**Expected Results (Deliverables)**

The results of a project that seeks to reduce error in an ocean model are the amount of error reduced in those models. $J(x)$, the cost function we are seeking to reduce, is a function of many variables. We are able to control a single variable in that cost function, $x_{obs}$ (a vector that contains observed values, including salinity, temperature, current velocities, etc.)

**Equation:**

$$J(x) = (x - x_{obs})^T \cdot O^{-1} \cdot (x - x_{obs}) + (x - x_b)^T \cdot B^{-1} \cdot (x - x_b)$$

**Hypothesis**

The hypothesis for this work is less of a question than a statement: Determine the method (algorithm) that best reduces the cost function, $J(x)$, in a given amount of time (i.e., we can only obtain a certain number of $x_{obs}$).

**Independent and dependent variables**

Of the variables in $J(x)$ above, we can collect and observe $x_{obs}$. The O and B represent covariance matrices – essentially a multiplier or weight of either the observed or the background data, respectively. The variable $x_b$ is background data – preexisting data that goes into the first iteration of an ocean model simulation.

**Measures**

I plan to measure the results of my experiments by comparing the total amount of error reduced in each simulation by grabbing $x_{obs}$ in different places. Because we cannot grab every $x$ in a simulation (a simulation consists of thousands of potential data points, each with a certain amount of error), we have to create a path that collects the $x_{obs}$ in the most efficient manner and in a manner that produces the best results.

**Experiment Protocol**

Because the initial data ($x_b$) for the experiment is available, I plan on conducting the experiment by writing a Java program that searches for the best path to reduce error. This can be executed by using offline search methods (depth and breadth first), and online search methods (best first, $A^*$), and examining the paths that produce the best results.