MATH 365
Curve Fitting

1 Introduction

Curve fitting is closely related to interpolation. Both are used to study how well a specific experiment's data is consistent with conventional ideas about the phenomena. For example, population growth often occurs exponentially. The process of interpolation or curve fitting involves using data to find coefficients or parameters in the model that describes our pre-conceived notion about the how the observed phenomena is expected to behave.

Curve fitting and interpolation differ in that interpolation is used to get an exact fit to data points while curve fitting typically only approximately matches the data. Recall for interpolation we find a unique polynomial of degree n given $n+1$ data points. If the the data set is large we can find a polynomial that fits a subset of the data. Curve fitting on the other hand, may produce a curve that doesn't fit any of the data.

2 Least squares curve fitting

The most common way to find coefficients or parameters in a mathematical model it to use the least squares method that minimizes the distance between the data and curve. Let's begin curve fitting with fitting the Billing's, MT census data to a polynomial. The (x, y) data points for the hispanic population are (1980, 781), (1990, 871), (2000, 1257), (2010, 1835). If we chose to interpolate all of the data the result would be a cubic polynomial (why?).

2.1 Quadratic curve

Rather than interpolate the census data, let's fit it to the quadratic polynomial

$$y(x) = c_0 + c_1 x + c_2 x^2.$$

This results in the non-square system of equations

$$\begin{bmatrix} 781 \\ 871 \\ 1257 \\ 1835 \end{bmatrix} = \begin{bmatrix} 1 & 1980 & 1980^2 \\ 1 & 1990 & 1990^2 \\ 1 & 2000 & 2000^2 \\ 1 & 2100 & 2100^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$$

or

$$\mathbf{y} = \mathbf{A}\mathbf{c}.$$

There are not a unique coefficients c_0, c_1, c_2 that satisfy this system of equation because there are more equations than unknowns. Instead let's find a set of coefficients $\mathbf{c}$ and form

$$\hat{y}(x) = c(1) + c(2)x + c(3)x^2 \tag{1}$$

so that $\hat{y}(x_i) \approx y_i$. This means there will be residuals or errors in our quadratic polynomial.

In least squares curve fitting we find coefficients $\mathbf{c}$ that minimize the sum of squared errors or residuals

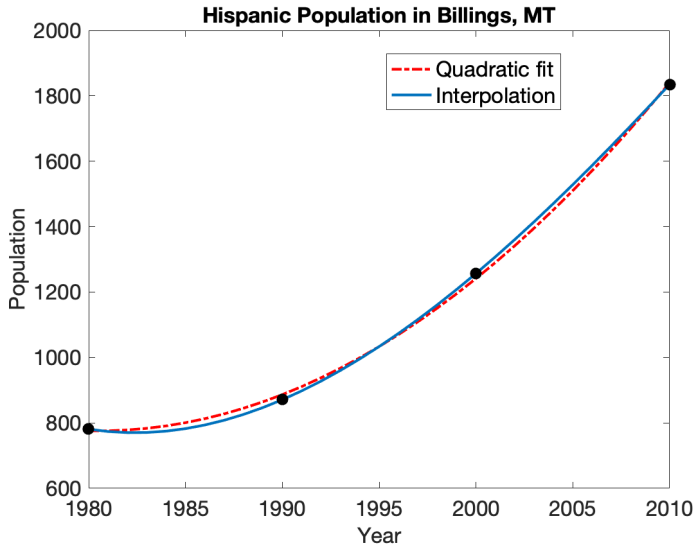
$$S = \sum_{i=1}^{n} (y_i - \hat{y}(x_i))^2.$$

The coefficients in $\mathbf{c}$ that minimize S are given by

$$\mathbf{c} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \tag{2}$$

With most software packages you will get a more accurate answer if you find $\mathbf{c}$ by solving the system of equations $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$ rather than computing $(\mathbf{A}^T\mathbf{A})^{-1}$ and using it to find $\mathbf{c}$.

Here is a plot of the data, the 3rd degree interpolating polynomial, and the curve that results when the data are fit to the quadratic polynomial (1):



The graph shows that there is not a significant difference between the cubic and quadratic polynomial models. We can find when they differ the most, and the corresponding predictions with the following commands:

```
[maxValue,maxIndex] = max(abs(yhat-px));
year=xs(maxIndex)*10+1960;
quad_est=yhat(maxIndex);
interp_est=px(maxIndex);
```

Note that values of the quadratic polynomial are located in yhat, the cubic interpolating polynomial is px, and each polynomial is evaluated at xs $\in [2, 5]$. The result of the commands are year = 1987, quad_est = 827.74 and interp_est = 808.19. Based on these values we conclude that the year the models differ the most is in 1987 when the quadratic curve fit predicts there were 828 hispanic people and the cubic interpolant predicts 808 people.

2.2 Exponential curve

Since we are dealing with population data, let's look at fitting the data to an exponential curve. Define the curve as

$$y(x) = c_0 + c_1 e^x.$$

Assume that we have transformed the years in the census data points so that they are (2, 781), (3, 871), (4, 1257), (5, 1835). When we fit these data to the exponential curve we get the non-square system of equations

$$\begin{bmatrix} 781 \\ 871 \\ 1257 \\ 1835 \end{bmatrix} = \begin{bmatrix} 1 & e^2 \\ 1 & e^3 \\ 1 & e^4 \\ 1 & e^5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}.$$

This can also be written as a matrix system,

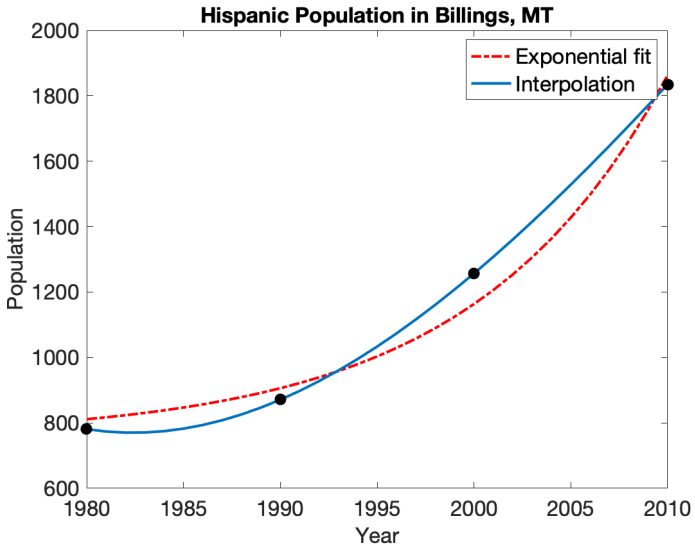
$$\mathbf{y} = \mathbf{A}_e \mathbf{c}_e$$

where the data vector $\mathbf{y}$ stayed the same as when we fit the data to a quadratic, but the matrix and corresponding coefficient vector changed because we changed the model to an exponential one.

We solve the system $\mathbf{A}_e^T \mathbf{A}_e \mathbf{c}_e = \mathbf{A}_e^T \mathbf{y}$ for the coefficients $\mathbf{c}_e$ and then form the model

$$\hat{y}_e(x) = c_e(1) + c_e(2)e^x. \tag{3}$$

Lastly, evaluate $\hat{y}_e(x)$ at a large number of values for x to get an idea of how well your model fits the data. Here is a plot of the data, the 3rd degree interpolating polynomial, and the curve that results when the data are fit to the exponential model (3):



Just by looking at the graph, it appears the exponential model (3) is the worse model for the hispanic population data, as compared to interpolation and the quadratic curve. This is because it does the worst job representing the data.

3 Rank and Pseudoinverse

The conditioning of the matrix $\mathbf{A}^T\mathbf{A}$ determines the stability of our fit to the curve $\hat{y}(x)$ or $\hat{y}_e(x)$. A problem is unstable if we change the data slightly and get a drastically different result for the coefficients $\mathbf{c}$. As we saw with interpolation it is best to transform the Census data years to smaller numbers by $(x - 1960)/10$ so that for the hispanic population, we have $x \in [2, 5]$. However, it is not always possible to transform the data to create a better conditioned matrix $\mathbf{A}^T\mathbf{A}$.

We cannot create a better conditioned matrix by transforming the data when $\mathbf{A}$ is not full rank. The rank of a matrix is the size of the largest collection of linearly independent columns or rows of $\mathbf{A}$. Columns or rows are linearly independent if none of them can be written as a linear combination of the others. Unfortunately many problems in real applications result in a rank deficient matrix because the model is not compatible with the data, or because there is not enough data to resolve the model. You can find the rank of a matrix in MATLAB by typing `rank(A)`. If $\mathbf{A} \in \mathbb{R}^{m \times n}$ a full rank matrix has rank $= \min(m, n)$, which means the matrix $\mathbf{A}^T\mathbf{A}$ is well-conditioned.

When $\mathbf{A}$ is not full rank we use the pseudoinverse to find the least squares estimate. A true inverse matrix satisfies $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. The pseudoinverse $\mathbf{A}^\dagger$ satisfies $\mathbf{A}\mathbf{A}^\dagger\mathbf{A} = \mathbf{A}$ In MATLAB we can find the least squares solution with the *pinv* command

```
c=pinv(A)*y
```

This function works both when $\mathbf{A}$ is full rank, and when it is not.

4 Epidemic Modeling

An epidemic model characterizes the spread of an infectious or contagious illness through a population. It requires parameters such as the current number of infected people and the probability a person will become infected. Data are used to identify these parameters and then the model can be used to understand the effect of different interventions, such as vaccinations. These results are often used to inform public health intervention decisions.

For the remainder of this section we make the following assumptions

- The number of infectious people at the start of day t is denoted by $I(t)$.
- The total population is denoted by N and we assume it is constant for all time.
- Once a person is *infected* they become *infectious* and stay *infectious* forever.
- If a person is infectious, they are equally likely with probability p to infect each noninfectious person on a given day.

4.1 Stochastic Modeling

Before using data to find parameters, let's assume we know the parameters and simulate how an illness spreads through a population. Let $x_n(t)$ be the infectious status of person n at the start of day t:

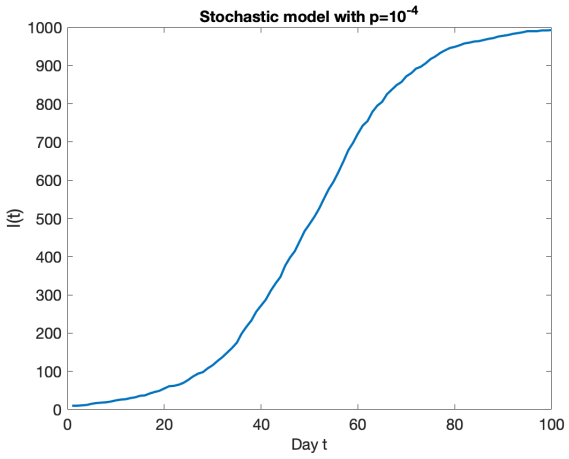
$$x_n(t) = \begin{cases} 1 & \text{if infectious} \\ 0 & \text{if not infectious} \end{cases}.$$

This means $I(t) = \sum_{n=1}^{N} x_n(t)$.

Let's build a stochastic model by assuming that the probability p a person becomes infected is known. Here is pseudocode to calculate the infectious status of the whole population over the time period $t = 1, \ldots, T$.

```
for t=1:T-1
    for i=1:N
        x(i,t+1)=x(i,t);   % Initialize infection status to be same as previous day
        for j=1:N
            find myrand   % Randomly choose a number between 0 and 1
            if myrand < p and x(j,t)=1
                x(i,t+1)=1   % Person i is infected by person j
            end
        end
    end
end
```

We view results from the stochastic simulation by plotting the number of infected people $I(t)$ over time t. Here are results of the first 100 days with $p = 10^{-4}$, a total population $N = 1000$, and an initial infected population of 10.



4.2 Deterministic Modeling

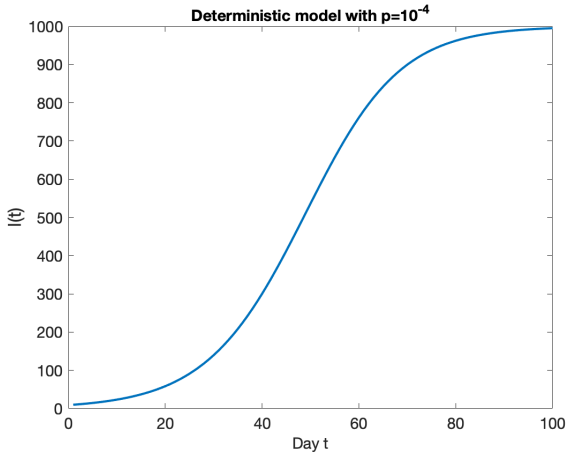
Each time we run the stochastic model we will get a different curve. That is because myrand is randomly chosen and we get a different value for it each time it is chosen. Rather than randomly choose if a person is infected, we can calculate the expected number of people infected on a day t. This results in the following deterministic model:

$$I(t+1) = I(t) + pI(t)\,(N - I(t)) \tag{4}$$

For these models, given $I(1)$ we can compute $I(2)$, $I(3), \ldots$

The deterministic models are much more efficient to compute than the stochastic models and their predictions may be just as reasonable. The advantage of the stochastic model is that it can give some idea of the uncertainty of its predictions via multiple simulations.

We view results from the deterministic model simulation by plotting the number of infected people $I(t)$ over time t. Here are results of the first 100 days with $p = 10^{-4}$, a total population $N = 1000$, and an initial infected population of 10.



This graph looks nearly identical to that created by the stochastic model. However, this curve is a bit smoother.

4.3 Continuous time modeling

The epidemic models we have discussed so far are called discrete-time models. They are discrete because time t takes on only integer values. Now we will approximate these models by continuous-time processes.

Consider that $I'(t) \approx \frac{I(t+1)-I(t)}{1}$. This means that (4) can be approximated by

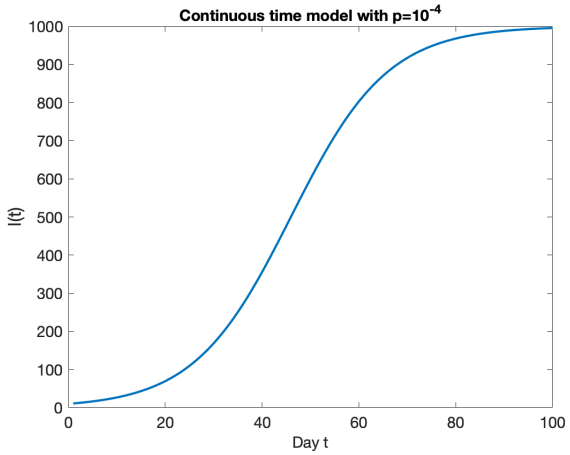
$$I'(t) = pI(t)\,(N - I(t)) \tag{5}$$

We can solve this differential equation exactly

$$I(t) = \frac{NI0}{I0 + (N - I0)\,e^{-pNt}} \tag{6}$$

where $I0$ is the initial population. I recommend differentiating (6) and verifying yourself that (5) is true.

We view results from the continuous time model simulation by plotting the number of infected people $I(t)$ over time t. Here are results of the first 100 days with $p = 10^{-4}$, a total population $N = 1000$, and an initial infected population of $I0 = 10$.



This graph looks identical to the one created with the deterministic model.

4.4 Fitting the model to data

In Sections 4.1-4.3 we input parameters representing the total population N, probability an infectious person will infect another person p and the initial number of infected people. Given these parameters all three models produced similar predictions of how the epidemic will spread over the next 100 days. Now we will address the situation where we want to use data to find values for these parameters.

Given a set of data points (t_j, I_j), $j = 1, \ldots, n$ representing the number of infected people on a specific day the goal is to find values for the parameters N, p and the number of people infected initially. We could try different values of the parameters, simulate the model, and see if we can find a set that produces a curve that looks like the data. While trying different values for the parameters may be a good way to understand how an epidemic could propagate, this approach is not very practical. This is because we may never find a curve that matches the data, or never be sure if our curve is "close" enough to the data.

The most common way to find parameters using data is to use the same least squares curve fitting from Section 2 and in the Regression lab. In the case of epidemic modeling, we want to find parameters N, p and $I0$ (initial number of infected people) that minimize the sum of squared errors or residuals

$$S = \sum_{j=1}^{n} (I_j - I(t_j))^2. \tag{7}$$

We can do this for the deterministic and continuous time models. Note that there is no closed form expression for $I(t)$ in the stochastic model. There are approaches to finding parameters using a least squares fit when there is no closed form expression for the model, but we will not cover them in this class.

Recall formula (2) that gives parameters that minimize S. If we are to use formula (2) we need to write (4) or (6) as a matrix system like

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \mathbf{A} \begin{bmatrix} N \\ p \\ I0 \end{bmatrix}. \tag{8}$$

For the deterministic model (4) we could write

$$I(t+1) - I(t) = pI(t)\,(N - I(t)) \tag{9}$$

and use differences of data points for the left hand side of (8). However, the right-hand side of (8) can be written as a linear function of p or N, but not both. It is also not clear how to find $I0$. If we let $t_1 = 0$ for the first data point, then we could let $I_1 = I0$. However, this might not be the best choice of $I0$ in order to make the residuals $I_j - I(t_j)$ small.

The problem is that we cannot find a matrix $\mathbf{A}$ for (8) because the model is not linear in the parameters. This leaves us with two choices: re-formulate the problem so that it is linear or use nonlinear least squares. We will explore both.

4.4.1 Linear deterministic model

Let's assume we know the total population N and that the initial number of infected people is given by the data I_1. We will use data (t_j, I_j), $j = 1, \ldots, n$ to find p that minimizes (7).

Plugging data into (9) gives the following system of equations

$$\begin{bmatrix} I_2 - I_1 \\ I_3 - I_2 \\ \vdots \\ I_n - I_{n-1} \end{bmatrix} = \begin{bmatrix} I_1(N - I_1) \\ I_2(N - I_2) \\ \vdots \\ I_{n-1}(N - I_{n-1}) \end{bmatrix} \begin{bmatrix} p \end{bmatrix}.$$

Now we can use formula (2), or more accurately solve the corresponding system of equations, to find the single parameter p.

Given p we can use the deterministic model to predict the number of infected people. Once you do this, graph the simulation along with the data to see how well your curve fits the data. Since we used least squares to find the fit there's a good chance your curve will not go through all of the data points.

4.4.2 Linear continuous time model

Let's assume we know the total population N, and we use data (t_j, I_j), $j = 1, \ldots, n$ to find the initial number of infected people and p. We will make a transformation of variables to get a transformed model that depends linearly on its parameters.

Let $Z(t) = \log(N/I(t) - 1)$, then the continuous time model (6) becomes

$$Z(t) = Z0 - pNt. \tag{10}$$

The model (10) depends linearly on pN and $Z0$. If we plug data into (10) we get

$$\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix} = \begin{bmatrix} 1 & -t_1 \\ 1 & -t_2 \\ \vdots & \vdots \\ 1 & -t_n \end{bmatrix} \begin{bmatrix} Z0 \\ pN \end{bmatrix}.$$

There are a few things to keep track of with this transformation

- Begin the procedure by transforming the data (t_j, I_j) to (t_j, Z_j) where $Z_j = \log(N/I_j - 1)$
- The vector of solutions contains pN so don't forget to divide by N to get p.
- The vector of solutions contains $Z0$ so transform back to get the initial population: $I0 = \frac{N}{1+e^{z0}}$.

Given p and $I0$ we can use the continuous time model to predict the number of infected people. Once you have values for $I(t)$, again graph them along with the data to see how well your curve fits the data. Since we used least squares to find the fit there's a good chance your curve will not go through all of the data points.

4.4.3 Nonlinear least squares

Nonlinear least squares uses the same sum of squares in (7) to find parameters that best fit the data. However, when $I(t)$ is nonlinear in the parameters, we don't have an analytical expression for parameters that minimize the sum of squares, like (2).

Methods for finding parameters that minimize (7) in the nonlinear situation are beyond the scope of this class. Instead we will use the MATLAB function *lsqcurvefit*. I recommend you read the MATLAB documentation for it. This function requires the following inputs:

- An initial estimate for the parameters. Put the parameters in a vector $\mathbf{c} = [I0, p, N]^T$ and call initial estimates c0.
- The data, let's call them (tdata, Idata).
- A function for $I(t)$. For the deterministic model this would be

```
function Id = Idfun(c,t)
    I0=c(1); p=c(2); N=c(3);
    T=t(end);
    Id=zeros(T,1);Id(1)=I0;
    for t=1:T-1
        Id(t+1)=Id(t)+p*Id(t)*(N-Id(t));
    end
end
```

  For the continuous time model this would be

```
function Ic = Icfun(c,t)
    I0=c(1); p=c(2); N=c(3);
    Ic=N*I0./(I0+(N-I0)*exp(-p*N*t));
end
```

  Pass the $I(t)$ function into the *lsqcurvefit* function by using the function handle `@` in the input argument.

Given a set of data (tdata,Idata), the following MATLAB commands estimates the parameters in a vector c.

```
c0 = [10, 1e-4, 1000];
c = lsqcurvefit(@Idfun, c0, tdata, Idata)
```

Similar commands can be given to find the parameters in the continuous time model.

Once you've found the parameters, plug them into the deterministic or continuous time model to predict the number of infected people. When you graph your curve for $I(t)$ along with the data there's still a chance your curve will not go through all of the data points.

5 Individual Lab questions

Please answer the following questions in Blackboard.

Part 1

1. Assume we fit the census data (2, 781), (3, 871), (4, 1257), (5, 1835) to the quadratic curve $y(x) = c_0 + c_1 x + c_2 x^2$ by solving $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$. Find $\mathbf{A}^T\mathbf{A}$.
2. Assume we fit the census data (2, 781), (3, 871), (4, 1257), (5, 1835) to the exponential curve curve $y(x) = c_0 + c_1 e^x$ by solving $\mathbf{A}_e^T\mathbf{A}_e\mathbf{c}_e = \mathbf{A}_e^T\mathbf{y}$. Find $\mathbf{A}_e^T\mathbf{A}_e$.
3. Assume we fit the census data of the hispanic population in Billings, MT to the quadratic curve $y(x) = c_0 + c_1 x + c_2 x^2$ by solving $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$. Use the quadratic curve to estimate the population in 1987.
4. Assume we fit the census data of the hispanic population in Billings, MT to the exponential curve $y(x) = c_0 + c_1 e^x$ by solving $\mathbf{A}_e^T\mathbf{A}_e\mathbf{c}_e = \mathbf{A}_e^T\mathbf{y}$. Use the exponential curve to estimate the population in 1987.

Part 2

1. For the stochastic model pseudocode on page 5, how would you specify the initial population of 10?
2. For the deterministic model in (4), how would you specify the initial population 10 in your code?
3. For the continuous time model in (6), how would you specify the initial population 10?
4. If $I(t) = \frac{NI0}{I0+(N-I0)e^{-pNt}}$ find $I'(t)$

Part 3

1. Assume we use data to estimate the parameter p in the linear deterministic model. The formula for p follows (2) as described in Section 4.4.1. For this situation, what is the dimension of the matrix $\mathbf{A}$ in (2)?
2. If $Z(t) = \log(N/I(t) - 1)$ then what is $I(t)$?
3. When fitting the curve to data, what are the unknown parameters in the linear deterministic model
4. When fitting the curve to data, what are the unknown parameters in the nonlinear deterministic model
5. When fitting the curve to data, what are the unknown parameters in the linear continuous time model
6. When fitting the curve to data, what are the unknown parameters in the nonlinear continuous model

6 Group Work

Part 1

1. Please write answers to the following questions on notecards.

   (a) Data for the black population in Billings, MT is (1960, 187), (1970, 74), (1980, 121), (1990, 159), (2000, 153), (2010, 326). Write the system of equations that results when you fit the data to the quadratic polynomial $y(x) = c_0 + c_1 x + c_2 x^2$.

   (b) Data for the black population in Billings, MT is (1960, 187), (1970, 74), (1980, 121), (1990, 159), (2000, 153), (2010, 326). Write the system of equations that results when you fit the data to the exponential curve $y(x) = c_0 + c_1 e^x$.

   (c) Data for the black population in Billings, MT is (1960, 187), (1970, 74), (1980, 121), (1990, 159), (2000, 153), (2010, 326). Write the system of equations that results when you fit the data to the exponential curve $y(x) = c_0 + c_1 e^x + c_2 \frac{e^x}{1+x}$.

2. Consider the following integral

$$erf(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-x^2} \, dx.$$

   It is called the error function and arises in the analytical solution of the heat equation, a partial differential equation in physics. It can also be used to express the standard normal distribution if we evaluate it at $t/\sqrt{2}$ rather than t. There is no analytical solution for the integral and people often use a table to look up values for it.

   (a) Create a your own "table" of values by generating 11 data points $y_k = erf(t_k)$ with $t_k = (k-1)/10$, $k = 1, \ldots, 11$. Use the MATLAB function *erf* to generate the data points.

   (b) Fit the data to a polynomial of degree 4:

$$p_4(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4,$$

   i.e. find least squares estimates for $\mathbf{c}$ by solving $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$. This means you need to form the matrix $\mathbf{A}$, data vector $\mathbf{y}$, and use the backslash operator to solve for $\mathbf{c}$.

   (c) Assume the MATLAB function *erf* finds the exact solution. Plot the error $|p_4(ts) - erf(ts)|$ where $ts = 0 : 0.01 : 1$.

   (d) Polynomials are not good basis function with which to approximate $erf(t)$. Using the same data as in 2a, fit it to the curve

$$f(t) = c_0 + e^{-t^2}(c_1 + c_2 z + c_3 z^2 + c_4 z^3), \quad z = \frac{1}{1+t}.$$

   Similar to 2b Form the matrix $\mathbf{A}$, data vector $\mathbf{y}$ and use the backslash operator to solve for $\mathbf{c}$.

   (e) Again assume the MATLAB function *erf* finds the exact solution. Plot the error $|f(ts) - erf(ts)|$ where $ts = 0 : 0.01 : 1$.

   (f) Use your function $f(t)$ to evaluate $erf(.05)$.

Part 2

1. Implement the pseudocode on page 5.

   (a) Things to think about: (i) Blackboard question 1. explains how to specify the initial population, (ii) use the MATLAB function *rand* to find a scalar random number between 0 and 1 (iii) the *if* statement in the pseudo code has two conditions, use the & command to make sure both are satisfied (iv) test the equality in the *if* statement using ==, not = (v) use the *sum* command in MATLAB to find the total number of infected people.

   (b) Run and graph multiple simulations on the same plot. Use the same values for p, N and $I(1)$ as in the lab.

   (c) Try different values for the number of people infected initially.

   Be prepared to discuss (i) why do you get different graphs each time you run the stochastic model, (ii) the magnitude of the difference between multiple simulation of the stochastic model with the same parameters (iii) how the graph changes when you change the number of people initially infected.

2. Implement the deterministic model in equation (4) and plot the number of infected people vs time. Use the same values for p, N and $I(1)$ as you did for the stochastic model.

   Be prepared to discuss the complexity of the code and the amount of time it takes to compute the stochastic model vs the deterministic model.

3. Implement the continuous time model in equation (4) and plot the number of infected people vs time. Use the same values for p, N and $I(1)$ as you did for the deterministic and stochastic model. Your plot should look the same as those for the stochastic and deterministic models.

   Be prepared to discuss the differences between your deterministic model code and your code for the continuous time model.

Part 3

1. Follow the instructions in the first part of the homework and load the monthly AIDS diagnoses data for the Boston area into MATLAB. Here are some tips: Once you download the data from the CDC I suggest first importing it into an Excel file and deleting all rows and columns that don't contain the number of infected people. Then use the MATLAB functions *readtable* and *table2array* to put it in an array. Be prepared to discuss if these data satisfy our assumptions for the model as described at the beginning of Section 4.

2. Use the MATLAB function *cumsum* to form a data vector **I** that contains the number of infected people. Plot the data and be prepared to discuss how well it appears the epidemic models we have been discussing are a good choice for this data set.

3. Use the deterministic model (4) to fit a curve to the data in the following manner:

   (a) Identify values for the total population N and the initial number of infected people $I0$.

   (b) Follow the instructions in 4.4.1 to estimate p. Be prepared to interpret your value for p.

   (c) Use your estimate of p and values for N and $I0$ you identified in 3a to form the curve for the deterministic model. Plot the curve and identify how well your curve fits the data.

4. Use the continuous time model (6) and fit the curve to the data in the following manner:

   (a) Form a function for the model that depends on N, p, and $I0$.

   (b) Read the documentation for the MATLAB function *lsqcurvefit*. Identify the following inputs: initial estimates of N, p, and $I0$, data values for the independent variable t, data values for the number of infected people.

   (c) Use *lsqcurvefit* to find estimates for N, p, and $I0$.

   (d) Use your estimates of N, p, and $I0$ to form the continuous curve. Plot the curve and identify how well your curve fits the data.