Project 2:

Using linear systems for numerical solution of boundary value problems

Goal

Introduce one of the most important applications of Linear Algebra to Engineering.

General requirements

- You may work alone or with *one* other person. If you work with someone else, hand in *one* Answer Sheet with both of your names on it.
- No groups bigger than two. No collaboration between groups. **Please read "My policies on Projects" posted on the course website.**
- Write your answers on the Answer Sheet provided in the last few pages of this document. Staple *with a staple, not a paper clip*[1] all the paper showing your *neatly presented* [2] work to the Answer Sheet.

Introduction

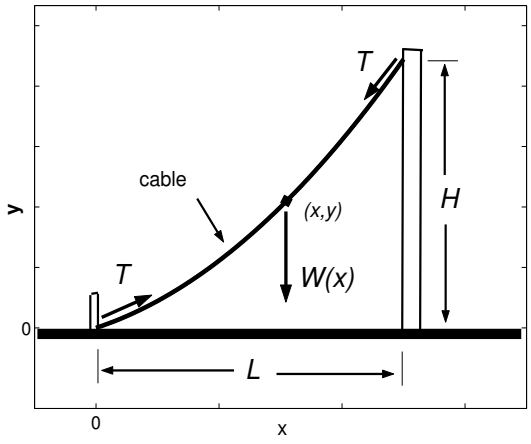
Most phenomena in Physics and Engineering are modeled by differential equations. You could have seen simple differential equations of the form $dy/dx = \text{const} \cdot y$ in Calculus I; perhaps, you are also taking a course (MATH 230 or MATH 271) about solving more general differential equations. A *boundary value problem* (BVP), whose approximate solution is illustrated in this Project, is just a differential equation supplied with two boundary values for the unknown variable y. BVPs occur in a variety of applications, such as (i) oscillation of membranes and strings (think of music!), (ii) sagging of beams and other structures under mechanical load, and (iii) radiation by atoms and molecules, to name just a few.

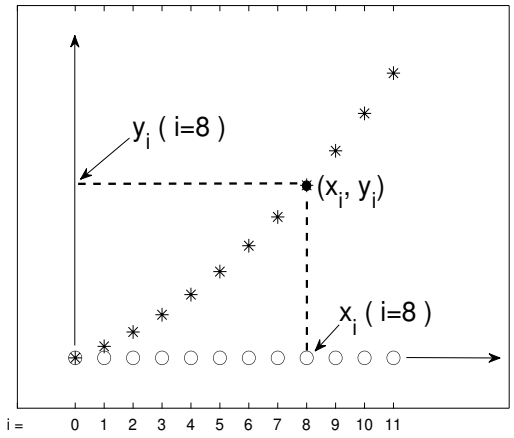
Most BVPs arising in modern applications of science and engineering are so complex that they cannot be solved exactly with pen and paper. Then one has to resort to solving them approximately on a computer. There is a whole area of applied mathematics which deals solely with this kind of solution of differential equations. It is called Numerical Analysis, and Linear Algebra lies at its foundation. Numerical analysis is very much driven by the need to solve practical problems.

In this Project, you will learn how to apply your knowledge of linear systems to solving one simple BVP. Linear systems enter this problem from two different angles. First, you will set up and solve a linear system to find an approximate formula for the second derivative of a function. Second, you will use that formula to set up another linear system, which describes the numerical solution of the BVP. Then you will use MATLAB to solve the latter linear system on a computer. Note that **the goal of this Project** is *not* to teach you how to solve BVPs but rather to introduce you to more applications of Linear Algebra.

[1] Your grade will be reduced by 5% if you hand in a pile of non-stapled sheets.

[2] I will reduce your grade by an amount left to my discretion in each particular case if your work is presented in a messy way and I have to waste time deciphering it.





Problem description and method of solution

A flexible power cable has one end staked to the ground and the other end fastened H ft up a pole L ft away, as shown in the left figure above. Let y be the cable's elevation at a point located x horizontal units from the stake. Our **goal** will be to find $y(x)$. We will do so by (approximately) solving the BVP satisfied by $y(x)$.

This BVP consists of a second-order differential equation and two boundary conditions. The differential equation is:

$$\frac{d^2y}{dx^2} = \frac{W(x)}{T}\,, \tag{1}$$

where the notations W and T are as follows (see also the left figure above):

- $W(x)$ is the cable's weight per unit length at a point $(x, y(x))$; W can be constant, if the cable is uniform, or it can depend on x if, for example, part of the cable is iced;
- T is the (constant) tension of the cable.

We assume that both $W(x)$ and T are known; their values will be given later on.

Since the ends of the cable are pinned at specific locations, the following *boundary conditions* must supplement the differential equation:

$$y(0) = 0 \qquad \text{and} \qquad y(L) = H\,. \tag{2}$$

The above equations (1) and (2) form a BVP for the cable's elevation, $y = y(x)$, as a function of the distance, x, from the stake.

The BVP (1) & (2) is sufficiently simple so that one can solve it exactly using just pen and paper. However, as you read in the Introduction, most differential equations of practical interest cannot be solved exactly. To solve them approximately, one has to use an approach that we illustrate below for the simple BVP (1) & (2). This approach relies on Linear Algebra.

The **key idea** behind finding an approximate solution to BVP (1) & (2) is this. Instead of looking for the cable's elevation, $y(x)$, for *all* values of x within the interval $[0,\, L]$, limit your goal to finding y only at *discrete* values of x on that interval. This is illustrated in the Figure at the top of this page: The left panel shows the original, continuous function $y(x)$, whereas the right panel shows a *discretized* version of the same function. The points $(x_i,\, y_i)$ in the right panel represent these discrete values; the notations "x_i" and "y_i" will be explained below.

We will first describe how these discrete points $(x_i,\, y_i)$ are set up (i.e., defined) and then will explain how they can be found using a discrete (and therefore *approximate*) form of the differential equation (1).

The very first step of defining a discretized solution for the cable's elevation is to subdivide the interval $[0,\, L]$ into subintervals, which we will assume to be of equal length. To that end, *choose* a number n of subintervals; then the length of one subinterval is

$$h = L/n\,.$$

(Note that this h is *not at all related* to the height H of the right endpoint of the cable.) For example, in the right panel of the Figure, $n = 11$, and so $h = L/11$. In general, the person solving a BVP has some freedom in deciding on what n should be. However, in this Project, you will be told what to take for n.

Now consider discrete points

$$x_i = i\,h, \qquad \text{where } i = 0, 1, \ldots, n. \tag{3a}$$

These points subdivide the interval $[0,\, L]$ into n subintervals of length h: $[0,\, h]$, $[h,\, 2h]$, $[2h,\, 3h]$, …, $[(n-1)h, L]$. In particular, note that

$$x_0 = 0 \qquad \text{and} \qquad x_n = L. \tag{3b}$$

Now do some work on your own: Look at the right panel of the Figure and count how many points x_i are there in total. Also, how many of these points are *strictly inside* the interval $[0,\, L]$? Now generalize your answers for an arbitrary n and record it somewhere in your worksheets. You will need to refer to these answers when you do Steps 3 and 4 of this Project.

To complete the description of the discretized solution for the cable elevation, denote

$$y(x_i) = y_i. \tag{3c}$$

We are now ready to state the problem that you will solve in this Project:

You will need to find the finitely many values y_i, $i = 1, 2, \ldots (n-1)$, which will yield an approximate solution to the BVP (1) & (2). The reason that this solution is only approximate rather than exact is because we will have to approximate the differential equation (1) for a continuous function $y(x)$ by a finite set of linear equations (i.e., a *linear system*) for discrete values y_i.

Clearly, if the discretization step h is taken to be very small, there should be almost no difference between the continuous variables x and y, on one hand, and the discrete sets $\{x_0, x_1, x_2, \ldots, x_n\}$ and $\{y_0, y_1, y_2, \ldots, y_n\}$, on the other. Yet, this discretization trick will allow you to replace the continuous BVP given by Equations (1) and (2), which you do not know how to solve, by a linear system, which you know how to solve. The steps of going from the continuous BVP to a linear system are described below.

Step-by-step instructions

1. Your goal in this part is to derive an *approximate* formula for the second derivative, $y''(x)$, that involves only the *discrete* values of $y(x)$ at certain points. It is shown in higher-level courses like Numerical Analysis that for any sufficiently smooth function $y(x)$, the second derivative $y''(x)$ at a

point $x = a$ (for some arbitrary a) can be approximated using the values of $y(x)$ at three nearby points: $x = a - h$, $x = a$, and $x = a + h$:

$$y''(a) \approx A_{-1}y(a-h) + A_0 y(a) + A_1 y(a+h)\,, \tag{4}$$

where A_{-1}, A_0, A_1 are some constant coefficients.

Find A_{-1}, A_0, A_1 in Equation (4), as demonstrated (for a different problem) in Example 8 on pp. 88–89 of the textbook and in Appendix A found after these Instructions.

You may solve the resulting linear system either by hand (using its augmented matrix!) or by command `RowReduce` in Mathematica.[3]

Write your formula (i.e., Equation (4) with your numerical values for the coefficients A_{-1}, A_0, A_1) on the Answer Sheet *and* attach to it the paper with your derivation. Check your answer by comparing it with the solution of Exercise 21 of Section 1.8.

Note: If you have been unable to obtain the correct result yourself, you should still put the result from Exercise 21 on the Answer Sheet and state *there* where you got your answer from. You will receive 5 points for doing so. On the other hand, the correct answer stated without a reference to Exerise 21 (or without your own work clearly presented) will receive only 1 point.

2. Let now a be one of the points x_i defined in Equation (3a) above: $a = x_i$. For a nearby point $a - h$, we then have:

$$a - h = x_i - h = ih - h = (i-1)h = x_{i-1}.$$

According to Equation (3c), we can write $y(a) = y(x_i) = y_i$, and then similarly:

$$y(a-h) = y(x_{i-1}) = y_{i-1}.$$

Use these observations to rewrite the approximate formula you have obtained for $y''(a)$ (that is, for $y''(x_i)$) in terms of y_{i-1}, y_i, etc.

Now, let x_i be one of the *interior* points on the interval $[0,\, L]$ (i.e., when $0 < i < n$). At such an interior point, approximate the differential equation (1) using the approximate formula for $y''(x_i)$ that you have just written down. To this end, simply substitute that formula into the left-hand side of Equation (1). On the right-hand side of (1), use similar notations, i.e., $W(x_i) = W_i$. Write the resulting equation at the point x_i in the Answer Sheet.

Thus, the outcome of this step should have the form

$$(\quad)y_{i-1} + (\quad)y_i + (\quad)y_{i+1} = (\text{something known})\,, \tag{5}$$

and your job is to supply the coefficients or expressions inside the parentheses. Note that W_i and T are considered to be *known* in this problem. Also note that the coefficients in $(\cdots)$ in Eq. (5) must be in terms of h.

[3] See Mathematica's Help on how to use that command.

3. Here you will practice setting up a linear system to approximate Equation (1) by using a *small* number of discretization points. Later you will need to use a large number of discretization points.

   Draw the interval $[0,\ L]$, subdivide it into 5 subintervals, and label the resulting points x_0, x_1, ..., x_5 (see Equations (3a) and (3b)).

   Write your equation (5) for $i = 1$. Then repeat this for each *interior* point x_i of $[0,\ L]$. Note that each of your equations couples the value y_i to the values y_{i-1} and y_{i+1}. Whenever you require the values y_0 and y_n, these are found from the boundary conditions (2). Thus, the equations at the interior points $x_i,\ i = 1, \ldots, 4$ form a linear system for the values y_i. Write this linear system for the unknown values $y_1,\ y_2,\ y_3,\ y_4$ in a matrix form. Keep using the notation h for $L/5$, etc.. [4]

4. Let now n denote the number of subintervals into which discretization points x_1, x_2, etc. divide the interval $[0,\ L]$.

   Set up and solve a linear system (5) for the case $n = 20$ using MATLAB (see Appendix B after these Instructions). Use the values $L = 100$ ft, $H = 12$ ft, $T = 3000$ lb, and

$$W(x) = \begin{cases} 6 & \text{lb/ft (iced cable)}, \quad 0 \le x \le L/2\,\text{ft} \\ 3 & \text{lb/ft (clean cable)}, \quad L/2 < x \le L\,\text{ft}. \end{cases}$$

   Write the numerical values of your solution at several locations along the cable, as requested in the Answer Sheet. Round these values to *4 decimal places*. Also, use Matlab to plot the graph $y = y(x)$ which you obtained on the interval $0 \le x \le L$ (i.e., include the end points). Label the axes. For example, to label the x-axis, type:
   `xlabel('whatever you want to use for the x-label','fontsize',12).`

   Print your computer output, including the plot of $y(x)$, and staple it to the Answer Sheet.

Extra credit (partial credit is given only if the solution is mostly correct)
Consider a *broken* power line, where one end of the cable is still fastened to the pole but the other (free) end lies on the ground. At the point of contact with the ground, the cable's slope is zero, and therefore the condition

$$y'(0) = 0 \tag{6}$$

must be added to the boundary conditions (2). (Note that in this case, the tension T at the point of contact of the cable with the ground is provided solely by the static friction between the cable and the ground.) The resulting BVP, consisting of Equations (1), (2), and (6), has a solution only for one particular value of L. This L is the distance from the pole to the point where the broken cable first touches the ground. Find this L. Use the values: $T = 500$ lb, $W(x) = 6$ lb/ft (assume the cable is uniformly iced), and the other parameters as before.

Hints: 1. One way to solve this problem is to use only two (which ones?) of the three boundary conditions (2) and (6) and view this as a differential equation with an *initial* (as opposed to boundary) condition. Then use the third boundary condition to find L. If you use this method, not related to MATH 122, you must explain every step of your work.

[4] That is, your coefficients must include h but not L.

2. Another way to solve the same problem will actually involve linear algebra and will require only a small modification of the setup that you have used in Steps 3 and 4. You may first decide on a number of subintervals n into which you will subdivide the (now unknown) interval $[0,\, L]$. Thus, unlike in Steps 3 and 4, h is now unknown. Next, approximate the first derivative in the boundary condition (6) by the formula

$$y'(a) \approx \frac{y(a+h) - y(a)}{h}\,. \tag{7}$$

Now, count the number of equations and unknowns in your equations. Manipulate them to obtain a linear system that can have a unique solution. From that solution, deduce L.

Note: You will receive separate credit for solving this problem using either of the Hints. (I.e., you can receive up to $9 + 23$ points for it; see the Answer Sheet.) Now, if you solve the problem using these two different methods, you will obtain answers that differ by more than just in their decimal places. You will receive additional credit for explaining where this difference comes from. You must point at a *specific place or equation* in your work where this difference came from and explain why you think it is this place/equation.

Appendix A

Here I present an example of finding the coefficients in a formula similar to Equation (4) of this Project. An analogous derivation is found in Example 8 on pp. 88–89 of the text book.

Recall from Calculus I that the *first* derivative of a function $y(x)$ is defined as

$$y'(a) = \lim_{h\to 0} \frac{1}{h}\left(y(a+h) - y(a)\right),$$

which means that

$$y'(a) \approx \frac{1}{h}\left(y(a+h) - y(a)\right) \qquad \text{for sufficiently small } h. \tag{A1}$$

Here we have written $y(a)$, $y(a+h)$ instead of $y(x)$, $y(x+h)$ to indicate that "a" is a particular value of the variable "x". Thus, below, notation "a" will be reserved for some fixed value, while "x" will denote a variable that can take on a continuum of values.

Note that Equation (A1) is satisfied *exactly* if $y(x) = 1$ (or any const) and $y(x) = x$. For example, if one substitutes $y(x) = x$ into (A1), one has:

$$\text{l.h.s.} = x' = 1; \qquad \text{r.h.s.} = \frac{(a+h) - a}{h} = 1; \qquad \Rightarrow \qquad \text{l.h.s.} = \text{r.h.s.}. \tag{A1*}$$

You should similarly verify yourself that (A1) holds for $y(x) = 1$. If, however, $y = x^2$, the right-hand side of Equation (A1) yields $2a + h$, which equals $y'(a) = 2a$ only approximately when h is small. You should verify this statement yourself as well.

Then we ask: Can one modify the expression on the right-hand side of Equation (A1) so that the new expression would *exactly* equal $y'(x)$ not only for $y = \mathrm{const}$ and $y = x$, but also for $y = x^2$? The short answer is 'yes'. The details are provided below.

We look for the required expression in the form given by the right-hand side of Equation (4): [5]

$$y'(a) \approx A_{-1}y(a-h) + A_0 y(a) + A_1 y(a+h)\,. \qquad (A2)$$

Recall that we want this equation to be satisfied *exactly* rather than approximately whenever $y = 1$, $y = x$, and $y = x^2$. (If you forgot why, re-read the paragraph containing Equation (A1′).) Then we simply substitute each of these three functions into Equation (A2):

$$\underline{\text{when } y(x) = 1}: \qquad 1' = 0 = A_{-1}\cdot 1 + A_0 \cdot 1 + A_1 \cdot 1 \qquad (A3a)$$

Note that the three "1"s on the r.h.s. occurred because $y(a-h) = y(a) = y(a+h) = 1$ when $y(x) \equiv 1$.

$$\underline{\text{when } y(x) = x}: \qquad x' = 1 = A_{-1}\cdot (a-h) + A_0 \cdot (a) + A_1 \cdot (a+h) \qquad (A3b)$$

Similarly to the previous note, the coefficients of A_1, A_0, A_1 came from the fact that $y(a-h) = a-h$ etc. for $y(x) = x$.

$$\underline{\text{when } y(x) = x^2}: \qquad (x^2)'|_{\text{at } x=a} = 2a = A_{-1}\cdot (a-h)^2 + A_0 \cdot (a)^2 + A_1 \cdot (a+h)^2 \qquad (A3c)$$

The linear system (A3) for the unknown coefficients A_{-1}, A_0, A_1 can be simplified once we notice that we have required it to be true for *any* $x = a$. In particular, it must be true for $a = 0$. Substituting $a = 0$ into Equations (A3) and transforming the resulting linear system to the reduced echelon form, we find (verify):

$$A_{-1} = -1/(2h), \quad A_0 = 0, \quad A_1 = 1/(2h)\,. \qquad (A4)$$

The values for A_{-1}, A_0, A_1 in Equation (4) are obtained similarly, also by requiring that that equation be exact for $y = 1$, $y = x$, and $y = x^2$.

Food for thought:[6] Substitute the coefficients (A4) into formula (A2). You will obtain a different approximation to $y'(a)$ than (A1). Which approximation is more accurate? You may investigate this by taking $y(x) = 2x^2 + 3$ and $a = 1$, $h = 0.1$. Is your finding consistent with the conditions from which you have derived (A4)?

Appendix B

The hints about Matlab listed below may help you in successfully completing this Project.

- Read Appendix A in the textbook (you do *not* need the material from subsections A.4, A.5, A.10, and A.11) and Matlab Primer by K. Sigmon, posted on the class web site (you do *not* need the material from pp. 10–14 and 18–21). You may, but do not have to, read other Matlab resources posted there.
- The coefficient matrix for solving the linear system for $y_1, y_2, \ldots, y_{n-1}$ has the dimension $(n-1) \times (n-1)$. Thus, for $n = 20$, it contains 361 entries. However, only the elements on the main diagonal and the two diagonals closest to it are nonzero (you may easily see this after you complete Step 3 of this assignment). You must *not* type all the nonzero entries one by one into the matrix; if you do

[5] You may wonder why the expressions on the right-hand sides in Equations (A2) and (4) have the same form, even though the derivatives on the left-hand sides are different. The answer is given in courses like Numerical Analysis, which is offered at UVM as MATH 237 and 337.

[6] You do not need to do what is described in this paragraph to be able to complete this Project.

so, **your score will be reduced**. Instead, use the `for`-loop of Matlab (see the Primer, Sec. 6). For example, here is the loop that defines matrix

$$A = \begin{pmatrix} 5a & 6a^2 & 0 & 0 \\ 0 & 5a & 6a^2 & 0 \\ 0 & 0 & 5a & 6a^2 \\ 0 & 0 & 0 & 7a \end{pmatrix}$$

for some given value of parameter a:

```
N = 4;      % dimension of matrix A
a = 0.1;    % some arbitrarily chosen value
for  i = 1 : N-1
    A(i,i) = 5*a;
    A(i,i+1) = 6*a^2;
end
A(N,N) = 7*a;
A           % this simply displays the result
```

A few clarifications about this example are in order. ***Read them!*** Really, read them. They will likely help you avoid some agonizing moments when constructing the coefficient matrix in Step 4.

First, note that the entries which you do not specify are set to zero by default in Matlab.

Second, note that in the above loop, we defined nonzero entries in the first $(N-1)$ rows of matrix A. This is because the indices of such entries in these rows follow a certain pattern. The indices of the nonzero entry in the last row do not follow that pattern, and hence we had to define that entry outside the loop.

Likewise, in the matrix that you have to set up for this Project, indices of nonzero entries follow the same pattern in most, but not in all, rows. Therefore, first identify those rows with such a pattern and define nonzero entries in these rows within a loop. Then, define nonzero entries whose indices do not follow a patern, outside the loop.

Third, in the vector on the right-hand side of your linear system, you should also determine which entries follow a pattern and define them within a loop (or loops). Then, define the entries which do not follow a pattern outside the loop.

Fourth, notice that the above example defines the matrix's dimension, N, *only once*, in the first line. It then refers to 'N' rather than to '4' as the matrix's dimension. This way, one can *easily* change the dimension by changing N only in the first line and *not* needing to change anything else in the code by hand. Similarly, parameter a is defined *once* in the second line and then conveniently used inside the code. Use this good practice of *avoiding hardcoding* (i.e., defining parameters at the beginning and then referring to them instead of using "hard" numbers) in your code.
Your score will be reduced by 1 point for every instance of hardcoding.

Finally, but very importantly: Make sure that the matrix you have constructed is indeed the one you intended to construct. You may do so by displaying that matrix for some n less than 20. (For example, you may use $n = 5$, since in this case you know from Step 3 what your matrix should look

like.) This, on one hand, will not change the logic of your script and, on the other, will produce a matrix of a smaller dimension, which you should be able to inspect on the screen.

Another convenient method to see what your matrix A looks like is to click on the small cell symbol next to A in Matlab's `Workspace` (this is usually found in the right-hand margin of your Matlab window).

- To solve a linear system, one has many options in Matlab.
  - The best one is to use the *backslash* notation which Matlab's founders invented: see Sec. 3 in Matlab Primer or simply google 'backslash matlab'.
  - Another way is to use the inverse matrix A^{-1}, for which Matlab's notation is `inv(A)`. (This may be slower than other options if A has large dimensions, but for a small matrix as here, it will work as fast as the backslash.)
  - The already familiar command `rref(B)`, where B is the augmented matrix, is not preferred in this case. This is because one first needs to create B using a matrix concatenation command and then extract the solution as the last column of the rref'ed B.

- This comment is about plotting $y(x)$ on the interval $[0,\ L]$. Note that your solution $y_1,\ \ldots,\ y_{n-1}$ does not include the boundary values $y(0)$ and $y(L)$ because they were known, not solved for. Therefore, you need to append these values to your solution vector. See Sec. A.4 in Appendix A of the textbook or Sec. 5 in the Matlab Primer. Here is an example of how to append values to a row vector:

```
a = [1 2];
a = [0 a 3]
```

  Note that since your solution $\{y_1, y_2, \ldots\}$ is a column (rather than row) vector, you will need to modify the above example by invoking a transposition command.

- Finally, you will need to define a vector `x` in order to plot your solution versus it. This can be done with a `for`-loop, but a much better way is described in Sec. 10 of the Primer; a related example is also found in Sec. 11 there. Note that your vector `x` must contain *all* points, including $x = 0$ and $x = L$.

If you would like to learn more about numerical methods,

Take the course MATH 337.

Acknowledgement

This project is based on an idea by Professor Scott C. Fulton (Clarkson University) and some material from Chapter 8 of the book "Numerical Analysis: A Practical Approach" by M. Maron and R. Lopez, 3rd Ed., Wadsworth (Belmont, 1991).

Submission instructions:

1. Please submit as a single pdf file only page 2 (i.e., the next page), of this Answer Sheet, as well as all supporting handwritten work. Please do *not* include either this page with submission instructions or/and the Project description itself.

2. There will be a Matlab plot, as well as a printout of your code, that you will need to include in your submitted work for this project. In order both to organize your work and to save paper/file size, please put any of your printouts and plots into a Word file (see below), convert it to a pdf, and submit only that file. If this is not done, I will **reduce your score by 5 points**.

   To include a figure into your main file, I recommend the following steps. Create a .png, .jpeg, or .tiff figure; Insert it into a Word document; Convert the Word document into a pdf. The syntax in Matlab for creating a .png file is: print -dpng 'foldername/filename'; see help print for more details. It is similar for other picture formats.

3. In addition to the printout of your code included in the submitted file, you must also *e-mail me this code*. When I run it, this code should produce your plot. If you submit *either* a hard or a soft copy of your code, *but not both*, I will **reduce your score by 5 points**.

4. The subject line of your e-mail submission of the code must contain the string MATH_122. It is case-insensitive; thus, math_122 or Math_122 will also work. However, the underscore must be present: e.g., math122 or math 122 will *not* work. Moreover, the subject line must indicate the fact that this is a submission of a code for Project 2. I will **reduce your score by up to 2 points** if this is not done.

5. Your code must be named according to this convention: `p2_YourName(s).m`. E.g., my own code would be named `p2_tlakoba.m`. If your code name does not follow this convention, I will **reduce your score by 2 points**.

6. Finally, a **strong request**: Contrary to what you might have learned in other classes, I ask you **not** to put commands `clc` and `close all` at the beginning of your codes. If I want to clear my command window and close figure windows, I will do so myself, without unsolicited outside "help".

Answer Sheet is continued on next page

Reminder: ***You MUST staple pages with your neatly presented work to get full credit.***

Name(s): _____

1. (27 points) Approximate formula for $y''(a)$ (attach a *Mathematica* printout if applicable):

2. (15 points) Approximation to the differential equation (1) at the interior point x_i:

3. (27 points) Linear system (in matrix form) for the case $n = 5$:

4. (31 points) Solution for the elevation y_i computed for $n = 20$ and rounded to four decimal places:

i	5	10	15
x_i, ft			
y_i, ft			

Attach your code and a Matlab-produced plot of your solution; do not forget to label the axes.

Extra credit (9 points for following Hint 1; 23 points for following Hint 2; 5 more points for explaining why the answers obtained when following these two Hints are different) Attach your work.

$$L = \qquad \text{ft}$$