Modeling Diffusion in the Antarctic Circumpolar Current

By:

Clifford H. Hodges

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology

April 30, 2004
[June 2004]

Author_____
Department of Electrical Engineering and Computer Science
April 30, 2004

Certified by_____
John Marshall
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Modeling Diffusion in the Antarctic Circumpolar Current

By:

Clifford H. Hodges

Submitted to the Department of Electrical Engineering and Computer Science

April 30, 2004

In Partial Fulfillment of the Requirements for the Degree of Master of Engineering in
Electrical Engineering and Computer Science

ABSTRACT

In order to understand the role of eddies in lateral mixing in a rotating fluid, a small scale laboratory model is constructed. An experiment is carried out in a rotating, differentially heated annulus and the evolution of a dye tracer mixed by turbulent motions is studied. Images are analyzed to extract the concentration mappings of tracer throughout the tank at each time instance and a diffusion coefficient *K(r)* is inferred.

Thesis Supervisor: John Marshall, PhD
Title: Professor of Atmospheric and Oceanic Sciences

Table of Contents

1. Introduction

Large-scale eddies under the influence of the earth's rotation play a central role in transferring properties around the globe. In the atmosphere these eddies are the 'weather systems' of middle latitudes – they are a major agency of lateral heat transfer carrying heat from the equator to the pole. The ocean is also full of 'weather systems', but they have a much smaller scale than their atmospheric counterpart. Their role in the general circulation of the ocean is much less clear, although it seems clear that they are central to the Antarctic Circumpolar Current (ACC) [1], carrying properties from one side of the current to the other - see Figure 1. In order to better understand the ACC, the role of these small-scale eddies must be characterized . Recently, *Marshall et al (2004)* have used an idealized tracer field driven by surface currents observed by satellite, to estimate the rate at which properties are transported by the eddy field across the mean path of the ACC [8].

In this project we describe a laboratory experiment designed to complement the *Marshall et al* study in a more controlled setting. The experiment is conducted to record the transport of dye across a cylindrical tank of water. Image analysis is performed on the resulting data to quantify the mass distribution of dye over time, and a diffusion coefficient *K(r)*, where *r* is the radius, is inferred.

[1] The Antarctic Circumpolar Current (ACC), which encircles the Antarctic continent and flows eastward through the southern portions of the Atlantic, Indian, and Pacific Oceans, is arguably the "mightiest current in the oceans" [1]. The ACC is the only current that flows completely around the globe and has a volume transport of approximately 108 m3/s, which is equivalent to 500 Amazon rivers. Edmond Halley, the British astronomer, discovered the ACC while surveying the region during the 1699-1700 HMS Paramore expedition. Later, the famous mariners James Cook in 1772-1775, Thaddeus Bellingshausen in 1819-1821, and James Clark Ross in 1839-1843 described the Atlantic Circumpolar Current in their journals. Other notable early expeditions were made by Sir Drake, who reached the tip of South America in 1578, Abel Tasman, who sailed south from Australia into the Southern Ocean in 1642, James Weddell in 1823, and by the HMS Challenger in 1873-74 [3].

2. Theory and Background

The tracer mixing experiment to be conducted in the laboratory is a small-scale model of a planetary-scale phenomenon. The figure below shows an idealized instantaneous tracer distribution in the ACC obtained by driving an advection-diffusion model using observations of surface ocean currents from satellite altimetry.

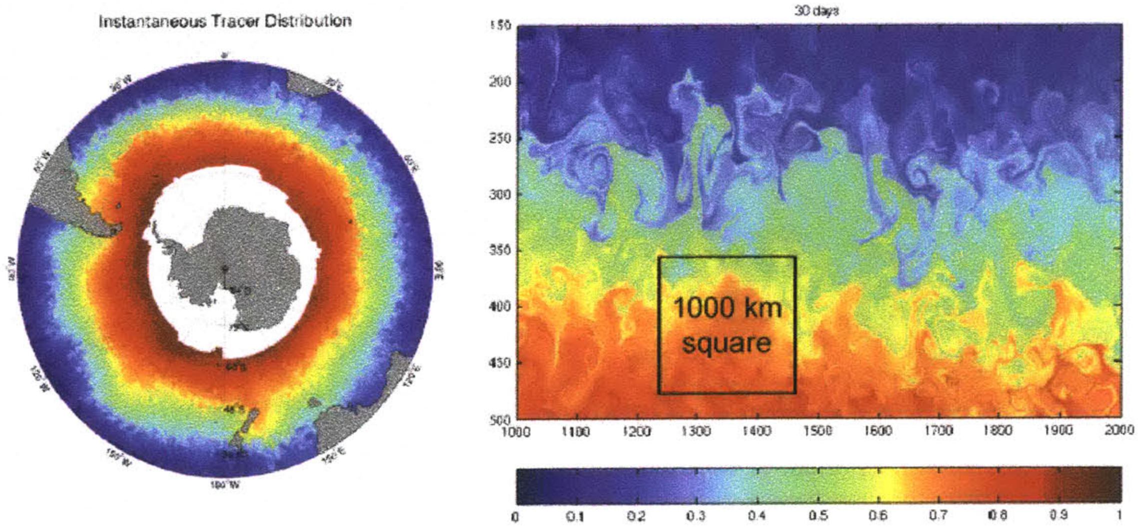


Figure 1 – Idealized tracer distribution obtained from satellite observation of surface ocean currents. *Marshall et al* have used this idealized tracer field to estimate diffusion rates across the ACC.

The tracer distribution q, in this case is governed by:

$$\frac{\partial q}{\partial t} + v \bullet \nabla q = k \nabla^2 q \qquad (1)$$

where *k* is a "small-scale" diffusion coefficient. From a study of the evolution of the idealized tracer initially given a monotonic distribution across, we can estimate the gross transfer properties of the eddy field. For this characterization, the problem can be constrained to 2-dimensions because the vertical component of velocity at the ocean surface is zero. For a 2-dimensional area as shown in Figure 2, the tracer evolves according to:

$$\frac{\partial}{\partial t} = -\nabla \bullet (N_q + qv) \qquad (2)$$

Here q**v** is the advection of q by the 2d flow and **N**q is the non-advective which we set equal to the diffusive flux $-k\nabla q$.

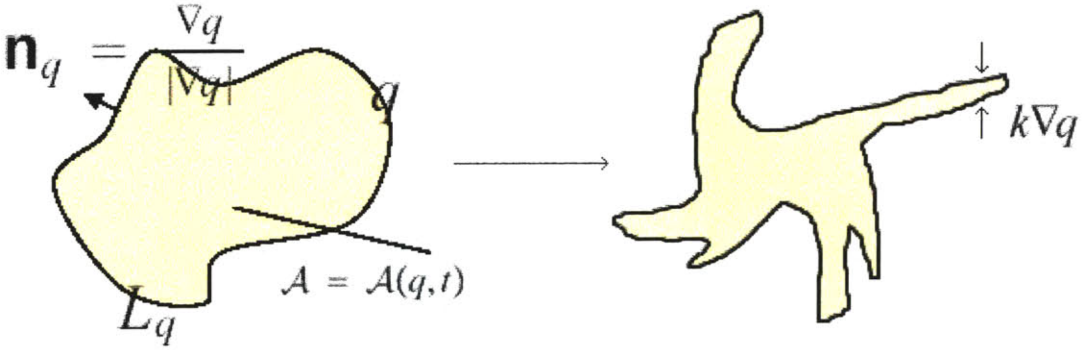

$\mathbf{n}_q = \frac{\nabla q}{|\nabla q|}$
q
$\mathcal{A} = \mathcal{A}(q,t)$
L_q
$k\nabla q$

Figure 2 – 2-Dimensional tracer area evolution shows two tracer profiles of equal area. Eddies strain the tracer contours creating small-scale gradients which are acted on by diffusion. The mixing of tracers can thus be viewed as a redistribution process of mass across the tracer contours due to microscale diffusion. [4]

The study described here serves to extract eddy diffusivities to compare to these theoretical models. The dye distribution in the laboratory experiment evolves from a ring of dye floated on the surface of an eddying flow. Figure 3 below shows a typical dye distribution.

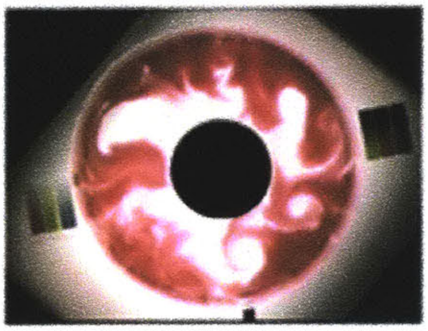


Figure 3 – Dye distribution from an experiment where the dye was delivered as a ring at the edge of the tank, this image is approximately 1 minute into the experiment. Eddies can be seen throughout the tank carrying the dye across the domain.

From the laboratory experiment discussed in the following section, an azimuthal averaging of the mass distribution of dye in the cylindrical tank yields the schematic picture shown below in Figure 4. The bold step function is how the dye distribution appears ideally at the start of the experiment, positioned around the edge of the tank. Beginning as a tight annular ring, the dye slowly spreads out and eventually reaches a state of equal concentration throughout the tank.

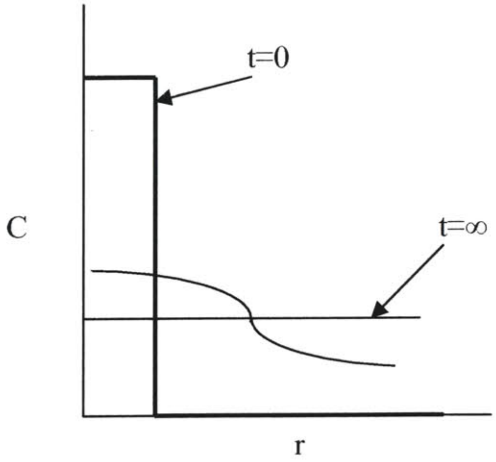


Figure 4 – A theoretical evolution of dye concentration is shown above. The dye begins as a thin band represented by the step function at t=0. At t=∞ the dye has diffused to uniform concentration throughout the tank.

Thus we have condensed the problem to a 2-dimensional scenario in cylindrical coordinates where the average dye concentration varies with radial distance over time. This concentration distribution allows us to extract a diffusivity *K(r)*, which is assumed constant over time, but varies with space radially throughout the tank.

The values obtained for *K(r)* may then be compared to different theoretical models. By solving the diffusion equation (below) in radial coordinates – where C is the tracer concentration and r is the radial distance – we obtain a numerical solution.

$$\frac{\partial C}{\partial t} = \frac{1}{r} \cdot \frac{\partial}{\partial r} \cdot (r \cdot K(r) \cdot \frac{\partial C}{\partial r}) \qquad (3)$$

Another complimentary method to find effective diffusivities is that of *Nakamura* [4]. Effective diffusivity is a measure of the geometric structure of a tracer field - mixing regions have large stretching rates, thus producing large effective diffusivities. While barrier regions have small stretching rates producing small effective diffusivities [5]. The simplest way of expressing effective diffusivity is with the following equation:

$$K_{eff} = k \frac{L^2}{L_{eq}^2} \qquad (4)$$

Where L is the length of a chosen tracer contour and L_{eq} is the circumference of a circle enclosing the same area as the tracer contour. Thus diffusivity can be understood as a simple ratio of lengths. *Nakamura* explained how tracer distributions as in Figure 1, can be used to estimate this "effective diffusivity" as the rate at which a tracer is diffused from one side of the domain to the other.

So the presented challenge is to (1) set up a laboratory analogue, (2) measure dye concentrations, and (3) from those dye concentrations infer a *K(r)*.

3. Laboratory Work

The laboratory work conducted is a small-scale modeling of the theory discussed in the previous section in order to test the numerical methods and compare the results with measured diffusivities in the lab. The experiment uses a cylindrical cross-sectioned container filled with salt-water and placed on a circular rotating table as shown below in Figure 5 to model the southern oceans.

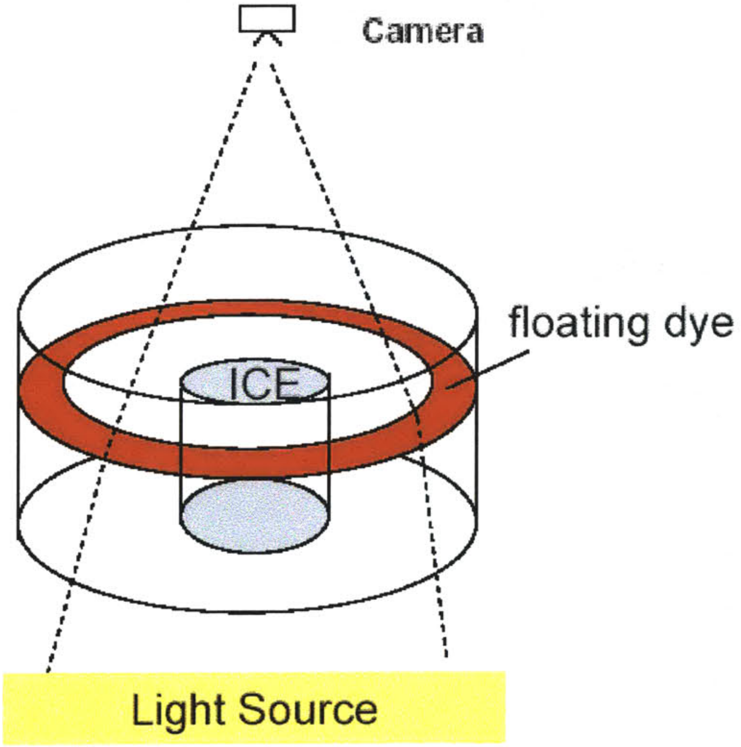
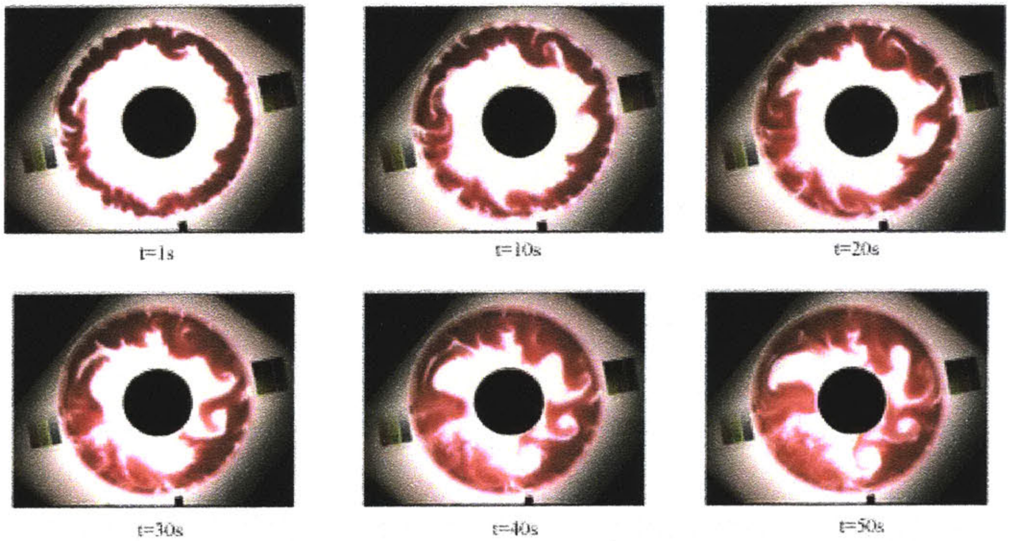


Figure 5 – The laboratory setup consists of a cylindrical water tank placed on a rotating circular table, with a camera attached to the ceiling and centered on the table. The tank rests on top of a light emitting table and contains an ice bucket at it's center. Detailed dimensions of the equipment are included in appendix B.

Salt water is used to increase the density of the water in the tank, allowing the delivered ring of dye to float. This keeps the dye at the surface of the water and constrains the analysis to 2-dimensions. A small can is placed in the center of the tank which will later hold ice, a light table is positioned underneath the tank, and a camera is then placed several feet above the table and attached to a rotating axel which is synchronized to spin with the table.

The purpose of the ice is to create flows in the tank and model the south pole.[2] The temperature gradient induced by the ice generates currents in the tank which stir the tracer. A thin ring of floating dye is then delivered around the edge of the tank and a camera begins recording the evolution of the dye tracer.[3] Actual frame sequences from a performed experiment are shown below in Figure 6 at intervals of 10 seconds.



t=1s t=10s t=20s

t=30s t=40s t=50s

Figure 6 – The evolution of the dye tracer is shown above. The sequence shown above is observed over 50 seconds, with captured frames every 10 seconds. Normal experiment durations are 1-3 minutes.

[2] Detailed information on laboratory equipment may be found in Appendix B.

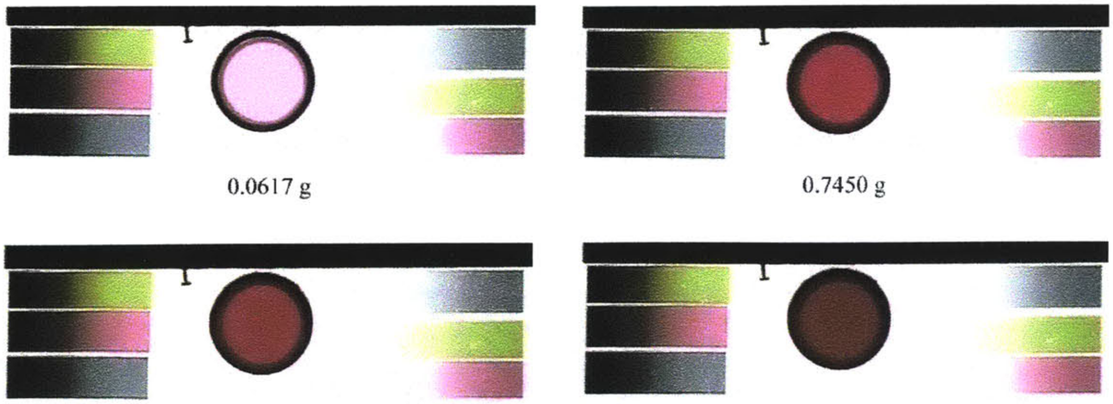
[3] In all experiments and throughout the remainder of this paper, "dye" refers to a 10:1 mixture of water to red food coloring. "Water" refers to salt water raised to approximately 38 psu using consumer-grade kosher salt.

3.1 Calibration

The laboratory equipment must first be tested and calibrated to create a basis of understanding to perform a colorimetric analysis of the frame sequences of dye evolution. This colorimetric analysis of obtaining dye masses from perceived dye color intensities is discussed in depth in section 4. The calibration is performed by placing a small, known mass of dye into a clear plastic cylinder that is placed on the light table where the center of the tank normally lies. By measuring the intensity of the light that passes up through the dye, we can create a 1:1 relation between perceived color intensity and dye concentration. Forty of these calibration experiments were run with dye masses ranging 0-4 grams of dye.[4] After these data points were collected, curves were fitted to the data to produce an invertible function by which we can later obtain mass from a given color intensity. This fitting was done in MATLAB and it was found that a simple sum of 2 decaying exponential functions fit the data quite well. The results from this phase produce calibration curves, relating the color intensity of each channel (R,G,B) to dye mass.[5] Examples of the calibration experiments are shown below in Figure 7.

[4] Table of measured dye masses from the calibration experiments is available in Appendix A.

[5] A detailed plot of the data and calibration curves is attached in Appendix A.



0.0617 g

0.7450 g

1.0333 g

2.7239 g

Figure 7 – Images from the calibration experiments. The cylindrical container of dye has a diameter of 6cm and a height of 12cm. Included in each image are color swatches that range over the entire RGB color scale. These areas of constant color are used to account for any exposure changes by the camera.

3.2 Experiment

Once the calibration curves have been calculated, the experiment with the rotating annulus may be carried out. A detailed description of the lab equipment is included in appendix B. On top of the rotating table is placed a small light table containing four 18-watt fluorescent bulbs. Several inches above the light table rests a large sheet of plastic. The water tank rests on the plastic and contains a small ice bucket at its center. Water is then poured into the tank and the table begins to spin at 9.5 rev/min. The tank is left to spin for approximately 15 minutes to ensure that the water has reached a state of solid body rotation. Position markers and color swatches are also placed on the plastic to be used later in the image analysis phase for image registration and color-correction as discussed in sections 4.2.1 and 4.2.2.

Once solid-body rotation has been achieved, ice is added to the center bucket. The purpose of the ice is to induce circulation by modeling the presence of the pole. A thin ring of dye is delivered around the edge of the tank.[6] The evolution of the dye is captured by a video camera suspended from the ceiling and attached to a rotating camera connection. The camera used is a consumer-grade Sony Digital Camcorder. Once the entire evolution of the dye is recorded, it is transferred to disk by capturing frames every second at a resolution of 640x482 pixels. The experiments are run on for an average of 1-3 minutes.

[6] See appendix C for detailed explanation of delivery device.

4. Image Analysis

4.1 Colorimetric Analysis Theory

In order to obtain the tracer mass distribution which will be used to calculate the aforementioned azimuthal averages, an image analysis technique is needed. The basic idea of our method was developed by *Hill* [6] and is based on the Beer-Lambert relation. Also known as Beer's Law, this relation states that when light passes through a given material, the amount of light absorbed by that material is proportional to the concentration of the absorbent species. In our case, that is when light passes through the water and the dye in the tank, a portion of that light is attenuated by the dye. By performing the calibration experiments discussed in section 3.1, we can come up with a direct relation between the perceived light intensity and the concentration of the dye that the light had to pass though on its way to the camera lens.

The light table on which the tank stands is emitting photons upward toward the camera. As the photons pass through the water and the dye, a portion of the light is attenuated that is proportional to the amount of dye present. The amount of light attenuated then results in a particular color intensity recorded by the camera – 0-254 per channel on the 8-bit RGB scale used by the camera. According to Beers Law, the observed color intensity *I* of light passing through an absorbent medium is:

$$I = I_o e^{-\alpha l c} \qquad (4)$$

where I_o is the intensity of the incident light, l is the distance that the light travels through the material (the path length), c is the concentration of absorbing species in the material, and α is the absorption coefficient of the absorber. In our work, I_o is ignored as the base case is the tank with no dye, thus appearing as white to the camera and saturating all channels. I is measured on the RGB scale and α is obtained from the calibration experiments. Then we invert the curves to obtain *lc*. Using this idea of light attenuation and the calibration curves in Appendix A, a concentration-mapping of dye is calculated throughout the tank for each image and the azimuthal averaging can be performed as discussed in section 3.

4.2 Data Correction

Before the colorimetric analysis may be performed, errors in the data collection process must be corrected. The image capturing device is set up to rotate synchronously with the table, thus creating a rotating frame. The synchronization system is, however, not without flaws. There is a small high-frequency vibration in the rotation device combined with a bit of slippage (perhaps in the camera mount or the table itself) which causes a few degrees in rotation over the course of a 1-3 minute experiment. Thus the time-lapse images must be registered. Additionally, the capture device is a consumer-grade camera that has some inherent color-correction mechanisms in it that cannot be disabled. Thus there are slight changes in brightness and tone that must also be corrected.

4.2.1 Image Registration

The first step in the data-correction phase is to register the images. This is done by masking out most of the images and using a few target zones that were placed in the rotating frame as key reference points for registration. Below is a picture of the rotating frame as well as the mask.

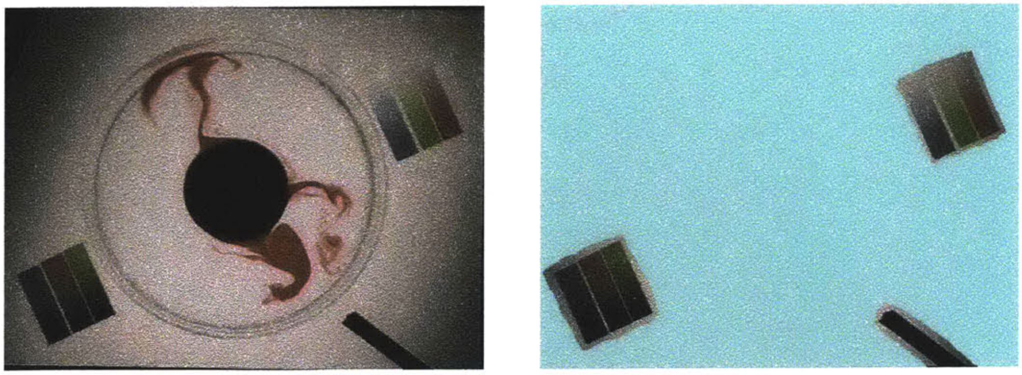


Figure 8 – On the left is shown an image of the entire rotating frame. Color swatches and flat bar are included in the frame to be used for color-correction and image registration. The image on the right is the mask used for registration. Only the color swatches and bar are visible and are used as regions of interest.

In this case, the registration is done as an optimization problem where the image to be registered is compared to the base image in the sequence. A sum-squared difference between the transformed image and the base image is calculated giving us an optimized fit using a 2-norm.[7]

4.2.2 Color Correction

The color-correction is performed in a similar manner as the registration. Each image is compared to the base image in the sequence. The entire image except for the RGB color swatch is masked out and each color swatch is examined as a region-of-interest. A comparison of the known color regions is performed in order to produce a transformation function for each channel to mathematically explain the color-correction that occurred in each frame. Then, each color channel of each image is, in effect, run backward through this function to correct for this problem.[8]

[7] The registration algorithm was implemented by Ed Hill, a postdoc researcher in the department of Earth and Planetary Sciences at MIT. The MATLAB code is attached in Appendix D.

[8] The color-correction algorithm was also implemented by Ed Hill in MATLAB and the code is attached in appendix D.

4.3 Concentration Calculation

Once the images have been registered and color corrected, the colorimetric analysis may be performed to produce concentration mappings for the image. This is done by examining each pixel of each image individually and finding the corresponding match of color-channel intensity to dye concentration using the calibration curves.

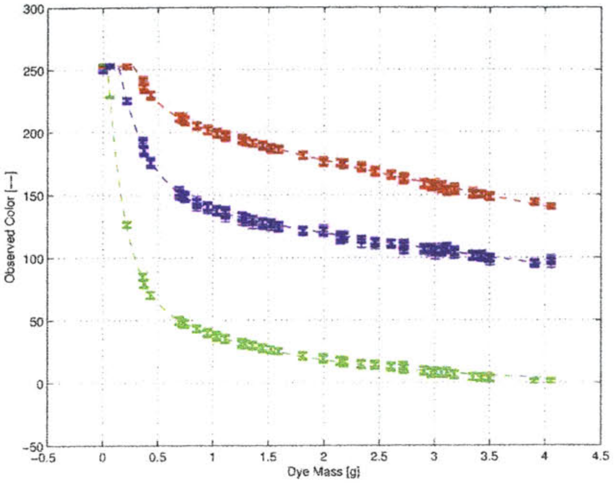


Figure 9 – The exponential curves from the calibration experiments relating color intensity to dye mass. The sill regions can be seen in the red and blue channels. While the green channel maintains a 1:1 relation throughout the range of masses. Observed color is shown on the standard 8-bit (0-254) intensity scale for RGB images.

It is noteworthy to mention here that while there are 3 calibration curves (one for each color channel, RGB) not all were used in this analysis. It was found that when using a red dye, the green color channel provided us with the best resolution. Looking at the curves above and in Appendix C, it may be seen that there is a small sill region in the low mass range for both the red and blue channels. Basically, the camera does not have the resolution to pick up small amounts of dye in those sill regions and saturates them all to white. Only the green channel produced an invertible curve. In our experiments, the ranges of mass that are being recorded per pixel are in the 0g-1g range, thus using the red or blue channels would not be possible. For this reason, all further analysis is done using data from the green channel only.

Once the images have been transformed into mass-mappings for the entire experiment, an azimuthal averaging is performed at each time instance to condense the problem to a 2-dimensional space of radius (r) and concentration (C). Some of these azimuthal averages are shown below in Figure 10.

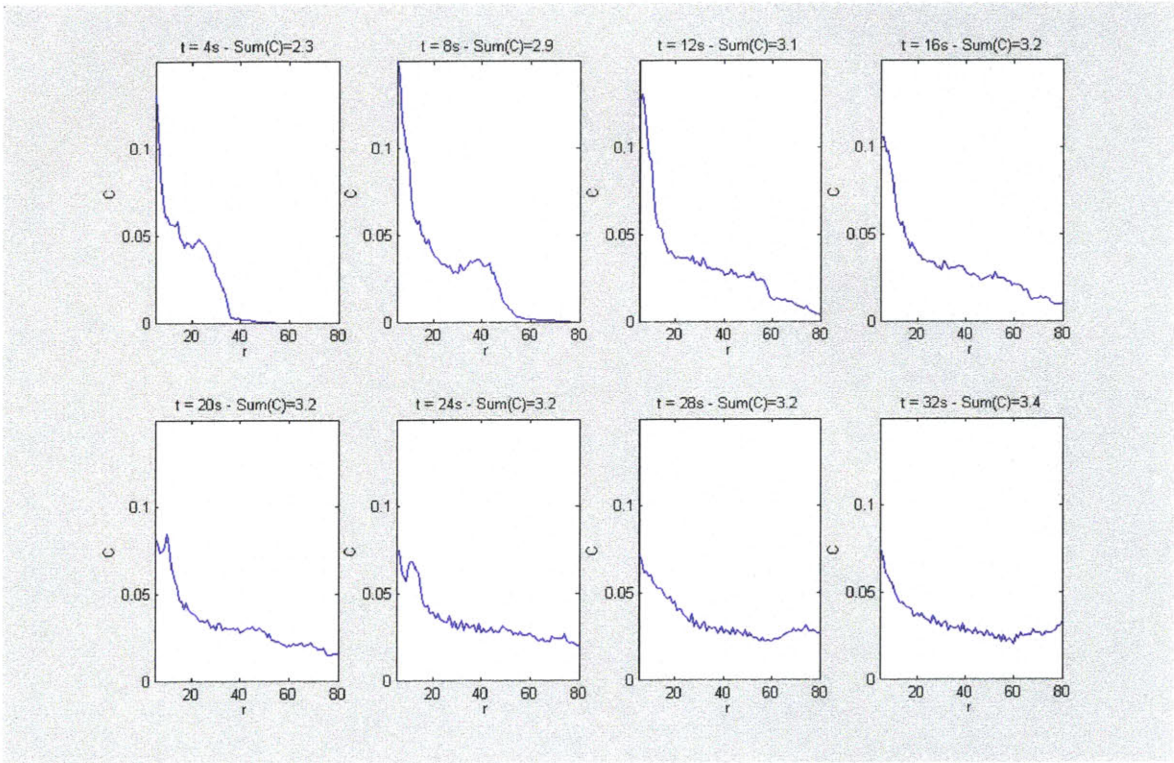


Figure 10 – Azimuthal averages of dye concentration over a short 32 second experiment. The vertical axis shows dye concentration and the horizontal axis is radial distance moving left to right from the ice bucket to the edge of the tank. The sharp increase in concentration at the edge of the bucket is due to the imperfections in data collection as discussed in this section.

The above figure shows the basic properties expected and discussed in section 2 as the dye diffuses from left to right over time. The behavior is not ideal however, due to the sharp spikes in concentration at the edge of the ice bucket (left side of graphs in above figure). This is due to imperfections in data collection. Our method of colorimetric analysis cannot distinguish between different materials in the frame of reference. It purely recognizes darker areas as regions of higher dye concentration. Thus when an edge of the bucket, or a shadow on the bottom of the tank appears in an area from which data is collected, the increase in perceived mass is unavoidable. Even with such errors, over a 2 minute experiment, the total estimated mass in the tank varies by no more than 12% of its original value.[9]

To reduce error, the experiment was modified to deliver a ring of dye at the halfway point in the tank, between the ice bucket and the edge of the tank. In doing so, dye may be measured before it reaches the edges where the data becomes corrupted. Below, Figure 11 shows the concentration averages for such an experiment. This experiment, dated April 12, 2004, is the basis for our analysis. All further calculations refer to work done with this data.

[9] The MATLAB code which calculates the dye concentrations and azimuthal averages, as well is the percentage error in total mass was implemented by the author and is attached in Appendix E.

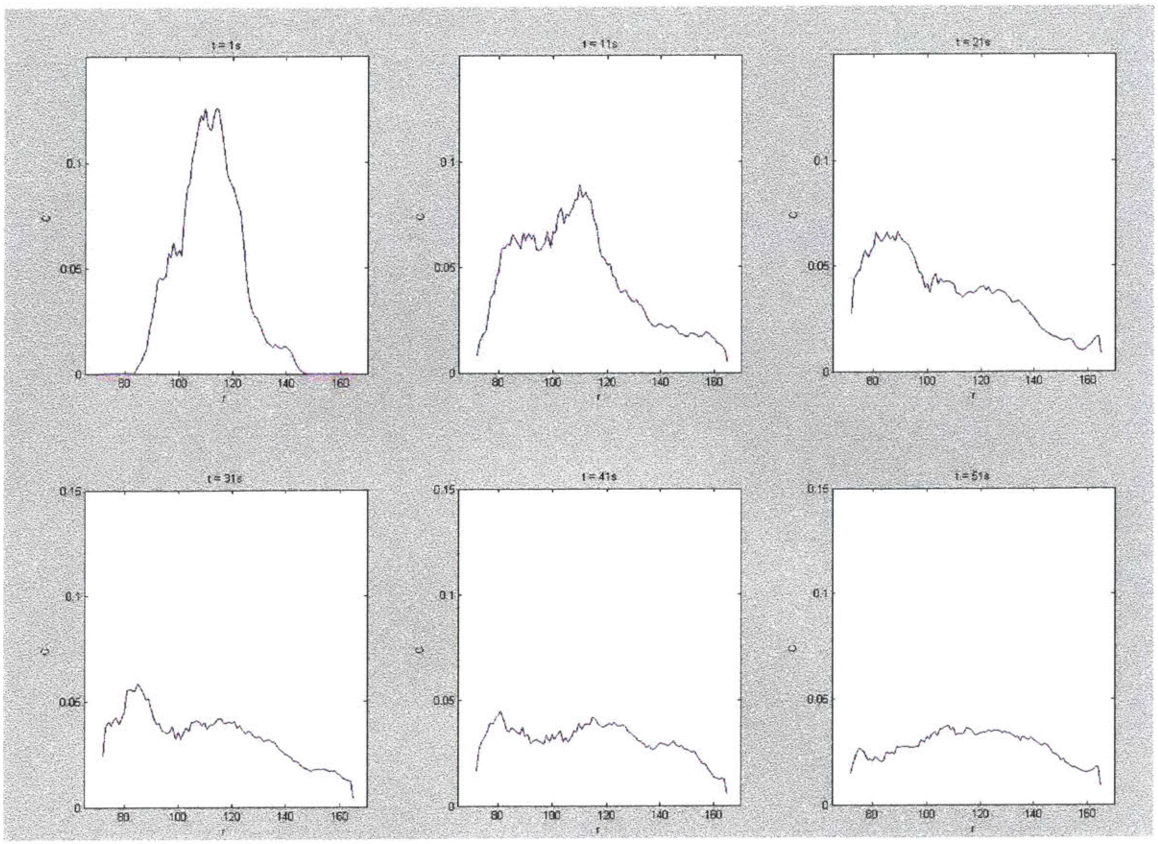


Figure 11 – Actual diffusion from the 04/12/2004 experiment in which the dye was delivered as a ring, halfway between the ice bucket and edge of the tank. The x-axis plots radial distance throughout the tank, and the y-axis plots dye concentration. The dye concentrations are in fact azimuthal averages, thus the y-axis is in essence plotting $C(r) = \frac{1}{2\pi r} \int C dr$

The azimuthal averaging is performed by a simple radial averaging. The concentration mappings have already been discretized into unit pixels by the camera. For each radius, r_i, from the edge of the ice bucket, out to the edge of the tank, each concentration value of every pixel at radius r_i is summed together and divided through by $2\pi r$. So the azimuthal concentration average at radius r_i, A_{ri} is equal to

$$A_{r_i} = \sum_{C} \frac{C_{r_i}}{2\pi r_i} \qquad (5)$$

5. Diffusion Coefficient Calculations

To fit a diffusion model to the experimental data, we approach it as an optimization problem. The first step is to solve the diffusion equation numerically to evolve a concentration profile (such as those shown in Figure 11) according to the radial diffusion equation discussed in section 2. In our analysis, the diffusion equation is viewed as a flux equation:

$$\frac{\partial C}{\partial t} = \frac{-1}{r} \cdot \frac{\partial F}{\partial r} \quad , \quad \text{where} \quad F = -rK(r)\frac{\partial C}{\partial r} \qquad (6)$$

Using a Lax Wendroff finite difference method[10], we choose to arrange the equation as

$$\{C^{i+1}\} = [A]\{C^{i}\} \qquad (7)$$

Where the vector of the concentration profile C^{i+1} at time $t = i+1$ is simply a time-independent matrix A multiplied by the concentration profile of the previous time step. The difference method used is an explicit, cell-centered, finite difference model that is 1st order accurate in both time and space. A conservative discretization is used to ensure that mass is conserved in both time and space. The grid for this difference method is shown below in Figure 12.

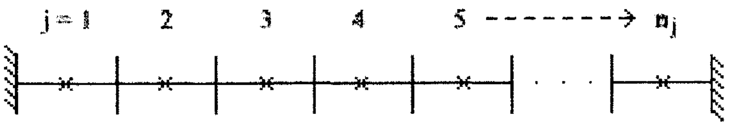
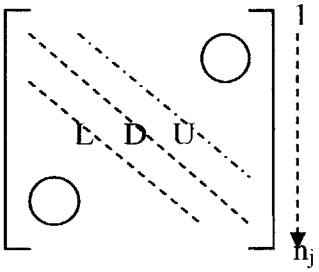


Figure 12 – The grid for the cell centered finite difference method used. The radial length of the tank is divided up into n_j cells, where n_j is on the order of 100.

[10] The equations used in this difference method were derived jointly by the author and Ed Hill.

The discretization starts at the left edge of Figure 12 which is the edge of the ice bucket in the experiment. The radial spacing is then discretized into n_j cells, where n_j is on the order of 100 (more discussion of this is included in section 5.1). With the arrangement shown above in equation (7), A has the below form.

A =



$$L_j = \frac{-\Delta t}{r_j(r_{j+1/2} - r_{j-1/2})}\left[\frac{-r_{j-1/2}K_{j-1/2}}{r_j - r_{j-1}}\right]$$

$$D_j = 1 + \frac{-\Delta t}{r_j(r_{j+1/2} - r_{j-1/2})}\left[\frac{-r_{j+1/2}K_{j+1/2}}{r_j - r_{j+1}} + \frac{-r_{j-1/2}K_{j-1/2}}{r_j - r_{j-1}}\right]$$

$$U_j = \frac{-\Delta t}{r_j(r_{j+1/2} - r_{j-1/2})}\left[\frac{-r_{j+1/2}K_{j+1/2}}{r_j - r_{j+1}}\right]$$

We choose a Δt (order 10^{-4} s) and K (order 10^{-5} m^2/s) – then, given an initial C profile, Eq (7) provides a prediction of the C distribution at any later time which can be compared with observations.[11] The values chosen for Δt and *K(r)* were selected empirically as best estimates to maintain stability of the problem. Below, Figure 13 shows the simulated evolution of a starting concentration profile that is identical to that in Figure 10 overlayed on the experimental diffusion shown in Figure 10.

[11] The MATLAB code which defines the diffusion simulator was implemented by the author and is attached in Appendix F.

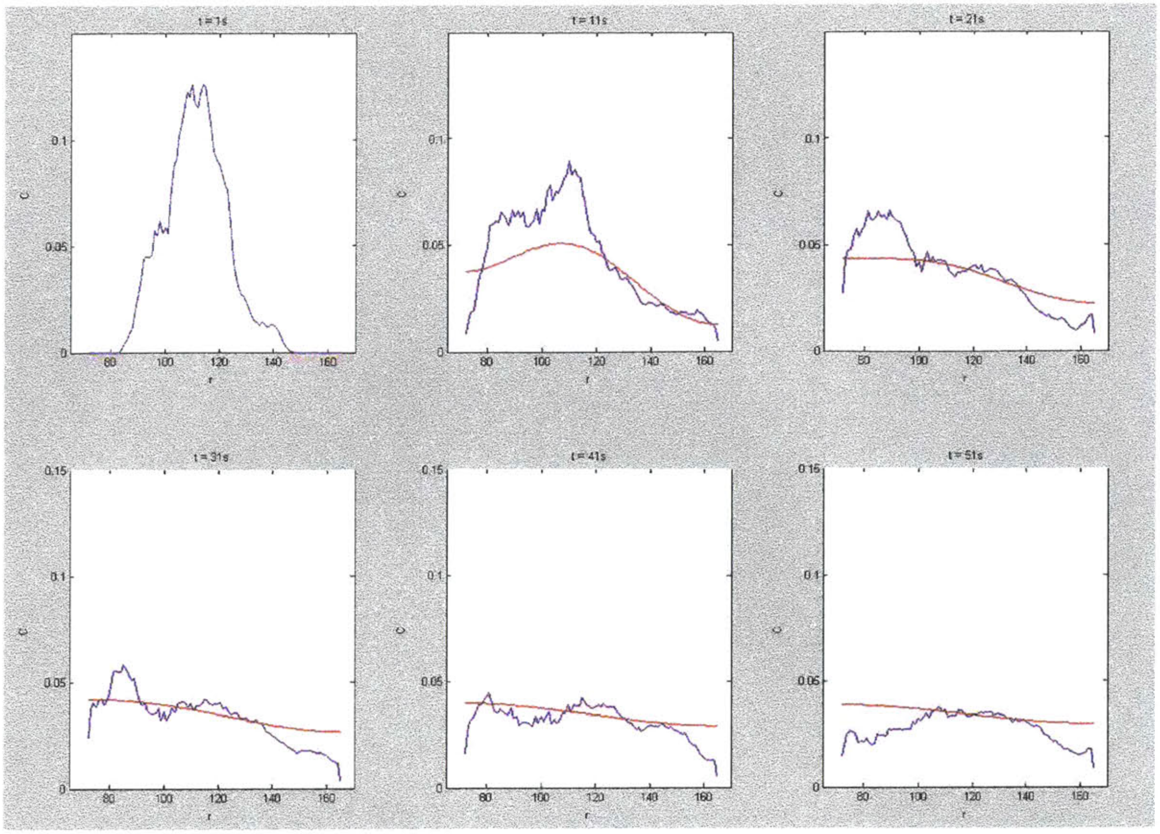


Figure 13– Simulated diffusion using the optimized *K(r)* results described in section 5.1. The experimental data is shown in blue and the simulated diffusion is overlayed in red. The axis are identical to Figure 11 with the x-axis plotting radial distance and the y-axis plotting $C(r) = \frac{1}{2\pi r}\int C dr$.

5.1 Diffusion Coefficient Optimization Results

In order to fit a diffusion coefficient *K(r)* to the experimental data, we use a simple optimization approach.[12] An initial guess for *K(r)* is fed into the optimizer in the form of a linear spline. *K(r)* is held at zero at both edges of the domain and 3 decision variables are used, evenly spread throughout the rest of the tank. The optimizer then calls the MATLAB function *fminsearch*, to use a low-dimension Nelder-Mead simplex optimization method to fit a *K(r)* to the data. [9] In addition to *K(r)* being held at zero on the edges, the only other constraint on the optimizer is that *K(r)* must be strictly non-

[12] The optimization algorithm was implemented by the author in MATLAB and is attached in appendix F.

negative throughout the tank. The fitted output for *K(r)* is shown below in Figure 14 when using 3 decision variables in the linear spline of *K(r)* along with the objective function for this optimization.

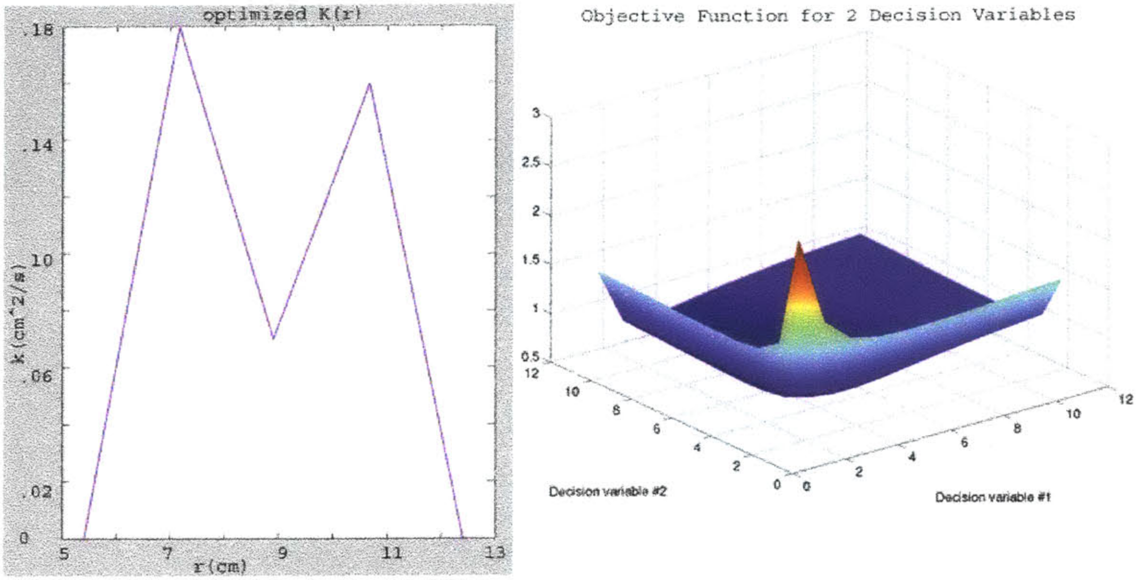


Figure 14 – On the left, the *K(r)* profile as fitted to the experimental data by the optimizer. The three spline points rise from zero to .18 cm^2/s, then down to .07 cm^2/s, and up again to .16 cm^2/s before returning to zero at the edge of the tank. The graph on the right shows the objective function of a hyper-plane cross-section for the optimization shown on the left. The third data point is held constant at its value of .16 cm^2/s and the first two decision variables are left free for optimization.

The objective function in Figure 14 shows that the optimization of *K(r)* is in fact finding a good fit and not getting stuck at local minima. The surface plot is quite smooth with a well defined global minimum.

The diffusion coefficient may be viewed as a velocity scale multiplied by a length scale and a correlation coefficient:

$$K(r) = c \cdot v \cdot L \qquad (8)$$

In our experiments, the velocity scale is approximately 2 cm/s and the length scale (distance from the ice bucket to the edge of the tank) is 10 cm. Thus the optimized *K(r)* values shown in Figure 14 yield a correlation coefficient on the order of *c = .01*.

As further demonstration of the validity of our method, it can be shown that Sum-Squared Error (SSE) of the simulated concentration values versus the experimental data also decreases as the number of decision variables in *K(r)* increases.

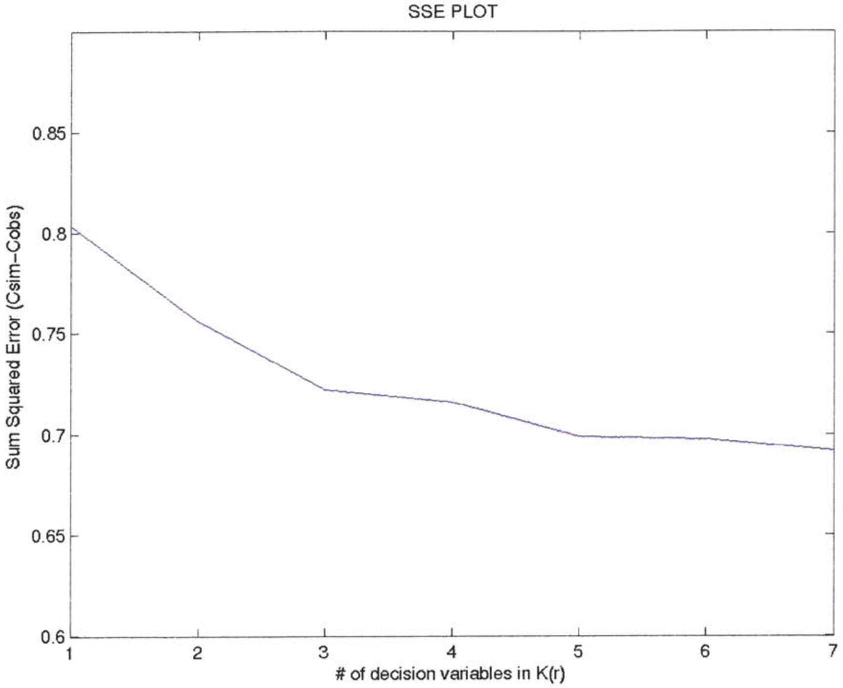


Figure 15 – The SSE plot of simulated concentration versus experimental concentration data is shown above. As the number of decision variables (spline points) in *K(r)* increases, the SSE decreases.

6. Conclusion

As it currently stands, the laboratory modeling and diffusion coefficient calculations have been a success as a "proof of concept". A laboratory model was constructed with a cylindrical water tank inspired by the southern oceans. A dye tracer delivered to the surface of the water was observed as it evolved through the tank and these experiments were recorded. The data correction issues (image registration and color-correction) have been resolved and the colorimetric analysis has proven to be effective. By conducting calibration experiments, we were able to use a Beer-Lambert type approach to relate the amount of light attenuated by the dye directly to the amount if dye present. In so doing, we were able to accurately map the concentration of dye throughout the tank. With this information, we then calculated the radial average of dye in the tank and used these calculations to extract a diffusion coefficient, *K(r)*, from the observed data.

The two major pushes in possible future work with this project are 1) comparison of the results with other theoretical models, and 2) improvement on the laboratory setup to acquire more effective data. A particular theoretical model of interest is that proposed by *Nakamura* [4]. A recommended future course of action would be to use the observed concentration maps from this study and apply them to the *Nakamura* theory to extract a *K(r)* in a different manner. Then these two methods may be compared.

It is also recommended that a larger-scale laboratory model be constructed. Due to the current size of the tank, the dye tracer diffuses rather quickly throughout the tank (approximately 1-2 minutes). To increase the observable time scale, a tank of at least twice the current size is advised. Additionally, this would create the need for a larger light-table apparatus as well as possibly requiring the camera unit to be placed farther away from the tank.

List of References

[1] Pickard, G. L., and W. J. Emery, 1990: Descriptive Physical Oceanography, An Introduction. Permagon Press, 5th Edition

[2] Deacon, G., 1984: The Antarctic Circumpolar Ocean. Cambridge University Press

[3] Nowlin, W. D., Jr., and J. M. Klinck, 1986: The Physics of the Antarctic Circumpolar Current. Rev. Geophys.

[4] Nakamura. N. 1995: Two-Dimensional Mixing, Edge Formulation, and Permeability Diagnosed in an Area Coordinate. Journal of the Atmospheric Sciences. Vol 53, No.11. American Meteorological Society.

[5] Haynes, P. and Shuckburgh, E. 2000: Effective Diffusivity as a Diagnostic of Atmospheric Transport. Paper number 2000JD900093. American Geophysical Union.

[6] E.H. Hill III, and C.T. Miller. Evaluation of Path-Length Estimators for Characterizing Multiphase Systems Using Polyenergetic X-ray Absorption.Soil Science 167(11):703-719, 2002..

[7] Lee, Sanboh, H-Y Lee, I-F Lee, and C-Y Tseng. 2004: Ink Diffusion in Water. European Journal of Physics. Vol 25 (2004). Institute of Physics Publishing.

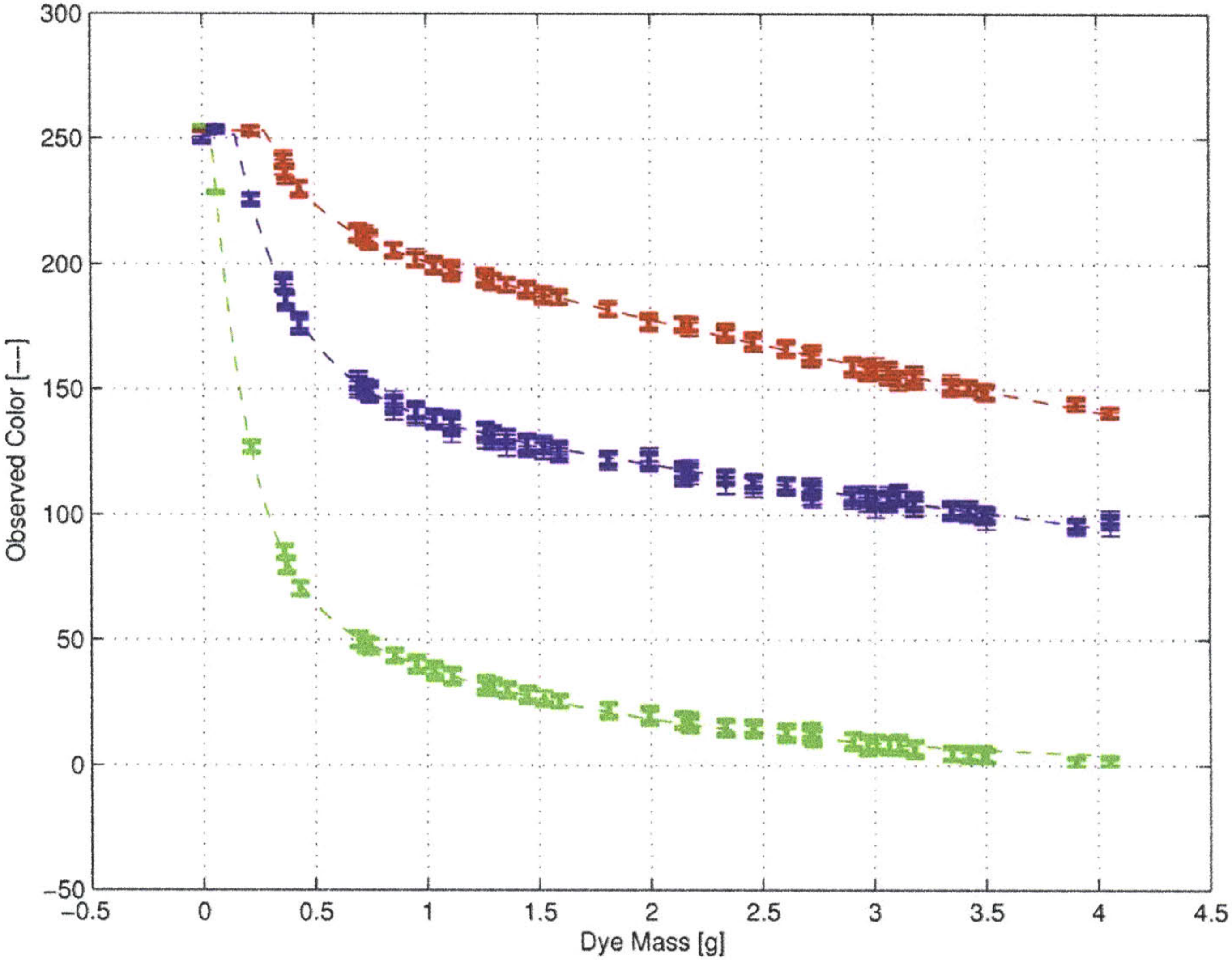
[8] John Marshall, Emily Shuckburgh, Helen Jones, and Chris Hill. 2004. New Estimates of Near Surface Eddy Diffusivity in the Southern Ocean and Implications for the Dynamics of the Antarctic Circumpolar Current: To be submitted to Journal of Physical Oceanography.

[9] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, Vol. 9 Number 1, pp. 112-147, 1998.

Appendix A: Calibration Experiment

The data points in the calibration curve on the following page are the measured dye masses from the 40 calibration experiments. Below is a table with these values with mass represented in grams.

Experiment #	Mass (g)	Experiment #	Mass (g)
0	0.0000	21	1.2934
1	0.0617	22	1.4479
2	0.3633	23	1.3569
3	0.2173	24	2.1753
4	4.0545	25	2.7165
5	1.1108	26	3.0077
6	1.9935	27	2.3361
7	3.1106	28	2.1458
8	0.3723	29	2.4591
9	0.6938	30	2.7239
10	0.7450	31	3.0644
11	0.8533	32	2.9039
12	0.4326	33	2.9692
13	0.9513	34	2.6066
14	0.7242	35	3.5026
15	1.5225	36	3.4901
16	1.2619	37	3.9038
17	1.1113	38	3.4251
18	1.5911	39	3.3476
19	1.8107	40	3.1799
20	1.0333		


300
250
200
150
100
50
0
−50
−0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Observed Color [---]
Dye Mass [g]

Appendix B: Laboratory Equipment

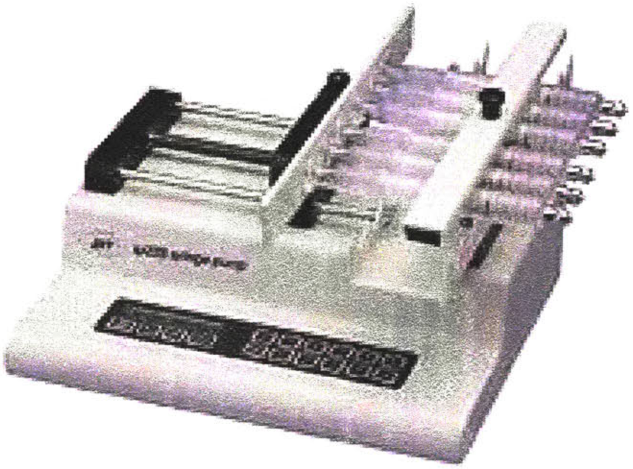
Water Tank	
Description:	Plastic Cylindrical tank to hold salt water and contain the experiment.
Diameter:	29.5 cm
Water Height:	10 cm
Height above table:	20 cm

Camera	
Description:	Digital Video Camera used to record experiments.
Vendor:	Sony
Water Height:	DCR-TRV20
Height above table:	84 cm

Light Table	
Description:	Artists light table positioned underneath tank to illuminate water.
Vendor:	Gagne Inc.
Model:	1824
Power:	Four 18-watt fluorescent bulbs

Appendix C: Dye Delivery Device

The dye delivery device consists of a World Precision Instruments SP220i Infusion Pump, 3 plastic syringes, and three 4 ft. long pieces of surgical tubing. The surgical tubing connects to the syringes, and the ends are held together just above the surface of the water in the spinning tank. This allows for the slow, controlled release of bands of dye around the tank. The SP220i Infusion Pump is shown below.



Appendix D: Image Registration and Color Correction

```
#  Create mask for large image.
#
#  Note: mask created using "The Gimp" with a painbrush and
#    the color 0,255,255 and then read into MatLAB
#
matlab65 -nojvm
clear all
  m0 = imread('register/mask_000.jpg');
  m1 = (m0(:,:,1) < 20) & (m0(:,:,2) > 240) & (m0(:,:,3) > 240);
  m2 = uint8(~m1);
  image(m2)
  imwrite(m2,'register/mask_001.jpg','Quality',100);
  save m2 m2

#  Determine the registration params
#
matlab65 -nojvm
clear all
  load m2
  mask = double(m2);
  bimg = double(imread('raw_data/t0412_000.jpg'));

  ofun = inline('fit_imtrans(P1,P2,x,P3)', 3)
  opts = optimset;
  opts = optimset('MaxFunEvals',45);

  res = [];
  xstart = [ 0 0 0 ];
  for i=1:61,
    comm = sprintf('img = double(imread(''raw_data/t0412_%03d.jpg''));', i);
    disp(comm)
    eval(comm);
    fit = fminunc(ofun, xstart, opts, bimg, img, mask);
    xstart = fit;
    res = [ res ; i fit ];
  end

  sprintf(' %3d  %12.5f  %12.5f  %12.5f\n', res')
  fid = fopen('res_003.txt','w');
  fprintf(fid, ' %3d  %12.5f  %12.5f  %12.5f\n', res');


#   Using the registration values from the full-size images,
#   create the transformed images.
#
matlab65 -nojvm
clear all

  load res_003.txt
  res = res_003(:,2:4);

  n = 1;
  for i=1:61,
    comm = sprintf('img = imread(''raw_data/t0412_%03d.jpg'');', i);
    disp(comm)
    eval(comm);
    x = [ res(n,1) res(n,2) res(n,3) ];

    simg = size(img);
    U = [ 0 0 ; 1 0 ; 0 1 ];
    M = U;
    phi = x(3)*(pi/180);
    rot = [ cos(phi) sin(phi) ; cos(phi+pi/2) sin(phi+pi/2)];
    M(2:3,:) = M(2:3,:)*rot;
    M(:,1) = M(:,1) + x(1);
    M(:,2) = M(:,2) + x(2);
    T = maketform('affine', U, M);
    dit = imtransform(img, T, ...
            'XData', [1 simg(2)], 'YData', [1 simg(1)], ...
            'FillValues', 0);

    comm = sprintf('imwrite(dit,''registered/r_0412_%03d.jpg'',%s);', i, '''Quali
ty'',100');
    eval(comm);
    n = n + 1;
  end
```

```
#  Create a mask for the color correction
#
cp registered/r_0412_061.jpg registered/r_mask_01.jpg
gimp registered/r_mask_01.jpg
#  use color [ 0 255 255 ] and save as "r_mask_01.jpg"

matlab65 -nojvm
clear all
  m2 = imread('registered/r_mask_01.jpg');
  image(m2)
  m3 = (m2(:,:,1) < 10) & (m2(:,:,2) > 240) & (m2(:,:,3) > 240);
  m3 = uint8(~m3);
  image(m3)
  imwrite(m3,'registered/r_mask_02.jpg')
  save registered/rmask_3 m3



#  Calculate color-corrected images using the registered images
#
cp raw_data/t0412_000.jpg registered/r_0412_000.jpg

matlab65 -nojvm
clear all

  base = imread('registered/r_0412_000.jpg');
  load registered/rmask_3
  mask = double(m3 == 0);

  npts = sum(sum(mask));
  broi = ones(npts,3);
  sb = size(base);
  n = 1;
  for i=1:sb(1)
    for j=1:sb(2)
      if mask(i,j),
        broi(n,:) = [ base(i,j,1) base(i,j,2) base(i,j,3) ];
        n = n + 1;
      end
    end
  end

  x = [ 0 linspace(10,250,4) 255 ];
  xi = [ 0:255 ];
  for i=1:61
    itinfo = sprintf('i = %3d', i);
    disp(itinfo);

    file = sprintf('registered/r_0412_%03d.jpg', i);
    all = imread(file);

    sizeb = sb;
    roi = ones(npts,3);
    n = 1;
    for ii=1:sb(1)
      for jj=1:sb(2)
        if mask(ii,jj),
          roi(n,:) = [ all(ii,jj,1) all(ii,jj,2) all(ii,jj,3) ];
          n = n + 1;
        end
      end
    end
    b = zeros(3,n);
    s = zeros(3,n);
    cl = 'rgc';
    ic = 1;
    for ic=1:3,
      B = double( broi(:,ic) );
      S = double( roi(:,ic) );
      idc = [ 0 255 ];
      y = fitcspline(B,S,x);
      yi = interp1(y,x,xi);

      % Insert plots here, if necessary

      call(:,:,ic) = uint8(interp1(x,y,double(all(:,:,ic))));
    end

    fcout = sprintf('corr/c_0412_%03d.jpg', i);
    imwrite(call, fcout, 'Quality', 100);
  end


function [y] = fitcspline(B,S,x)
%FITCSPLINE  Function to used to fit the sum of exponential functions
%   FITCSPLINE is a function that returns a spline that best fits
%   a set of color data.
%
%   [y] = fitcspline(B,S) where B is vector of "base" color values,
%   S is a vector of sequential color data corresponding to B, x is a
%   vector of knot locations and y is a matching set of knot values
%   suitable for use in a spline interpolation function.

nobs = length(B);
if (nobs ~= length(S)),
  disp('ERROR: input vectors must be the same length')
  return
end

nv = length(x) - 2;
ofun = inline('norm(P2 - interp1(P1,[0 x 255],P3), 2)', 3);
opts = optimset;
nm1 = length(x) - 1;
x0 = x(2:nm1);
ys = fminsearch(ofun, x0, opts, x,B,S);
y = [0 ys 255];


function [sse] = fit_imtrans(bimg,img,x,mask)
%FIT_IMTRANS Function to fit image transforms for registration
%   FIT_IMTRANS is a function that returns the image registration
%   "goodness of fit" which is calculated as a 2-norm.
%
%   [norm] = fit_imtrans(img,M,fv) where "sse" is the sum-squared
%   difference between the base image "bimg" and the transformed image
%   calculated using the second image "img" and a 3x1 "x" vector that
%   is used to define an affine transform.

sbimg = size(bimg);
simg = size(img);
smask = size(mask);
ck = max(abs(simg-sbimg));
if (ck > 0),
  disp('ERROR: the sizes of bimg and img must match')
  return
end
ck = max(abs(smask-sbimg(1:2)));
if (ck > 0),
  disp('ERROR: mask size must equal one layer of the images')
  return
end
ms = prod(size(x));
if (ms ~= 3),
  disp('ERROR: the length of the x vector must be 3')
  return
end

U = [ 0 0 ; 1 0 ; 0 1 ];
M = U;
phi = x(3)*(pi/180);
rot = [ cos(phi) sin(phi) ; cos(phi+pi/2) sin(phi+pi/2)];
M(2:3,:) = M(2:3,:)*rot;
M(:,1) = M(:,1) + x(1);
M(:,2) = M(:,2) + x(2);

T = maketform('affine', U, M);
dit = imtransform(img, T, ...
                  'XData', [1 sbimg(2)], 'YData', [1 sbimg(1)], ...
                  'FillValues', 0);

%  ub = uint8(bimg);
%  ud = uint8(dit);
%  figure(1), image(ub), axis image
%  figure(2), image(ud), axis image

sse = 0;
for k=1:sbimg(3)
  diff = (dit(:,:,k) - bimg(:,:,k)).*mask;
  sse = sse + norm(diff, 2);
```

```
end

curr = sprintf('  %12.5f', x, sse);
disp(curr)



function [C] = fitsef(A,B)
%FITSEM  Function to used to fit the sum of exponential functions
%   FITSEF is a function that returns the sum of n exponential
%   functions.
%
%   [C] = fitsef(A,B) where A is vector of parameters, B is a
%   vector of observed data, and C is a vector of function results
%   matching the observation locations (B)

n = length(A);
if (mod(n,2) ~= 1) || (n < 3),
  disp('ERROR: The number of parameters must be odd and >=3.')
  return
end

m = length(B);
C = zeros(size(B));
coefs = [2:2:n];
for i=1:m,
  tot = 0.0;
  t = B(i) - abs(A(1));
  if t < 0.0,
    tot = tot + sum(A(coefs));
  else
    for j=coefs,
      tot = tot + A(j)*exp(-abs(A(j+1))*t);
    end
  end
  C(i) = tot;
end
```

Appendix E: Concentration Calculation Code

```
  %% COPY BASE IMAGE OVER
  cp ./registered/r_0412_000.jpg ./corr/c_0412_000.jpg

  %% USE GIMP TO FIND CENTER OF TANK AND RADII
  gimp ./corr/c_0412_000.jpg

  matlab65 -nojvm
  rmin = 72;
  rmax = 165;
  xc = 318;
  yc = 253;


  %  Create Calibration curves from calibration phase:
  mi = zeros(3,256);
  st(1,:) = [ 0.2807  217.2397    0.1164   35.7772    4.8698 ];
  st(2,:) = [ 0.0382   76.4845   -0.7256  178.3388    6.2881 ];
  st(3,:) = [ 0.1466  148.7901   -0.1166  102.6139    3.8662 ];
  for ic=1:3
    x = linspace(abs(st(ic,1)), 4.1, 200);
    y = fitsef(st(ic,:),x);
    yi = [0:255];
    mi(ic,:) = interp1(y,x, yi);
  end
  for ic=1:3
    mimin = min(mi(ic,:));
    for j=240:256
      if not(mi(ic,j) > 0),
        % mi(ic,j) = mimin;
        mi(ic,j) = 0.0;
      end
    end
  end
  smi = prod(size(mi));
  for i=1:smi
    if not(mi(i) > 0)
      mi(i) = 11;
    end
  end

  %%USE GREEN(2) CHANNEL ONLY
  mi(2,256) = 0;
  mi(2,1:5) = 4.0;

  baseimage = imread('./corr/c_0412_000.jpg');
  basemass = base_azimuth(xc,yc,rmax,rmin,baseimage,mi);

  %%*** GET SIZE OF BASEMASS FOR BELOW NUMBERS
  finalresult(:,:,1) = zeros(2,69260);

  for i=1:61
    imagestr = sprintf('./corr/c_0412_%03d.jpg',i)
    image = imread(imagestr);
    result = azimuth(xc,yc,rmax,rmin,image,mi,basemass);
    finalresult(:,:,i) = result;

  end

 save masses finalresult

%*******************************************

C=zeros(2,94,61);
for k=1:61
  disp(k)
  for i=1:69260
    roundrad = round(finalresult(1,i,k));
    if(finalresult(2,i,k)>.044)
      C(1,roundrad-71,k) = C(1,roundrad-71,k)+(finalresult(2,i,k)/(2*pi*roundrad)
);
    end
    C(2,roundrad-71,k) = roundrad;
  end
end

%***************************
```

```
%%Non-dimensionalize the data

cfactor = max(max(C(1,:,:)));
rfactor = 165;

for k=1:61
 for j=1:94
   Cdim(1,j,k) = C(1,j,k)/cfactor;
   Cdim(2,j,k) = C(2,j,k)/rfactor;
 end
end

save concen C Cdim
```

```
function [result] = azimuth(xc,yc,rmax,rmin,pic,mi,basemass)
  count = 1;

  for i = 1:640
    for j = 1:482
      r = sqrt((i-xc)^2+(j-yc)^2);
      if ((r<=rmax)&&(r>rmin))
        pixelval = pic(j,i,2);
        pixelval = double(pixelval) + 1;
        pixelval = int16(pixelval);
        mass = mi(2,pixelval);
        result(1,count) = r;
        result(2,count) = mass-basemass(2,count);
        if result(2,count) < 0
          result(2,count) = 0;
        end
        count = count+1;
      end
    end
  end
```

```
function [result] = azimuth(xc,yc,rmax,rmin,pic,mi)
  count = 1;

  for i = 1:640
    for j = 1:482
      r = sqrt((i-xc)^2+(j-yc)^2);
      if ((r<=rmax)&&(r>rmin))
        pixelval = pic(j,i,2);
        pixelval = double(pixelval) + 1;
        pixelval = int16(pixelval);
        mass = mi(2,pixelval);
        result(1,count) = r;
        result(2,count) = mass;
        count = count+1;
      end
    end
  end
```

Appendix F: Diffusion Simulator and Optimizer Code

```
%% BEGIN SIMULATION CODE


matlab65 -nojvm
clear all
load concen

count=1;
for t=1:61
  for x=1:94
    xtc(count,1) = C(2,x,t);
    xtc(count,2) = t;
    xtc(count,3) = C(1,x,t);
    count=count+1;
  end;
end;

%----------------------

d=[30 30 30];
xstart=72;
xend=165;
nx=94;
dt=.0001;


D=Interp_D(d,xstart,xend,nx);
A = Build_A(D, xstart, xend, dt);
Csim = Run_Sim(A,xtc,dt);


%----------------------

%optimization
ofun = inline('Compute_Norm(x,P1,P2,P3,P4,P5)', 5)
opts = optimset;
x0 = d;
P1=xstart;
P2=xend;
P3=nx;
P4=dt;
P5=xtc;
fminsearch(ofun, x0, opts, P1, P2, P3, P4, P5)


function norm1 = Compute_Norm(d,xstart,xend,nx,dt,xtc)
%Compute_Norm Function to compute the norm of
%             simulated C and observed C
%   Compute_Norm ...

d = [0 d 0];
D=Interp_D(d,xstart,xend,nx);
A = Build_A(D, xstart, xend, dt);
Csim = Run_Sim(A,xtc,dt);

norm1 = norm(Csim(:,3)-xtc(:,3));



%%****************************
%%
%% Function Build_A
%% Constructs the matrix A for
%% the diffusion equation solver
%%
%% Author: Cliff Hodges
%% Date: 3/31/04
%%
%%****************************


function [A] = Build_A(D, xstart, xend, dt)

  % basic sanity checks
  ck = prod(size(dt));
  if (ck ~= 1),
    disp('ERROR: dt must be a scalar')
    return
  end
  ck = prod(size(xstart));
  if (ck ~= 1),
    disp('ERROR: xstart must be a scalar')
    return
  end
  ck = prod(size(xend));
  if (ck ~= 1),
    disp('ERROR: xend must be a scalar')
    return
  end

  nx = prod(size(D))-1;
  dr = (xend-xstart)/nx;
  for i=1:nx+1
    r(i)=xstart+((i-1)*dr);
  end;

  ri(1)=r(1);
  for i=2:((2*prod(size(r)))-1)
     ri(i) = ri(i-1)+(dr/2);
  end;

  Di = interp1(r,D,ri);
  Di(prod(size(Di)))=D(prod(size(D)));

  Di = [0 0 Di 0 0];
  ri = [(ri(1)-dr) (ri(1)-(dr/2)) ri (ri(prod(size(ri)))+(dr/2)) (ri(prod(size(ri
)))+dr)];

  % Create the first row of A
  LA(1)=0;
  DA(1)=Dj(dt,3,ri,Di);
  UA(1)=Uj(dt,3,ri,Di);

  % Create rows 2 thru nx-1 of A
  for j=2:(nx-1)
    LA(j)=Lj(dt,((2*j)+1),ri,Di);
    DA(j)=Dj(dt,((2*j)+1),ri,Di);
    UA(j)=Uj(dt,((2*j)+1),ri,Di);
  end;

  % Create bottom row of A
  LA(nx)=Lj(dt,((2*nx)+3),ri,Di);
  DA(nx)=Dj(dt,((2*nx)+3),ri,Di);
  UA(nx)=0;

  % return A
  A = [LA' DA' UA'];


%%end function Build_A



%%****************************************
%% HELPER FUNCTIONS to calculate individual
%% matrix element values
%%****************************************

%%function Lj

function Lj = Lj(dt,j,r,D)
  Lj=(-dt/(r(j)*(r(j+1)-r(j-1))))*((-r(j-1)*D(j-1))/(r(j)-r(j-2)));

%%end function Lj

%%****************************************

%%function Dj

function Dj = Dj(dt,j,r,D)
  Dj = 1+(-dt/(r(j)*(r(j+1)-r(j-1))))*(((r(j+1)*D(j+1))/(r(j+2)-r(j)))+((r(j-1)*D
(j-1))/(r(j)-r(j-2))));


%%end function Dj

%%****************************************
```

```
%%function Uj

function Uj = Uj(dt,j,r,D)
  Uj=(-dt/(r(j)*(r(j+1)-r(j-1))))*((-r(j+1)*D(j+1))/(r(j+2)-r(j)));

%%end function Uj



function [D] = Interp_D(d, xstart, xend, nx)
%Interp_D Function to assemble the D vector (dispersion values)
%    BUILD_D ...

% basic sanity checks
ck = prod(size(xstart));
if (ck ~= 1),
  disp('ERROR: xstart must be a scalar')
  return
end
ck = prod(size(xend));
if (ck ~= 1),
  disp('ERROR: xend must be a scalar')
  return
end
ck = prod(size(nx));
if (ck ~= 1),
  disp('ERROR: nx must be a scalar')
  return
end

nd = prod(size(d));
dx=((xend-xstart)/nx);
dd=((xend-xstart)/(nd-1));

for i=1:nd
  r(i)=xstart+((i-1)*dd);
end;

for i=1:nx+1
  ri(i)=xstart+((i-1)*dx);
end;

D = interp1(r,d,ri);



function [Csim] = Run_Sim(A,xtc,dt)
%Run_Sim Function to create the simulated C values
%    Run_Sim ...

%fill in first time step into Csim
Csim(1,:)=xtc(1,:);
test=xtc(1,2);
i=2;
while (xtc(i,2)==test)
  Csim(i,:)=xtc(i,:);
  i=i+1;
end;

xstart=Csim(1,1);
xend=Csim(i-1,1);
nx=i-1;

next_time=xtc(nx+1,2);
next_time_index=nx+1;

%create full A matrix from stored A
Afull(1,:) = [A(1,2) A(1,3) zeros(1,nx-2)];
for k=2:nx-1
  Afull(k,:) = [zeros(1,k-2) A(k,1) A(k,2) A(k,3) zeros(1,nx-k-1)];
end;
Afull(nx,:) = [zeros(1,nx-2) A(nx,1) A(nx,2)];

C=Csim(:,3);
for i=2:max(xtc(:,2))

  for q=1:(1/dt)
    C=Afull*C;
  end;
  if i==next_time
    %fill in Csim
    ttemp = zeros(1,nx);
    ttemp = ttemp';
    ttemp = ttemp+next_time;
    Ctemp = [xtc(1:nx,1) ttemp  C];
    Csim = cat(1,Csim,Ctemp);

    %update next_time
    if i ~= max(xtc(:,2))
      next_time_index=next_time_index+nx;
      next_time = xtc(next_time_index,2);
    end;
  end;
end;
```