Modeling the Metabolism of Ethanol to More Accurately Predict Blood Alcohol Content

New Mexico Adventures in
Supercomputing Challenge
Final Report
April 2, 2003

Team 009
Alamogordo High School

Levi Blackstone
Matthew Woller

Mr. Simon

Kevin Blackstone

Table of Contents:

Executive Summary:

The purpose of this project was to create a mathematical model to simulate the removal of alcohol from the human body in order to obtain a blood alcohol content (BAC) prediction. An accurate prediction of BAC would be useful in a variety of fields. Lawyers, police workers, scientists, and the average person could use such predictions in many different circumstances.

Compartmental modeling is a method of mathematical simulation where various parts of a system are represented as a "compartment" and equations are used to model transfer between compartments. It was hypothesized that a compartmental model representing the stomach, small intestine, and lean body mass could provide a more accurate estimate of BAC.

A computer program written in C++ was developed to implement the compartmental model and compare the accuracy of the predictions to those of currently existing models. Experimental data was obtained from the Texas Transportation Institute as a benchmark for the accuracy of the predictions. It was found that using this compartmental model, a more accurate prediction could be achieved in male subjects. Currently, the predictions for female subjects have not improved significantly. The model will continue to be modified to more accurately represent the process of alcohol elimination based on current alcohol research.

Introduction:

Predicting one's blood alcohol content (BAC) and the elimination rate of alcohol could be useful in a variety of circumstances. Individuals wishing to estimate their blood alcohol content after a certain time could benefit from such a prediction as well as professionals such as bartenders, police officers and lawyers. Perhaps a more important application would be to scientists investigating the effects of alcohol on human health by providing a guide for effective dosing methods.

Several general models for BAC predictions are currently used by the Traffic Department that use an average blood alcohol curve to estimate ethanol elimination. These models do not provide an accurate prediction in a majority of circumstances because of variations in a number of factors for each individual. Most general models merely account for the body weight and number of drinks consumed by the individual, ignoring individual height, weight, age, gender, drinking history, amount of food in the stomach and a host of other variables affecting the accuracy of the prediction. These models also use a general elimination curve to calculate alcohol elimination instead of modeling the process of absorption and elimination. Using the overall body mass in predictions creates especially inaccurate results because alcohol is only distributed in the lean body mass. People with a higher percentage of body fat reach higher blood alcohol contents after consuming the same dose of alcohol than people with a lower percentage of fat.

Two recently developed computer models (cBAC; Addiction Research Foundation, London, Ontario, Canada, 1991; and BACest; National Highway Traffic Safety Administration, Washington, DC, 1994) are used to predict BAC. The cBAC model uses height, weight, gender and (for men only) age to estimate total body water (TBW) and the BACest program uses only body weight and gender as variables, using a separate percentage of body weight for men and women to determine TBW. In a recent study, Davies and Bowen (2000) tested these models to determine the accuracy of the predicted peak BAC's compared to actual experimental data. They found that each model seriously underestimated the peak BAC's for their group of test subjects.

The process of ethanol metabolism can be modeled to provide more accurate estimates of blood alcohol content over time. Ingested ethanol is mainly metabolized by the enzyme alcohol dehydrogenase (ADH) which uses NAD(H) as its coenzyme. After ingestion, ADH acts upon the ethanol, converting it to acetaldehyde, which is acted upon by acetaldehyde dehydrogenase (ALDH) to form acetate. The acetate is further metabolized into carbon dioxide and water. The rate limiting factor for ethanol metabolism is the dissociation of the NADH-ADH enzyme complex. ADH causes the oxidation of ethanol to reduce NAD, meaning the rate of ethanol elimination is determined by the liver's capacity to re-oxidize NADH. Two other minor pathways exist for the metabolism of ethanol to acetaldehyde; the microsomal ethanol oxidizing system (MEOS) and the catalase pathway.

The purpose of this project was to develop a model predicting BAC that simulates the ethanol metabolism process and accounts for a variety of variables in an attempt to predict BAC more accurately than previous models. It was hypothesized that a compartmental model simulating the process of ethanol metabolism could be developed that would provide more accurate predictions of BAC than currently existing models.

Materials and Methods:

A review of current literature on ethanol pharmokinetics was conducted. It was learned from Kapur (1991) that most existing BAC prediction models are based on Widmark's (1932) mathematical equation. It was also learned that Watson *et al.* (1980) had developed regression equations to calculate total body water (TBW) to make an accurate estimation of the actual distribution volume of alcohol for each individual, this being the most notable update to Widmark's work. Pieters *et al.* (1990) developed a three-compartment model (stomach, small intestine, lean body mass) to simulate ethanol metabolism. It was assumed that first-pass metabolism was insignificant and that ethanol was released from the stomach at an apparent first-order rate, modified by a feedback variable. This model assumed that ethanol absorption from the small intestine followed a first-order rate and that ethanol metabolism from the lean body mass follows Michaelis-Menton kinetics.

A new compartmental model based on this information was developed. The rate of passage from the stomach to the small intestine was assumed to be zero-order and a rate constant was assigned based on evaluation of data obtained from Becky Davies at the Texas Transportation Institution. The rate of ethanol absorption from the small intestine into the lean body mass was assumed to be first-order and a new constant was assigned to this rate. New rate constants were assigned to the Michaelis-Menton equation used to model the elimination of ethanol from the lean body mass. Watson *et al.* TBW equations were used in this model.

A computer program was developed to test this new model in an efficient fashion and allow a user to input values for each subject to be tested. The results were analyzed to estimate errors associated with the prediction model using the root mean squared method.

Results:

These tables contain the data output from the program using the compartmental model. The data is represented graphically in Appendices A and B. A line was plotted to show positive or negative deviation from the true blood alcohol content for each subject; data appearing above the line were underestimated and data appearing below the line were overestimated.

OBSERVED VALUES VS. VALUES PREDICTED WITH NEW MODEL

sex	obs. BAC	obs. TBW	pred. BAC	pred. TBW
M	0.064	50.000	0.051	48.997
M	0.068	44.900	0.066	43.671
M	0.053	42.600	0.066	43.652
M	0.023	59.600	0.039	56.212
M	0.047	49.700	0.052	48.740
M	0.078	42.200	0.069	42.851
M	0.064	43.100	0.070	42.644
M	0.048	49.000	0.047	51.028
M	0.058	50.800	0.053	48.342
M	0.062	45.400	0.054	47.805
M	0.057	50.300	0.050	49.400
M	0.061	45.400	0.058	46.241
M	0.058	44.900	0.069	42.735
M	0.073	45.400	0.056	46.924
M	0.066	45.400	0.059	45.814
M	0.076	35.400	0.088	38.379
M	0.030	58.900	0.046	51.655
M	0.054	48.100	0.055	47.346
M	0.058	48.500	0.052	48.845
M	0.065	41.300	0.061	45.368
M	0.053	46.700	0.060	45.604
M	0.072	45.300	0.063	44.730
M	0.051	55.300	0.043	53.356
M	0.051	47.200	0.061	45.208
M	0.057	51.300	0.050	49.584
M	0.064	45.800	0.063	44.687
M	0.073	45.600	0.054	47.726
M	0.078	47.600	0.058	46.192
M	0.074	46.700	0.059	45.970
M	0.064	49.400	0.052	48.686
M	0.076	45.400	0.066	43.803
M	0.054	51.700	0.044	52.726
M	0.040	55.800	0.039	55.688
M	0.079	36.300	0.068	43.055
M	0.050	59.000	0.038	56.515
M	0.040	61.200	0.034	60.090
M	0.047	45.800	0.053	48.317
M	0.059	46.700	0.073	41.831
M	0.056	55.800	0.045	52.150
M	0.033	55.300	0.039	56.129
M	0.065	41.500	0.070	42.530
M	0.057	46.200	0.058	46.155
M	0.079	40.900	0.078	40.578
M	0.058	49.500	0.049	49.895
M	0.064	46.200	0.057	46.787
M	0.058	43.100	0.068	43.003
M	0.065	39.000	0.090	38.035
M	0.060	46.900	0.055	47.626
M	0.050	43.400	0.068	43.005
M	0.047	47.200	0.055	47.434
M	0.053	45.400	0.051	49.217
M	0.058	41.300	0.072	41.895
M	0.055	45.200	0.065	43.938
M	0.063	40.200	0.084	39.319
M	0.063	44.700	0.062	45.074
M	0.065	46.500	0.057	46.753
M	0.066	51.000	0.042	53.963
F	0.068	33.600	0.071	33.675

F	0.098	25.400	0.122	25.989
F	0.081	30.400	0.077	32.384
F	0.087	32.600	0.075	32.853
F	0.069	28.600	0.100	28.636
F	0.075	31.300	0.096	29.104
F	0.070	33.600	0.092	29.762
F	0.098	31.300	0.088	30.420
F	0.091	31.300	0.094	29.392
F	0.067	33.100	0.077	32.377
F	0.082	28.100	0.100	28.529
F	0.059	38.100	0.053	38.944
F	0.082	29.000	0.095	29.302
F	0.085	26.300	0.121	26.104
F	0.074	29.900	0.092	29.713
F	0.081	27.700	0.105	27.871
F	0.084	31.800	0.091	29.864
F	0.100	30.800	0.086	30.749
F	0.073	34.500	0.071	33.765
F	0.098	28.600	0.100	28.513
F	0.076	28.100	0.097	28.965
F	0.074	33.600	0.083	31.168
F	0.087	31.300	0.076	32.598
F	0.087	30.400	0.087	30.473
F	0.098	28.100	0.106	27.773
F	0.087	29.500	0.104	28.093
F	0.116	28.600	0.091	29.844
F	0.057	41.700	0.051	39.544
F	0.076	33.600	0.071	33.699
F	0.094	24.000	0.126	25.561
F	0.068	34.900	0.064	35.541
F	0.066	36.300	0.056	37.793
F	0.064	32.600	0.072	33.461
F	0.100	33.100	0.078	32.187
F	0.073	32.700	0.060	36.486
F	0.078	32.200	0.069	34.118
F	0.097	28.600	0.092	29.770
F	0.072	31.300	0.080	31.776
F	0.060	34.900	0.061	36.322
F	0.105	24.900	0.114	26.885
F	0.111	29.200	0.082	31.497
F	0.098	32.800	0.072	33.420
F	0.104	30.600	0.082	31.455
F	0.086	31.000	0.075	32.886
F	0.068	37.400	0.053	38.903
F	0.104	27.200	0.107	27.658
F	0.076	37.400	0.050	39.815
F	0.092	32.500	0.068	34.365
F	0.086	31.500	0.070	33.888
F	0.097	28.900	0.095	29.334
F	0.112	28.900	0.081	31.669
F	0.093	31.100	0.073	33.346
F	0.144	25.900	0.117	26.523
F	0.107	30.700	0.085	30.880
F	0.082	33.300	0.063	35.598
F	0.100	30.100	0.079	32.088
F	0.100	30.100	0.079	32.088

ROOT MEAN SQUARED VALUES (rms)
rms TBW men: 2.181 rms TBW women: 1.551
rms BAC men: 0.011 rms BAC women: 0.017

OBSERVED VALUES VS. VALUES PREDICTED WITH OLD MODELS

sex	obs. BAC	cBAC	BACest
M	0.064	0.043	0.040
M	0.068	0.048	0.050
M	0.053	0.048	0.050
M	0.023	0.037	0.030
M	0.047	0.043	0.040
M	0.078	0.049	0.050
M	0.064	0.050	0.050
M	0.048	0.041	0.040
M	0.058	0.043	0.040
M	0.062	0.044	0.040
M	0.057	0.042	0.040
M	0.061	0.046	0.040
M	0.058	0.049	0.050
M	0.073	0.045	0.040
M	0.066	0.046	0.050
M	0.076	0.055	0.060
M	0.030	0.040	0.040
M	0.054	0.044	0.040
M	0.058	0.043	0.040
M	0.065	0.047	0.050
M	0.053	0.045	0.040
M	0.072	0.046	0.040
M	0.051	0.037	0.030
M	0.051	0.045	0.040
M	0.057	0.041	0.040
M	0.064	0.046	0.040
M	0.073	0.043	0.040
M	0.078	0.044	0.040
M	0.074	0.045	0.040
M	0.064	0.042	0.040
M	0.076	0.047	0.040
M	0.054	0.038	0.030
M	0.040	0.035	0.030
M	0.079	0.048	0.050
M	0.050	0.035	0.030
M	0.040	0.032	0.020
M	0.047	0.042	0.040
M	0.059	0.050	0.050
M	0.056	0.030	0.030
M	0.033	0.035	0.030
M	0.065	0.049	0.050
M	0.057	0.044	0.040
M	0.079	0.052	0.050
M	0.058	0.040	0.030
M	0.064	0.044	0.040
M	0.058	0.048	0.040
M	0.065	0.055	0.050
M	0.060	0.043	0.030
M	0.050	0.048	0.040
M	0.047	0.043	0.040
M	0.053	0.041	0.030
M	0.058	0.050	0.040
M	0.055	0.047	0.040
M	0.063	0.054	0.050
M	0.063	0.046	0.040
M	0.065	0.044	0.040
M	0.066	0.037	0.030

F	0.068	0.064	0.060
F	0.098	0.084	0.100
F	0.081	0.067	0.060
F	0.087	0.066	0.060
F	0.069	0.076	0.080
F	0.075	0.074	0.080
F	0.070	0.073	0.080
F	0.098	0.071	0.070
F	0.091	0.074	0.080
F	0.067	0.067	0.070
F	0.082	0.077	0.080
F	0.059	0.055	0.050
F	0.082	0.074	0.080
F	0.085	0.083	0.100
F	0.074	0.073	0.080
F	0.081	0.078	0.080
F	0.084	0.072	0.070
F	0.100	0.070	0.070
F	0.073	0.064	0.060
F	0.098	0.076	0.080
F	0.076	0.075	0.080
F	0.074	0.069	0.070
F	0.087	0.066	0.070
F	0.087	0.071	0.070
F	0.098	0.079	0.080
F	0.087	0.077	0.080
F	0.116	0.073	0.080
F	0.057	0.053	0.040
F	0.076	0.063	0.060
F	0.094	0.087	0.100
F	0.068	0.060	0.050
F	0.066	0.056	0.050
F	0.064	0.064	0.060
F	0.100	0.067	0.070
F	0.073	0.058	0.050
F	0.078	0.063	0.060
F	0.097	0.073	0.070
F	0.072	0.068	0.060
F	0.060	0.058	0.050
F	0.105	0.081	0.090
F	0.111	0.069	0.070
F	0.098	0.064	0.060
F	0.104	0.069	0.070
F	0.086	0.065	0.060
F	0.068	0.054	0.040
F	0.104	0.081	0.090
F	0.076	0.053	0.040
F	0.092	0.062	0.050
F	0.086	0.063	0.060
F	0.097	0.074	0.070
F	0.112	0.068	0.070
F	0.093	0.064	0.060
F	0.144	0.083	0.100
F	0.107	0.070	0.070
F	0.082	0.060	0.050
F	0.100	0.067	0.060
F	0.100	0.067	0.060

ROOT MEAN SQUARED VALUES (rms)
rms BAC men cBAC: 0.013 rms BAC women cBAC: 0.016
rms BAC men BACest: 0.015 rms BAC women BACest: 0.017

Discussion:

The results support the hypothesis that a compartmental model can be used to more accurately predict BAC. The root mean squared analysis of the data showed that the accuracy of the predictions made with the new model was better in male subjects and remained about the same for female subjects. A trend of underestimation occurred in females under concentrations of approximately 0.09 while overestimation occurred in concentrations above 0.09. The male predictions tended to slightly underestimate until concentrations above 0.07, when the predictions began a trend of overestimation. These results are slightly more accurate compared to the predictions from the cBAC and BACest programs by Davies and Bowen (2000).

Recommendations:

To further improve the accuracy of the predictions of the new model, several additional factors will be accounted for. First-pass metabolism will be considered using another Michaelis-Menton equation in the stomach. The zero-order constant in the stomach will be changed to a modified first-order rate accounting for alcohol concentration remaining in the stomach and food in the stomach.

Acknowledgements:

Thank you to Kevin Blackstone for his assistance in obtaining research and for suggestions on all biological aspects of the project. Many thanks to Becky Davies from the Texas Transportation Institute for providing experimental data used to test the accuracy of our predictions. Thanks to Raul Cruz for suggestions regarding statistical analysis of the results. Thanks to the AiS preliminary judges for many helpful suggestions on our presentation. Thank you to Colleen Teske for explanations of integral calculus.

References:

B.T. Davies, M.A., and C.K. Bowen, M.S., Peak Blood Alcohol Prediction: An Empirical Test of Two Computer Models. J Alc Stud. 2000; 61:1

B.M. Kapur, Alcohol: Pharmacology, methods of analysis and its presence in the casualty room, In Drinking and Casualties: Accidents, poisonings and violence in an international perspective, Eds: N Giesbrecht, R Gonzalez, M. Grant et al., Tavistock/Routledge, London/New York, 1991; pp 172-187.

J. E. Pieters, Wedel, M. and Schaafsma, G. J. (1990) Parameter estimation in a three-compartment model for blood alcohol curves. *Alcohol and Alcoholism* 25, 17-24.

P.E. Watson, I.D. Watson and R.D. Batt, Prediction of blood alcohol concentrations in human subjects. Updating the Widmark equation. J Alc Stud. 1981; 42:547-556.

E.M.P. Widmark, Die theoritischen Grundlagen und die praktische Verwendbarkeit der gerichtlichmedizinischen Alkoholbestimmung. Berlin; Urban & Schwarzenberg; 1932. (Widmark EM. in Principles and applications of medicolegal alcohol determination; Translated from 1932 german edition, Biomedical Publications, Davis, CA. 1981.

```
/*
    File name: BACDRIVER.cpp
    Last modified: 03/14/03
    Authors: Levi Blackstone and Matthew Woller
    Copyright (c) 2003
*/

#include <iomanip.h>
#include <fstream.h>
#include <math.h>
#include <stdlib.h>

int numofsubs = 0;
const float pct_alcohol_beer = 0.06; //Percent of alcohol per unit of beer
const float pct_alcohol_wine = 0.10; //Percent of alcohol per unit of wine
const float pct_alcohol_hard = 0.48; //Percent of alcohol per unit of hard
liquor
const float hour_decrease_average = .017;//Hourly decrease in BAC for average
drinker
const float hour_decrease_above_average = .020;//Hourly decrease in BAC for
above average drinker
const float hour_decrease_below_average = .012;//Hourly decrease in BAC for
below average drinker
const float t = 0.01666666; //Time step constant (equivalent to one minute)
float g; //Rate constant (l/hr)
float V; //Maximum elimination rate when ethanol-oxidizing enzyme is
saturated (g/l * hr)
float K; //Ethanol concentration at which the elimination rate is 1/2 V (g/l)
int numofinputsubsmen = 0;
int numofinputsubswomen = 0;

struct subject {
     double h_in_in; //height in inches --- REQUIRED!
     double h_in_cm; //height in centimeters --- REQUIRED!
     double w_in_lbs; //weight in pounds --- REQUIRED!
     double w_in_kg; //weight in kilograms --- REQUIRED!
     int age_in_yrs; //age in years --- REQUIRED!
     char sex_MF; //sex - m = Male,f = Female --- REQUIRED!
     int num_of_drinks; //number of drinks consumed --- REQUIRED!
     char type_of_drink; //Type of drink(s) consumed --- MORE ACCURATE
RESULTS IF INCLUDED
     /* Insert Drinking history variable here!!!!! */
     float hours_since_eat; //hours since the subject last ate --- MORE
ACCURATE REUSLTS IF INCLUDED
     float hours_start_drinking; //hours since the subject started drinking
     float hours_stop_drinking; //hours since the subject stopped drinking
     float total_drinking_hours; //total hours that the subject had been
drinking.
     /* Insert disease varible here if possible */
     /* Insert time of day variable here if possible */
     double final_BAC_ML; //Final attained Blood Alcohol Content for our
program
     double final_BAC_TD; //Final attained Blood Alcohol Content for Traffic
Department Program
     double TBW; //Total body water of subject
     double TBW_TD; //Total body water of subject - Traffic Department
```

```
     double instant_BAC; //Theoretical instantaneous Blood Alcohol Content
for our program
     double instant_BAC_TD; //Theoretical instantaneous Blood Alcohol Content
for Traffic Department Program
     int drink_history_TD; //Drinking history of the subject (1 = Below
Average, 2 = Average, 3 = Above Average)
     float hourly_decrease_TD; //Hourly decrease associated with drinking
history
     double initial_ethanol_concentration; //Initial ethanol concentration in
stomach (g/l)
     double current_ethanol_concentration_stomach; //Current ethanol
concentration in stomach (g/l)
     double current_ethanol_concentration_intestine; //Current ethanol
concentration in small intestine (g/l)
     double current_ethanol_concentration_body; //Current ethanol
concentration in lean body mass (g/l)
     double stomach_to_intestine_rate; //Rate ethanol is moving from stomach
to intestine
     double intestine_to_body_rate; //Rate ethanol is moving from intestine
to body
     double net_stomach_rate; //Net rate of ethanol exchange for stomach
(g/hr)
     double net_intestine_rate; //Net rate of ethanol exchange for intestine
(g/hr)
     double net_body_rate; //Net rate of ethanol exchange for lean body mass
(g/hr)
     double grams_ethanol_stomach; //Grams of ethanol in the stomach
     double grams_ethanol_intestine; //Grams of ethanol in the small
intestine
     double grams_ethanol_body; //Grams of ethanol in the lean body mass
     double elimination_rate; //Grams of ethanol eliminated from lean body
mass
     double time; //Amount of time elapsed in compartmental model
     double peak_BAC; //Peak blood alcohol content in the body compartment
     float observed_TBW;
     double observed_BAC;
     double difference_TBW_M;
     double difference_TBW_F;
     double difference_BAC_F;
     double difference_BAC_M;
     double sigma_TBW_M;
     double sigma_TBW_F;
     double sigma_BAC_M;
     double sigma_BAC_F;
     double BAC_cBAC;
     double BAC_BACest;
     double difference_BAC_F_cBAC;
     double difference_BAC_M_cBAC;
     double difference_BAC_F_BACest;
     double difference_BAC_M_BACest;
     double sigma_BAC_F_cBAC;
     double sigma_BAC_M_cBAC;
     double sigma_BAC_F_BACest;
     double sigma_BAC_M_BACest;

};
```

```
/*
double lb_to_kg(double lbs){ //This function converts pounds to kilograms
     double kg = lbs / 2.2046;
     return kg;
}

double in_to_cm(double in){ //This function converts inches to centimeters
     double cm = in * 2.54;
     return cm;
}
*/

/*
double pct_to_oz_ethanol_drink(subject &sub){ //This function finds the
number of ounces of ethanol in a drink
     int num_drinks = sub.num_of_drinks;
     double pct;
     double oz;
     if (sub.type_of_drink == 'b'){
          pct = pct_alcohol_beer;
          oz = num_drinks * 12 * pct;
     }
     else if (sub.type_of_drink == 'w'){
          pct = pct_alcohol_wine;
          oz = num_drinks * 10 * pct;
     }
     else if (sub.type_of_drink == 'h'){
          pct = pct_alcohol_hard;
          oz = num_drinks * pct;
     }
     return oz;
}
*/

void TBW(subject &sub){ //This function determines the total body water for
the subject
     double tbw;
     if (sub.sex_MF == 'M'){
          tbw = (-0.09516 * (sub.age_in_yrs)) + (0.1074 * (sub.h_in_cm)) +
(0.3362 * (sub.w_in_kg)) + 2.447;
     }
     else if (sub.sex_MF == 'F'){
          tbw = (0.1069 * (sub.h_in_cm)) + (0.2466 * (sub.w_in_kg)) - 2.097;
     }
     sub.TBW = tbw;
}


/*NEED TO CHECK THE NUMBER FOR GRAMS PER OUNCE
float grams_per_oz_ethanol(subject &sub){ //Calculates the grams of ethanol
     float oz = pct_to_oz_ethanol_drink(sub);
     float grams = oz * 23.36;
     return grams;
}
*/

/*
```

```
float grams_ethanol_per_liter_TBW(subject &sub){ //Calculates concentration
in grams per liter
     float concentration = grams_per_oz_ethanol(sub) / sub.TBW;
     return concentration;
}
*/

void get_sub_info_compartment_quick(subject &sub){

     cout << "\nEnter Age:  ";
     cin >> sub.age_in_yrs;
     cout << "\nEnter Weight:  ";
     cin >> sub.w_in_kg;
     cout << "\nEnter Height:  ";
     cin >> sub.h_in_cm;
     cout << "\nEnter gender:  ";
     cin >> sub.sex_MF;
     cout << "\nEnter grams ethanol comsumed:  ";
     cin >> sub.grams_ethanol_stomach;
     cout << "\n\n";
}


/*
void get_sub_info(subject &sub){//This function gets the information on the
subject.
                                //All time-related values are based on the
original time that the test was administered
     char metric_or_not = 'e';//Allows the user to choose metric or English
System of Measurement
     cout << "\nWill you be using English units or metric units for this
subject(e/m)?\n-->";
     do { //This loop checks to make sure the user selected either (m)etric
or (e)nglish
          cin >> metric_or_not;
          if (metric_or_not == 'e'){
               cout << "\nSex (m/f)?\n-->";
               do {
                    cin >> sub.sex_MF;
                    if ((sub.sex_MF != 'M') && (sub.sex_MF != 'F'))
                         cout << "\nPlease enter a valid sex (m = male, f =
female).\n-->";
               } while ((sub.sex_MF != 'M') && (sub.sex_MF != 'F'));
               cout << "\nAge (years)?\n-->";
               cin >> sub.age_in_yrs;
               cout << "\nHeight (inches)?\n-->";
               cin >> sub.h_in_in;
               sub.h_in_cm = in_to_cm(sub.h_in_in);
               cout << "\nWeight (pounds)?\n-->";
               cin >> sub.w_in_lbs;
               sub.w_in_kg = lb_to_kg(sub.w_in_lbs);
               cout << "\n# of drinks consumed by subject?\n-->";
               cin >> sub.num_of_drinks;
               cout << "\nType of drink(s) consumed by subject (b = beer
[12oz], h = hard [1oz 96 proof shot], or w = wine [8oz])?\n-->";
               do { //This loop checks to make sure the user input a valid
drink type
```

```
                    cin >> sub.type_of_drink;
                    if ((sub.type_of_drink == 'b') || (sub.type_of_drink ==
'w') || (sub.type_of_drink == 'h')){
                         break;
                    }
                    else if ((sub.type_of_drink != 'b') && (sub.type_of_drink
!= 'w') && (sub.type_of_drink != 'h')){
                         cout << "\nPlease enter a valid drink type (----b,
w, h----)\n-->";
                    }
               } while ((sub.type_of_drink != 'b') && (sub.type_of_drink !=
'w') && (sub.type_of_drink != 'h'));
               cout << "\nHours since the subject last ate?\n-->";
               cin >> sub.hours_since_eat;
               cout << "\nHours since the subject started drinking
(approximate)\n-->";
               cin >> sub.hours_start_drinking;
               cout << "\nHours since the subject stopped drinking
(approximate)\n-->";
               do { //This loop checks to make sure the user input a valid
end time
                    cin >> sub.hours_stop_drinking;
                    if ((sub.hours_stop_drinking) >=
(sub.hours_start_drinking))
                         cout << "\nPlease enter a valid end time (----End
time later than start time----)\n-->";
               } while ((sub.hours_stop_drinking) >=
(sub.hours_start_drinking));
               cout << "\nWhat is the estimated drinking history (1 [
infrequent drinker ], 2 [ 4-5 drinks/wk ], 3 [ 6+ drinks/wk ] )?\n-->";
               do {
                    cin >> sub.drink_history_TD;
                    switch (sub.drink_history_TD){
                         case 1: sub.hourly_decrease_TD =
hour_decrease_below_average;
                                 break;
                         case 2: sub.hourly_decrease_TD =
hour_decrease_average;
                                 break;
                         case 3: sub.hourly_decrease_TD =
hour_decrease_above_average;
                                 break;
                         default: cout << "Please enter a valid drinking
history value (1, 2, 3).\n-->";
                    }
               } while ((sub.drink_history_TD != 1) && (sub.drink_history_TD
!= 2) && (sub.drink_history_TD != 3));
               sub.total_drinking_hours = sub.hours_start_drinking -
sub.hours_stop_drinking;
               numofsubs++;
               break;
          }
          else if (metric_or_not == 'M'){

               cout << "\nSex (m/f)?\n-->";
               do {
                    cin >> sub.sex_MF;
```

```
                    if ((sub.sex_MF != 'M') && (sub.sex_MF != 'F'))
                         cout << "\nPlease enter a valid sex (m = male, f =
female).\n-->";
               } while ((sub.sex_MF != 'M') && (sub.sex_MF != 'F'));
               cout << "\nAge (years)?\n-->";
               cin >> sub.age_in_yrs;
               cout << "\nHeight (centimeters)?\n-->";
               cin >> sub.h_in_cm;
               cout << "\nWeight (kilograms)?\n-->";
               cin >> sub.w_in_kg;
               cout << "\n# of drinks consumed by subject?\n-->";
               cin >> sub.num_of_drinks;
               cout << "\nType of drink(s) consumed by subject (b = beer
[12oz],h = hard [1oz 96 proof shot], or w = wine [8oz])?\n-->";
               do { //This loop checks to make sure the user input a valid
drink type
                    cin >> sub.type_of_drink;
                    if ((sub.type_of_drink == 'b') || (sub.type_of_drink ==
'w') || (sub.type_of_drink == 'h')){
                         break;
                    }
                    else if ((sub.type_of_drink != 'b') && (sub.type_of_drink
!= 'w') && (sub.type_of_drink != 'h')){
                         cout << "\nPlease enter a valid drink type (----b,
w, h----)\n-->";
                    }
               } while ((sub.type_of_drink != 'b') && (sub.type_of_drink !=
'w') && (sub.type_of_drink != 'h'));
               cout << "\nHours since the subject last ate?\n-->";
               cin >> sub.hours_since_eat;
               cout << "\nHours since the subject started drinking
(approximate)\n-->";
               cin >> sub.hours_start_drinking;
               cout << "\nHours since the subject stopped drinking
(approximate)\n-->";
               do { //This loop checks to make sure the user input a valid
end time
                    cin >> sub.hours_stop_drinking;
                    if ((sub.hours_stop_drinking) >=
(sub.hours_start_drinking))
                         cout << "\nPlease enter a valid end time (----End
time later than start time----)\n-->";
               } while ((sub.hours_stop_drinking) >=
(sub.hours_start_drinking));
               cout << "\nWhat is the estimated drinking history (1 [
infrequent drinker ], 2 [ 4-5 drinks/wk ], 3 [ 6+ drinks/wk ] )?\n-->";
               do {
                    cin >> sub.drink_history_TD;
                    switch (sub.drink_history_TD){
                         case 1: sub.hourly_decrease_TD =
hour_decrease_below_average;
                                 break;
                         case 2: sub.hourly_decrease_TD =
hour_decrease_average;
                                 break;
                         case 3: sub.hourly_decrease_TD =
hour_decrease_above_average;
```

```
                                break;
                        default: cout << "Please enter a valid drinking
history value (1, 2, 3).\n-->";
                    }
               } while ((sub.drink_history_TD != 1) && (sub.drink_history_TD
!= 2) && (sub.drink_history_TD != 3));
               sub.total_drinking_hours = sub.hours_start_drinking -
sub.hours_stop_drinking;
               numofsubs++;
               break;
          }
          else if ((metric_or_not != 'e') && (metric_or_not != 'M')){
               cout << "\nPlease enter a valid value for English or metric (-
---\"e\" or \"m\"----)\n-->";
          }
     } while ((metric_or_not != 'e') && (metric_or_not != 'M'));
}
*/

/*
void BAC_TD(subject &sub)//This function uses the Traffic Department formula
to calculate BAC
{
     double kg = sub.w_in_kg;//subject weight in kilograms
     double tbw_liters = (sub.sex_MF == 'M')?(kg * .58):(kg * .49);//Total
Body Water in liters
     sub.TBW_TD = tbw_liters;
     double tbw_milliliters = tbw_liters * 1000;//Total Body Water in
milliliters
     float g_1_oz_alcohol = 23.36;//Grams per one ounce of alcohol
     double g_alcohol_per_ml_water = g_1_oz_alcohol / tbw_milliliters;//Grams
alcohol per one milliliter water
     double g_alcohol_per_ml_blood = g_alcohol_per_ml_water * .806;//Grams of
alcohol per one milliliter blood
     double g_alcohol_per_100_ml_blood = g_alcohol_per_ml_blood * 100;//Grams
alcohol per 100 milliliters blood for one ounce of alcohol - Instantaneous
BAC
     double oz_consumed = pct_to_oz_ethanol_drink(sub);//Actual amount of
ethanol consumed in ounces
     double instant_BAC = oz_consumed *
g_alcohol_per_100_ml_blood;//Instantaneous BAC
     sub.instant_BAC_TD = instant_BAC;
     double final_BAC = instant_BAC - (sub.hours_start_drinking *
sub.hourly_decrease_TD);//Final BAC for the subject
     sub.final_BAC_TD = final_BAC;

}
*/

/*
void file_output(char file[], subject &sub[]){// This function outputs
program data to a separate text file file[]
     ofstream out(file);
     out << "\t*****Output file*****\n\n\n";
     for (int x = 0;x < numofsubs;x++){
          out << "\tSubject " << x+1 << "\n\n";
          out.setf(ios::left);
```

```
          out.setf(ios::showpoint|ios::fixed);
          out << setprecision(3);
          out << "Height (cm)     Weight (kg)     Age     Sex     Drink type
\n";
          out << setw(16) << sub[x].h_in_cm << setw(16) << sub[x].w_in_kg <<
setw(8) << sub[x].age_in_yrs << setw(7) << sub[x].sex_MF << "" <<
sub[x].type_of_drink << "\n\n";
          out << "# of Drinks     Last ate (h)     Started Drinking (h)
Stopped Drinking (h)     Total Body Water (kg)\n";
          out << setw(16) << sub[x].num_of_drinks << setw(17) <<
sub[x].hours_since_eat << setw(25) << sub[x].hours_start_drinking << setw(25)
<< sub[x].hours_stop_drinking << setw(25) << sub[x].TBW << "\n\n";
          out << "BAC     (T)raffic (D)ept. BAC     TD TBW     TD Instant BAC
TD Drinking History (1,2,3)     TD Hourly Decrease (BAC/hour)\n";
          out << setw(8) << sub[x].final_BAC_ML << setw(26) <<
sub[x].final_BAC_TD << setw(11) << sub[x].TBW_TD << setw(19) <<
sub[x].instant_BAC_TD << setw(32) << sub[x].drink_history_TD << setw(34) <<
sub[x].hourly_decrease_TD << "\n\n\n\n";
     }
     out << "\n\n\n\t\tAll values are rounded to three decimal places.";
     out.close();
}
*/

/*
void file_input(char file[], subject &sub){//This function inputs subject
data from file file[]
    ifstream in(file);
    int x = 0;
    while (!in.eof()){
        in >> sub[x].h_in_cm >> sub[x].w_in_kg >> sub[x].age_in_yrs >>
sub[x].sex_MF >> sub[x].num_of_drinks >> sub[x].type_of_drink >>
sub[x].hours_since_eat >> sub[x].hours_start_drinking >>
sub[x].hours_stop_drinking >> sub[x].drinking_history_TD;
        x++;
        numofsubs++;
    }
    in.close();
}
*/



/*
void display_TD(subject &sub){
     cout << "Height in cm: " << sub.h_in_cm << "\nWeight in kg: " <<
sub.w_in_kg << "\nAge in years: " << sub.age_in_yrs << "\nSex: " <<
sub.sex_MF << "\nNumber of drinks: " << sub.num_of_drinks << "\nType of
drink: " << sub.type_of_drink << "\nHours since ate: " << sub.hours_since_eat
<< "\nHours ago that subject started drinking: " << sub.hours_start_drinking
<< "\nHours since stopped drinking: " << sub.hours_stop_drinking << "\nTotal
hours drinking: " << sub.total_drinking_hours << "\nTBW: " << sub.TBW_TD <<
"\nDrinking History: " << sub.drink_history_TD << "\nHourly Decrease: " <<
sub.hourly_decrease_TD << "\nTheoretical Instantaneous BAC: " <<
sub.instant_BAC_TD;
     cout.setf(ios::showpoint|ios::fixed);
     cout << setprecision(3);
```

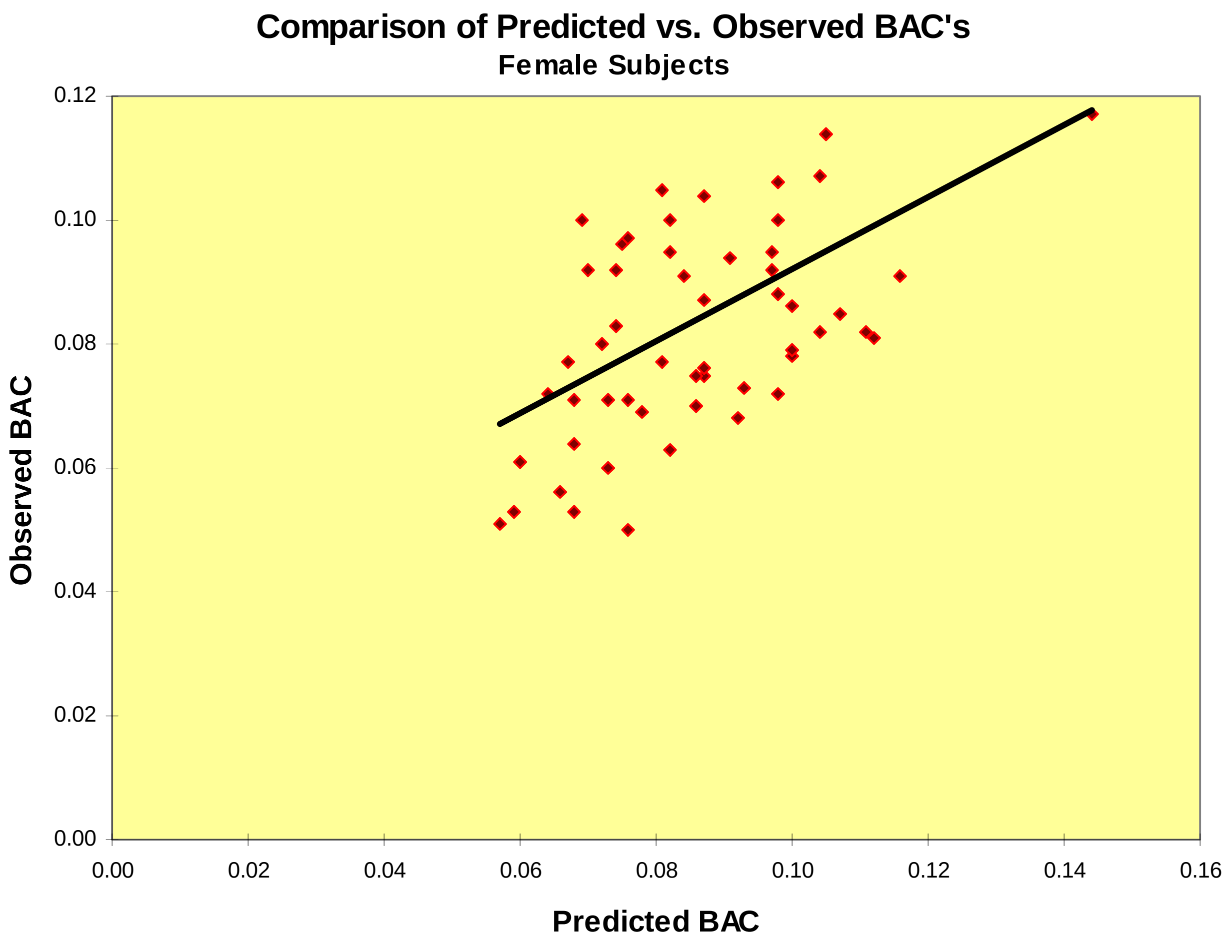
```
	if (sub.final_BAC_TD < 0.0000)
		cout << "\nAll ethanol (theoretically) has been metabolized.\n";
	else
		cout << "\nFinal BAC: " << sub.final_BAC_TD << endl;
}
*/

/*
THIS COMPARTMENTAL MODEL PROVIDES THE NUMBER OF GRAMS ETHANOL PER LITER
*/

/*
void initialize_compartment_model(subject &sub){
	sub.initial_ethanol_concentration = grams_ethanol_per_liter_TBW(sub);
//(Oz. ethanol * 23.36g) / liters of TBW
	sub.current_ethanol_concentration_stomach =
sub.initial_ethanol_concentration; //Initializes stomach concentration
	sub.current_ethanol_concentration_intestine = 0;
	sub.current_ethanol_concentration_body = 0;
	sub.time = 0;
}
*/

void stomach_compartment(subject &sub){//Rate constants taken from Table 2 in
Wedel et al. 1991
	/*
	float k; //Rate constant (l/hr)
	float a; //Feedback control constant (l^2/g^2)

	if (sub.sex_MF == 'M'){
		k = 3.79;
		a = -0.75;
	}
	else if (sub.sex_MF == 'F'){
		k = 3.38;
		a = -.92;
	}
	*/
	//sub.stomach_to_intestine_rate = k *
sub.current_ethanol_concentration_stomach;
	//sub.stomach_to_intestine_rate = k * pow((1 + a *
pow(sub.current_ethanol_concentration_stomach,2)), -1) *
sub.current_ethanol_concentration_stomach;

	if ((sub.grams_ethanol_stomach - (sub.stomach_to_intestine_rate * t)) >
0.00000){
		sub.stomach_to_intestine_rate = 57;
		sub.net_stomach_rate = -sub.stomach_to_intestine_rate;
	}
	else{
		sub.stomach_to_intestine_rate = 0;
		sub.grams_ethanol_stomach = 0.0;
		sub.net_stomach_rate = 0.0;
	}
}

void intestine_compartment(subject &sub){
```

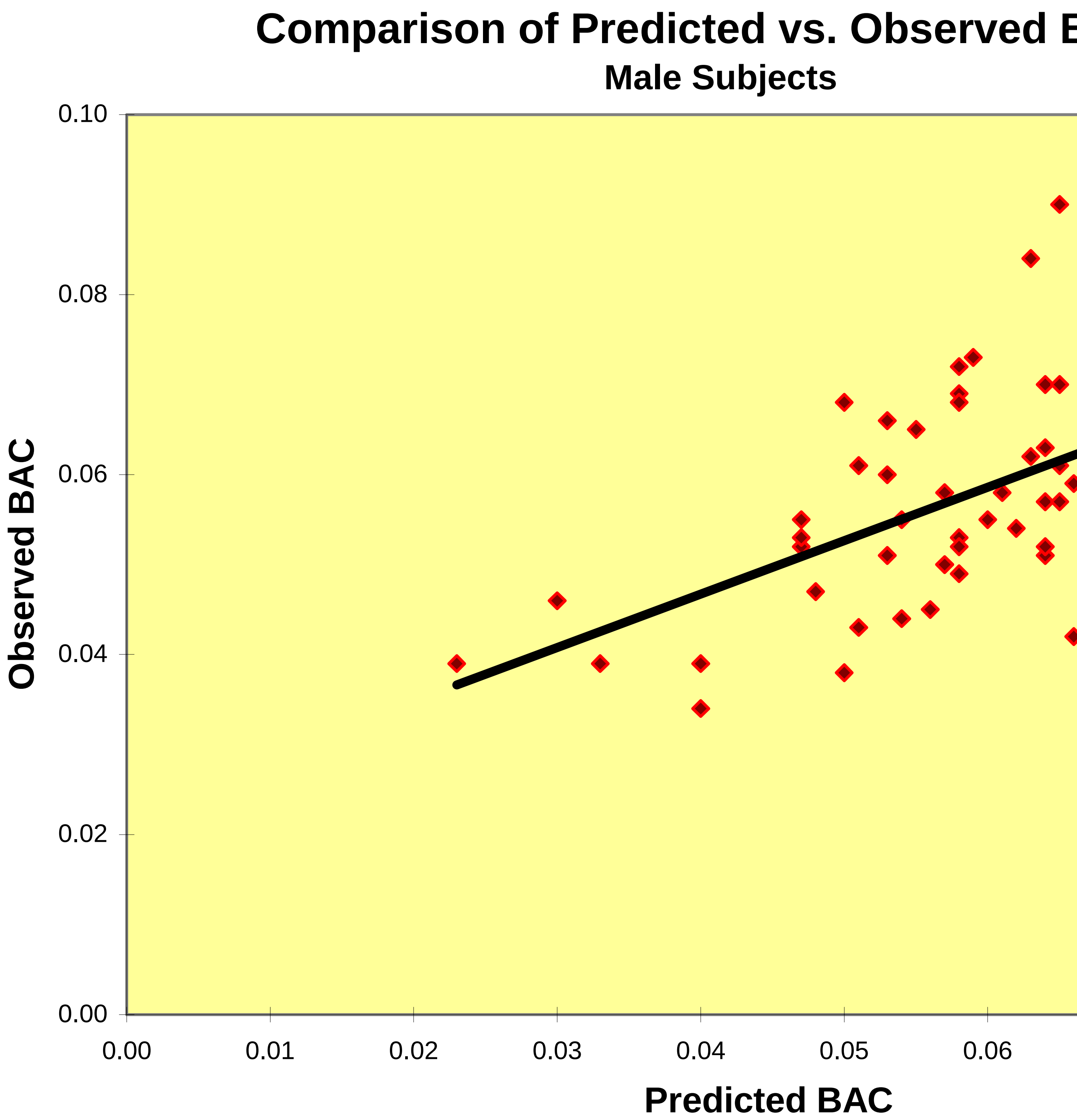
```
}

void body_compartment(subject &sub){

}

void update_compartments(subject &sub){//This function increments time and
updates ethanol concentrations
     if (sub.sex_MF == 'M')
          g = 12.50;
     else if (sub.sex_MF == 'F')
          g = 8.60;

     if (sub.sex_MF == 'M'){
          V = 0.152;
          K = 0.592;
     }
     else if (sub.sex_MF == 'F'){
          V = 0.300;
          K = 0.657;
     }
     sub.net_stomach_rate *= t;
     sub.net_body_rate *= t;
     sub.time += t;

     sub.grams_ethanol_stomach += sub.net_stomach_rate;
     sub.grams_ethanol_intestine += -sub.net_stomach_rate;
     sub.current_ethanol_concentration_intestine =
sub.grams_ethanol_intestine / sub.TBW;
     sub.intestine_to_body_rate = (g *
sub.current_ethanol_concentration_intestine);
     if (sub.stomach_to_intestine_rate > 0.000000)
          sub.net_intestine_rate = (sub.stomach_to_intestine_rate -
sub.intestine_to_body_rate) * t;
     else
          sub.net_intestine_rate = sub.intestine_to_body_rate * t;

     sub.grams_ethanol_intestine += -sub.net_intestine_rate;
     sub.grams_ethanol_body += sub.intestine_to_body_rate * t;
     sub.current_ethanol_concentration_body = sub.grams_ethanol_body /
sub.TBW;

     sub.current_ethanol_concentration_stomach = sub.grams_ethanol_stomach /
sub.TBW;

     sub.elimination_rate = ((V * sub.current_ethanol_concentration_body) /
(K + sub.current_ethanol_concentration_body));
     if (sub.intestine_to_body_rate > 0.000000)
          sub.net_body_rate = (sub.intestine_to_body_rate - (sub.TBW *
sub.elimination_rate)) * t;
     else
          sub.net_body_rate = (sub.TBW * sub.elimination_rate) * t;

     if (sub.intestine_to_body_rate > 0.000000)
          sub.grams_ethanol_body += sub.net_body_rate;
     else
```

```
          sub.grams_ethanol_body += -sub.net_body_rate;

     //cout << sub.grams_ethanol_body << "\t\t";
     //cout << sub.current_ethanol_concentration_body * 100<< "\t\t";
     //cout << sub.grams_ethanol_stomach << endl;
     //cout << sub.current_ethanol_concentration_stomach << "\t\t";
     //cout << sub.current_ethanol_concentration_intestine << endl;
     //cout << sub.grams_ethanol_intestine << "\t\t";
}

void find_peak_BAC(subject &sub){
     if(sub.current_ethanol_concentration_body > sub.peak_BAC)
          sub.peak_BAC = sub.current_ethanol_concentration_body;
}


void testing(subject &sub){//Testing to see if compartment model works
properly
     sub.grams_ethanol_stomach = 30;
     sub.initial_ethanol_concentration = sub.grams_ethanol_stomach / sub.TBW;
//Set initial concentration (g/l)
     sub.current_ethanol_concentration_stomach =
sub.initial_ethanol_concentration; //Initializes stomach concentration
     sub.current_ethanol_concentration_intestine = 0;
     sub.current_ethanol_concentration_body = 0;
     sub.grams_ethanol_intestine = 0;
     sub.grams_ethanol_body = 0;
     sub.time = 0;
     sub.peak_BAC = 0;
}

void get_compute_store(char infile[], char outfile[], subject &sub){
     ifstream in(infile);
     ofstream out(outfile);
     sub.sigma_TBW_M = 0;
     sub.sigma_TBW_F = 0;
     sub.sigma_BAC_F = 0;
     sub.sigma_BAC_M = 0;
     out.setf(ios::showpoint|ios::fixed);
     out << setprecision(3);
     out << "(sex, obs. BAC, obs. TBW, pred. BAC, pred. TBW)\n\n";
     while (!in.eof()){
          in >> sub.sex_MF >> sub.age_in_yrs >> sub.h_in_cm >> sub.w_in_kg >>
sub.observed_BAC >> sub.observed_TBW;
          testing(sub);
          TBW(sub);
          for(int i = 0; i < 120; i++){
              stomach_compartment(sub);
              intestine_compartment(sub);
              body_compartment(sub);
              update_compartments(sub);
              find_peak_BAC(sub);
          }
          if (sub.sex_MF == 'M'){
                    numofinputsubsmen++;
                    sub.difference_TBW_M = sub.observed_TBW - sub.TBW;
```

```
                    sub.difference_BAC_M = sub.observed_BAC - (sub.peak_BAC *
10);
                    sub.sigma_TBW_M += pow(sub.difference_TBW_M, 2);
                    sub.sigma_BAC_M += pow(sub.difference_BAC_M, 2);
          }
          else if (sub.sex_MF == 'F'){
                    numofinputsubswomen++;
                    sub.difference_TBW_F = sub.observed_TBW - sub.TBW;
                    sub.difference_BAC_F = sub.observed_BAC - (sub.peak_BAC *
10);
                    sub.sigma_TBW_F += pow(sub.difference_TBW_F, 2);
                    sub.sigma_BAC_F += pow(sub.difference_BAC_F, 2);
          }
          out << sub.sex_MF << "     " << sub.observed_BAC << "     " <<
sub.observed_TBW << "     " << sub.peak_BAC * 10 << "     " << sub.TBW <<
"\n";
     }
     double rms_TBW_M = sqrt(sub.sigma_TBW_M / numofinputsubsmen);
     double rms_BAC_M = sqrt(sub.sigma_BAC_M / numofinputsubsmen);
     double rms_TBW_F = sqrt(sub.sigma_TBW_F / numofinputsubswomen);
     double rms_BAC_F = sqrt(sub.sigma_BAC_F / numofinputsubswomen);
     out << "rms TBW men: " << rms_TBW_M << "  rms TBW women: " << rms_TBW_F
<< "\nrms BAC men: " << rms_BAC_M << "  rms BAC women: " << rms_BAC_F;
     out.close();
     in.close();

}

void get_compute_store_others(char infile[], char outfile[], subject &sub){
     ifstream in(infile);
     ofstream out(outfile);
     sub.sigma_BAC_M_cBAC = 0;
     sub.sigma_BAC_F_cBAC = 0;
     sub.sigma_BAC_M_BACest = 0;
     sub.sigma_BAC_F_BACest = 0;
     out << "(sex, obs. BAC, cBAC, BACest)\n\n";
     out.setf(ios::showpoint|ios::fixed);
     out << setprecision(3);
     while (!in.eof()){
          in >> sub.sex_MF >> sub.observed_BAC >> sub.BAC_cBAC >>
sub.BAC_BACest;
          if (sub.sex_MF == 'M'){
                    numofinputsubsmen++;
                    sub.difference_BAC_M_cBAC = sub.observed_BAC -
sub.BAC_cBAC;
                    sub.difference_BAC_M_BACest = sub.observed_BAC -
sub.BAC_BACest;
                    sub.sigma_BAC_M_cBAC += pow(sub.difference_BAC_M_cBAC,
2);
                    sub.sigma_BAC_M_BACest +=
pow(sub.difference_BAC_M_BACest, 2);
          }
          else if (sub.sex_MF == 'F'){
                    numofinputsubswomen++;
                    sub.difference_BAC_F_cBAC = sub.observed_BAC -
sub.BAC_cBAC;
```

```
                    sub.difference_BAC_F_BACest = sub.observed_BAC -
sub.BAC_BACest;
                    sub.sigma_BAC_F_cBAC += pow(sub.difference_BAC_F_cBAC,
2);
                    sub.sigma_BAC_F_BACest +=
pow(sub.difference_BAC_F_BACest, 2);
          }
          out << sub.sex_MF << "     " << sub.observed_BAC << "     " <<
sub.BAC_cBAC << "     " << sub.BAC_BACest << endl;
     }
     double rms_BAC_M_cBAC = sqrt(sub.sigma_BAC_M_cBAC / numofinputsubsmen);
     double rms_BAC_M_BACest = sqrt(sub.sigma_BAC_M_BACest /
numofinputsubsmen);
     double rms_BAC_F_cBAC = sqrt(sub.sigma_BAC_F_cBAC /
numofinputsubswomen);
     double rms_BAC_F_BACest = sqrt(sub.sigma_BAC_F_BACest /
numofinputsubswomen);
     out << "rms BAC men cBAC: " << rms_BAC_M_cBAC << "  rms BAC women cBAC:
" << rms_BAC_F_cBAC << "\nrms BAC men BACest: " << rms_BAC_M_BACest << "  rms
BAC women BACest: " << rms_BAC_F_BACest;
     out.close();
     in.close();

}

int main(){
     subject sub;
     //get_sub_info_compartment_quick(sub);
     //TBW(sub);
     get_compute_store("inputdata.txt", "outputdata.txt", sub);
     get_compute_store_others("inputdataothers.txt", "outputdataothers.txt",
sub);
     system("PAUSE");
     return 0;
}
```


Comparison of Predicted vs. Observed BAC's
Female Subjects
Observed BAC
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
Predicted BAC


Comparison of Predicted vs. Observed B
Male Subjects
Observed BAC
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Predicted BAC