT, A, BETA, A(1, LP1), R)
   CALL POSTPR(L, NL, N, NMAX, LNL2, EPS1, R, IPRINT, ALF, W, A,
  X A(1, LP1), BETA, IERR)
99 RETURN
200 FORMAT (9H     NU =, E15.7)
203 FORMAT (12H0  ITERATION, I4, 24H    NONLINEAR PARAMETERS)
206 FORMAT (25H      STEP RETRACTED, NU =, E15.7)
207 FORMAT (1H0, I5, 20H  NORM OF RESIDUAL =, E15.7)
208 FORMAT (34H     NORM(DELTA-ALF) / NORM(ALF) =, E12.3)
216 FORMAT (1H0, 7E15.7)
    END
C
      SUBROUTINE ORFAC1(NLP1, NMAX, N, L, IPRINT, B, PRJRES, IERR)
      DOUBLE PRECISION ACUM, ALPHA, B(NMAX, NLP1), BETA, DSIGN, PRJRES,
     X U, XNORM,AS,BS,S
C
      NL = NLP1 - 1
      NL23 = 2*NL + 3
      LP1 = L + 1
      DO 30 K = 1, NL
         LPK = L + K
         ALPHA = DSIGN(XNORM(N+1-LPK, B(LPK, K)), B(LPK, K))
         U = B(LPK, K) + ALPHA
         B(LPK, K) = U
         BETA = ALPHA * U
         IF (ALPHA .NE. 0.0) GO TO 13
         IERR = -8
         CALL VARERR (IPRINT, IERR, LP1 + K)
```

```
      GO TO 99
13    KP1 = K + 1
      DO 25 J = KP1, NLP1
         ACUM = 0.0
         DO 20 I = LPK, N
20          ACUM = ACUM + B(I, K) * B(I, J)
         ACUM = ACUM / BETA
         DO 25 I = LPK, N
25          B(I, J) = B(I, J) - B(I, K) * ACUM
30    B(LPK, K) = -ALPHA
   PRJRES = XNORM(NL, B(LP1, NLP1))
   IF (IERR .EQ. 4) GO TO 99
   DO 50 K = 1, NL
      LPK = L + K
      DO 40 J = K, NLP1
         JSUB = NLP1 + J
         B(K, J) = B(LPK, J)
40       B(JSUB, K) = B(LPK, J)
50    B(NL23, K) = XNORM(K, B(LP1, K))
99 RETURN
   END
C
   SUBROUTINE ORFAC2(NLP1, NMAX, NU, B)
   DOUBLE PRECISION ACUM, ALPHA, B(NMAX, NLP1), BETA, DSIGN, NU, U,
  X XNORM,AS,BS,S
   NL = NLP1 - 1
   NL2 = 2*NL
   NL23 = NL2 + 3
   DO 30 K = 1, NL
      KP1 = K + 1
      NLPK = NL + K
      NLPKM1 = NLPK - 1
      B(NLPK, K) = NU * B(NL23, K)
      B(NL, K) = B(K, K)
      ALPHA = DSIGN(XNORM(K+1, B(NL, K)), B(K, K))
      U = B(K, K) + ALPHA
      BETA = ALPHA * U
      B(K, K) = -ALPHA
      DO 30 J = KP1, NLP1
         B(NLPK, J) = 0.
         ACUM = U * B(K,J)
         DO 20 I = NLP1, NLPKM1
20          ACUM = ACUM + B(I,K) * B(I,J)
         ACUM = ACUM / BETA
         B(K,J) = B(K,J) - U * ACUM
         DO 30 I = NLP1, NLPK
30          B(I,J) = B(I,J) - B(I,K) * ACUM
   RETURN
   END
C
   SUBROUTINE DPFI (L, NL, N, NMAX, LPP2, IV, T, Y, W, ALF, ADA, ISEL,
  X IPRINT, A, U, R, RNORM)
   DOUBLE PRECISION A(NMAX, LPP2), ALF(NL), T(NMAX, IV), W(N), Y(N),
  X ACUM,ALPHA,BETA, RNORM, DSIGN, DSQRT, SAVE, R(N), U(L), XNORM,
  X AS,BS,S
   INTEGER FIRSTC, FIRSTR, INC(12, 8)
   LOGICAL NOWATE, PHILP1
   EXTERNAL ADA
```

```
      IF (ISEL .NE. 1) GO TO 3
         LP1 = L + 1
         LNL2 = L + 2 + NL
         LP2 = L + 2
         LPP1 = LPP2 - 1
         FIRSTC = 1
         LASTC = LPP1
         FIRSTR = LP1
         CALL INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
     X   IF'RINT, A, INC, NCON, NCONP1, PHILP1, NOWATE)
         IF (ISEL .NE. 1) GO TO 99
         GO TO 30
    3 CALL ADA (LP1,NL,N,NMAX,LPP2,IV,A,INC,T,ALF,MINO(ISEL,3))
      IF (ISEL .EQ. 2) GO TO 6
      FIRSTC = LP2
      LASTC = LPP1
      FIRSTR = (4 - ISEL)*L + 1
      GO TO 50
    6 FIRSTC = NCONP1
      LASTC = LP1
      IF (NCON .EQ. 0) GO TO 30
      IF (A(1, NCON) .EQ. SAVE) GO TO 30
         ISEL = -7
         CALL VARERR (IFRINT, ISEL, NCON)
         GO TO 99
   30 IF (PHILP1) GO TO 40
         DO 35 I = 1, N
   35       R(I) = Y(I)
         GO TO 50
   40    DO 45 I = 1, N
   45       R(I) = Y(I) - R(I)
   50 IF (NOWATE) GO TO 58
      DO 55 I = 1, N
         ACUM = W(I)
         DO 55 J = FIRSTC, LASTC
   55       A(I, J) = A(I, J) * ACUM
   58 IF (L .EQ. 0) GO TO 75
      DO 70 K = 1, L
         KP1 = K + 1
         IF (ISEL .GE. 3 .OR. (ISEL .EQ. 2 .GND. K .LT.NCONP1)) GO TO 66
         ALPHA = DSIGN(XNORM(N+1-K, A(K, K)), A(K, K))
         U(K) = A(K, K) + ALPHA
         A(K, K) = -ALPHA
         FIRSTC = KP1
         IF (ALPHA .NE. 0.0) GO TO 66
         ISEL = -8
         CALL VARERR (IFRINT, ISEL, K)
         GO TO 99
   66    BETA = -A(K, K) * U(K)
         DO 70 J = FIHSTC, LASTC
            ACUM = U(K)*A(K, J)
            DO 68 I = KP1, N
   68          ACUM = ACUM + A(I, K)*A(I, J)
            ACUM = ACUM / BETA
            A(K,J) = A(K,J) - U(K)*ACUM
            DO 70 I = KP1, N
```

```
70            A(I, J) = A(I, J) - A(I, K)*ACUM
75 IF (ISEL .GE. 3) GO TO 85
   RNURM = XNORM(N-L, R(LP1))
   IF (ISEL .EQ. 2) GO TO 99
   IF (NCON .GT. 0) SAVE = A(1, NCON)
85 IF (L .GT. 0) CALL BACSUE (NMAX, L, A, R)
   DO 95 I = FIRSTR, N
      IF (L .EQ. NCON) GO TO 95
      M = LP1
      DO 90 K = 1, NL
         ACUM = 0.
         DO 88 J = NCONPI, L
            IF (INC(K, J) .EQ. 0) GO TO 88
            M = M + 1
            ACUM = ACUM + A(I, M) * R(J)
88          CONTINUE
         KSUB = LP1 + K
         IF (INC(K, LP1) .EQ. 0) GO TO 90
         M = M + 1
         ACUM = ACUM + A(I,M)
90       A(I,KSUB) = ACUM
95    A(I, LNL2) = R(I)
99 RETURN
   END
C
   SUBROUTINE INIT(L, NL, N, NMAX, LPP2, IV, T, W, ALF, ADA, ISEL,
  X IPHINT, A, INC, NCON, NCONP1, PHILP1, NOWATE)
   DOUBLE FRECISION A(NMAX, LPP2), ALF(NL), T(NMAX, IV), W(N),
  X DSQRT,AS,BS,S
   INTEGER OUTFUT, P, INC(12, 8)
   LOGICAL NOWATE, PHILP1
    EXTERNAL ADA
   DATA OUTFUT /6/
   LP1 = L + 1
   LNL2 = L + 2 + NL
   IF (L .GE. 0 .AND. NL .GE. 0 .AND. L+NL .LT. N .AND. LNL2 .LE.
  X LPP2 .AND. 2*NL + 3 .LE. NMAX .AND. N .LE. NMAX .AND.
  X IV .GT. 0 .AND. .NOT. (NL .EQ. 0 .AND. L .EQ. 0)) GO TO 1
      ISEL = -4
      CALL VARERR (IFRINT, ISEL, 1)
      GO TO 99
 1 IF (L .EO. 0 .OH. NL .EQ. 0) GO TO 3
      DO 2 J = 1, LP1
         DO 2 K = 1, NL
 2          INC(K, J) = 0
 3 CALL ADA (LP1, NL, N, NMAX, LPP2, IV, A, INC, T, ALF, ISEL)
   NOWATE = .TRUE.
   DO 9 I = 1, N
      NOWATE = NOWATE .AND. (W(I) .EQ. 1.0)
      IF (W(I) .GE. 0.) GO TO 9
      ISEL = -6
      CALL VARERR (IFRINT, ISEL, I)
      GO TO 99
 9    W(I) = DSQRT(W(I))
   NCON = L
   NCONPI = LP1
   PHILP1 = L .EQ. 0
   IF (PHILP1 .OR. NL .EQ. 0) GO TO 99
```

```
      P = 0
      DO 11 J = 1, LP1
         IF (P .EQ. 0) NCONPI = J
         DO 11 K = 1, NL
            INCKJ = INC(K, J)
            IF (INCKJ .NE. 0 .AND. INCKJ .NE. 1) GO TO 15
            IF (INCKJ .EQ. 1) P = P + 1
   11       CONTINUE
      NCON = NCONPI - 1
      IF (IFRINT .GE. 0) WRITE (OUTPUT, 210) NCON
      IF (L+P+2 .EQ. LPP2) GO TO 20
   15 ISEL = -5
      CALL VARERR (IFRINT, ISEL, 1)
      GO TO 99
   20 DO 25 K = 1, NL
   25    IF (INC(K, LP1) .EQ. 1) PHILP1 = .TRUE.
   99 RETURN
  210 FORMAT (33H0  NUMBER OF CONSTANT FUNCTIONS =, I4 /)
      END
      SUBROUTINE BACSUB (NMAX, N, A, X)
      DOUBLE FRECISION A(NMAX, N), X(N), ACUM,AS,BS,S
      X(N) = X(N) / A(N, N)
      IF (N .EQ. 1) GO TO 30
      NP1 = N + 1
      DO 20 IBACK = 2, N
         I = NP1 - IBACK
         IP1 = I + 1
         ACUM = X(I)
         DO 10 J = IP1, N
   10       ACUM = ACUM - A(I,J)*X(J)
   20    X(I) = GCUM / A(I,I)
   30 RETURN
      END
      SUBROUTINE POSTPR(L, NL, N, NMAX, LNL2, EPS, RNORM, IPRINT, ALF,
     X W, A, R, U, IERR)
      DOUBLE FRECISION A(NMAX, LNL2), ALF(NL), R(N), U(L), W(N), ACUM,
     X EPS, PRJKES, RNORM, SAVE, DABS,AS,BS,S
      INTEGER OUTPUT
      DATA OUTPUT /6/
      LP1 = L + 1
      LPNL = LNLZ - 2
      LNLI = LPNL + 1
      DO 10 I = 1, N
   10    W(I) = W(I)**2
      IF (L .EQ. 0) GO TO 30
      DO 25 KBACK = 1, L
         K = LP1 - KBACK
         KP1 = K + 1
         ACUM = 0.
         DO 20 I = KP1, N
   20       ACUM = ACUM + A(I, K) * R(I)
         SAVE = R(K)
         R(K) =  ACUM / A(K, K)
         ACUM = -ACUM / (U(K) * A(K, K))
         U(K) = SAVE
         DO 25 I = KP1, N
```

```
25          R(I) = R(I) - A(I, K)*ACUM
30    ACUM = 0.
      DO 35 I = 1, N
35       ACUM = ACUM + R(I)
      SAVE = ACUM / N
      IF (NL .EQ. 0) GO TO 45
      CALL ORFAC1(NL+1, NMAX, N, L, IFRINT, A(1, L+2), FRJRES, 4)
      DO 40 I = 1, N
         A(I, LNL2) = R(I)
         DO 40 K = LP1, LNLI
40          A(I, K) = A(I, K+1)
45    A(1, LNL2) = HNOKM
      ACUM = RNORM*RNORM/(N - L - NL)
      A(2, LNL2) = ACUM
      CALL COV(NMAX, LFNL, ACUM, A)
      IF (IFRINT .LT. 0) GO TO 99
      WRITE (OUTFUT, 209)
      IF (L .GT. 0) WRITE (OUTPUT, 210) (U(J), J = 1, L)
      IF (NL .GT. 0) WHITE (OUTPUT, 211) (ALF(K), K = 1, NL)
      WRITE (OUTFUT, 214) RNORM, SAVE, ACUM
      IF (DABS(SAVE) .GT. EPS) WHITE (OUTPUT, 215)
      WRITE (OUTFUT, 209)
99    RETURN
209   FOHMAT (1HO, 50(1H'))
210   FORMAT (20HO  LINEAR PARAMETERS // (7E15.7))
211   FORMAT (23HO  NONLINEAR PARAMETERS // (7E15.7))
214   FORMAT (21HO  NORM OF RESIDUAL =, E15.7, 33H EXPECTED ERROR OF O
     XERVATIONS =, E15.7, / 39H   ESTIMATED VARIANCE OF OBSERVATIONS =
     X E15.7 )
215   FORMAT (95H  WARNING -- EXPECTED ERROR OF OBSERVATIONS IS NOT ZE
     X.   COVARIANCE MATRIX MAY BE MEANINGLESS. /)
      END
      SUBROUTINE COV(NMAX, N, SIGMA2, A)
      DOUBLE FRECISION A(NMAX, N), SUM, SIGMA2,AS,BS,S
C
      DO 10 J = 1, N
10       A(J, J) = 1./A(J, J)
      IF (N .EQ. 1) GO TO 70
      NM1 = N - 1
      DO 60 I = 1, NM1
         IP1 = I + 1
         DO 60 J = IP1, N
            JM1 = J - 1
            SUM = 0.
            DO 50 M = I, JM1
50             SUM = SUM + A(I, M) * A(M, J)
60          A(I, J) = -SUM * A(J, J)
70    DO 90 I = 1, N
         DO 90 J = I, N
            SUM = 0.
            DO 80 M = J, N
80             SUM = SUM + A(I, M) * A(J, M)
            SUM = SUM * SIGMA2
            A(I, J) = SUM
90          A(J, I) = SUM
      RETURN
      END
```

```
      SUEROUTINE VARERR (IFRINT, IERR, K)
      DOUBLE PRECISION AS,BS,S
      INTEGER ERRNO, OUTPUT
      DATA OUTPUT /6/
      IF (IFRINT .LT. 0) GO TO 99
      ERRNO = IABS(IERR)
      GO TO (1, 2, 99, 4, 5, 6, 7, 8), ERRNO
    1 WHITE (OUTPUT, 101)
      GO TO 99
    2 WRITE (OUTPUT, 102)
      GO TO 99
    4 WRITE (OUTPUT, 104)
      GO TO 99
    5 WHITE (OUTPUT, 105)
      GO TO 99
    6 WRITE (OUTPUT, 106) K
      GO TO 99
    7 WHITE (OUTPUT, 107) K
      GO TO 99
    8 WRITE (OUTPUT, 108) K
   99 RETURN
  101 FORMAT (46H0  FROBLEM TERMINATED FOR EXCESSIVE ITERATIONS //)
  102 FORMAT (49H0  PROBLEM TERMINATED BECAUSE OF ILL-CONDITIONING //)
  104 FORMAT (/ 50H INFUT ERROR IN FARAMETER L, NL, N, LFF2, OR NMAX. /:
  105 FORMAT (68H0  ERROR -- INC MATRIX IMPROPERLY SFECIFIED, OR DISAGRE
     XES WITH LFF2. /)
  106 FORMAT (19H0  ERROR -- WEIGHT(, I4, 14H) IS NEGATIVE  /)
  107 FORMAT (28H0  ERROR -- CONSTANT COLUMN , I3, 37H MUST BE COMPUTED
     XONLY WHEN ISEL = 1.  /)
  108 FORMAT (33H0  CATASTROPHIC FAILURE -- COLUMN , I4, 28H IS ZERO, SE
     XE DOCUMENTATION. /)
      END
      DOUBLE PRECISION FUNCTION XNORM(N, X)
      DOUBLE PRECISION X(N), RMAX, SUM, TERM, DABS, DSRHT
C             FIND LARGEST (IN ABSOLUTE VALUE) ELEMENT
      RMAX = 0.
      DO 10 I = 1, N
         IF (DABS(X(I)) .GT. RMAX) RMAX = DABS(X(I))
   10    CONTINUE
      SUM = 0.
      IF (RMAX .EQ. 0.) GO TO 30
      DO 20 I = 1, N
         TERM = 0.
         IF (RMAX + DABS(X(I)) .NE. RMAX) TERM = X(I)/RMAX
   20    SUM = SUM + TERM*TERM
   30 XNORM = RMAX*DSQRT(SUM)
   99 RETURN
      END
```

## APPENDIX E: ELECTRODE MOUNTING PROCEDURE

1. Drill .250 in. hole in plate.
2. Measure electrode O.D. with micrometer.
3. Using hand reamer with block to keep it aligned, ream hole so that the electrode can be press fit (if the fit is too tight, the electrode will deform and be wasted).
4. Clean electrode and hole with acetone.
5. Using the silver disc on the top of the electrode and the rubber hammer, tap the electrode part way into the plate.
6. Apply a bead of loctite cement around the electrode O.D.,
7. Tap electrode until it is flush with plate (+ .0015 in.). A dial indicator can be used to check this tolerance.
8. Wipe clean with acetone.

## APPENDIX F: MATERIALS SUPPLIERS

The cast aluminum plate was purchased from Castle Metals, San Francisco, 321-2500.

The plate was anodized at Sanford Metal Processing, Menlo Park, 327-5172.

The float glass was purchased at Acme Glass, Palo Alto, 321-7781.

If a thicker. pyrex glass is desired, try Heussner Optics, Santa Clara, 988-4214.

The brass parts used to construct the electrodes were purchased at Don's Hobby shop, Menlo Park, 322-7176.

The electrodes were gold plated at Hansen Labs, on campus.