MATH 635 Final Assessment

Matthew Tiger

December 13, 2015

Problem 1. Provide a rigorous proof of the case $x_0 = a$ in the Fundamental Lemma of the Calculus of Variations:

Theorem 1 (Fundamental Lemma of the Calculus of Variations)**.** *Suppose $M(x)$ is a continuous function defined on the interval $a \leq x \leq b$. Suppose further that for every continuous function $\zeta(x)$,*

$$\int_a^b M(x)\zeta(x)dx = 0.$$

Then

$$M(x) = 0 \text{ for all } x \in [a, b].$$

Solution. Suppose to the contrary that $M(x) \neq 0$ at the point $x_0 = a$. In that case either $M(a) > 0$ or $M(a) < 0$. Let us first assume that $M(a) > 0$. Due to the continuity of $M(x)$ there is some neighborhood of a where the function is positive, i.e. there is some $\delta > 0$ such that if $|x - a| < \delta$ then

$$|M(x) - M(a)| < \frac{M(a)}{2} \quad \text{for } x \in [a, b].$$

Thus, $0 < M(a)/2 < M(x)$ for $x \in [a, a + \delta)$. Choose the function $\zeta(x)$ to be the linear spline interpolating the points $(a, 3M(a)/2)$ and $(a + \delta, 0)$ with support on $[a, a + \delta)$, i.e.

$$\zeta(x) := \begin{cases} \frac{-3M(a)}{2\delta}(x - (a + \delta)) & \text{if } a \leq x < a + \delta \\ 0 & \text{if } a + \delta \leq x \leq b. \end{cases}$$

Clearly $\zeta(x)$ is continuous and positive on the interval $[a, a + \delta)$. Thus,

$$\int_a^b M(x)\zeta(x)dx = \int_a^{a+\delta} M(x)\zeta(x)dx > \frac{M(a)}{2}\int_a^{a+\delta} \zeta(x)dx > 0.$$

However, by our supposition

$$\int_a^b M(x)\zeta(x)dx = 0,$$

a contradiction. Therefore, if $M(a) > 0$, the function $M(x) \equiv 0$ on the interval $[a, b]$.

If $M(a) < 0$, then we can repeat the argument above replacing $M(x)$ with $-M(x)$. To demonstrate, let us investigate the case when $M(a) < 0$. Due to the continuity of $M(x)$ there is some neighborhood of a where $-M(x)$ is positive, i.e. there is some $\delta > 0$ such that if $|x - a| < \delta$ then

$$|-M(x) + M(a)| < \frac{-M(a)}{2} \quad \text{for } x \in [a, b].$$

Thus, $0 < -M(a)/2 < -M(x)$ for $x \in [a, a + \delta)$. Choose the function $\zeta(x)$ to be the linear spline interpolating the points $(a, -3M(a)/2)$ and $(a + \delta, 0)$ with support on $[a, a + \delta)$, i.e.

$$\zeta(x) := \begin{cases} \frac{3M(a)}{2\delta}(x - (a + \delta)) & \text{if } a \leq x < a + \delta \\ 0 & \text{if } a + \delta \leq x \leq b. \end{cases}$$

Clearly $\zeta(x)$ is continuous and positive on the interval $[a, a+\delta)$. Thus,

$$\int_a^b -M(x)\zeta(x)dx = \int_a^{a+\delta} -M(x)\zeta(x)dx > \frac{-M(a)}{2}\int_a^{a+\delta} \zeta(x)dx > 0.$$

However, by our supposition

$$\int_a^b M(x)\zeta(x)dx = 0,$$

a contradiction. Therefore, if $M(a) < 0$, the function $M(x) \equiv 0$ on the interval $[a, b]$ and we have proven both cases. $\square$

Problem 2. Consider the differential equation

$$y'' - y = -x, \quad 0 < x < 1 \quad y(0) = y(1) = 0 \tag{1}$$

as in Example 15.12 on page 502. Use the basis $\{\phi_j(x)\} = \{x^j(1-x)^j\}$, as in section 15.5.1, to compute approximations to the exact solution using the finite-element method.

Provide relative errors at the points 0.25, 0.50, and 0.75 of the approximations using the first $n = 2, 3, 4$ basis functions. Plot the corresponding approximations y_2, y_3, y_4, and the exact solution y. Then find the first value of j for which the relative error at all three points is less than 0.5%.

Solution. The differential equation presented in the problem is a second order linear differential equation. It is easily shown that the homogeneous solution is given by $y_h(x) = c_1e^{-x} + c_2e^x$ and that a particular solution is given by $y_p(x) = x$. Thus the general solution is $y(x) = c_1e^{-x} + c_2e^x + x$. Using the boundary conditions, we see that the exact solution is

$$y(x) = \frac{e^x e}{1-e^2} - \frac{e^{-x}e}{1-e^2} + x \tag{2}$$

We now wish to approximate the exact solution $y(x)$. Note that the exact solution to the differential equation (1) is a continuous function. This fact combined with the fact that $\{\phi_j(x)\}$ form a basis of the function space shows that the continuous function $y(x)$ can be approximated with a linear combination of the basis functions. Therefore, we wish to find an approximation $y_n(x)$ to the exact solution $y(x)$ where

$$y_n(x) = \sum_{j=1}^{n} a_j\phi_j(x). \tag{3}$$

Note that our basis functions $\{\phi_j(x)\}$ satisfy the boundary conditions, i.e. $\phi_j(0) = \phi_j(1) = 0$ so that $y_n(x)$ also satisfies the boundary conditions.

Corollary 15.2 suggests that if

$$\int_0^1 (y_n'' - y_n + x)\phi_i(x)dx = 0 \quad \text{for } i = 1, \ldots, n$$

then $y_n'' - y_n + x = 0$, i.e $y_n(x)$ satisfies the differential equation (1). If $y_n(x)$ satisfies the differential equation and the boundary conditions, then we know that $y_n(x)$ approximates the exact solution $y(x)$.

Therefore, we choose the coefficients a_j such that they satisfy the system of equations

$$\sum_{j=1}^{n} a_j \int_0^1 \phi_j''(x)\phi_i(x) - \phi_j(x)\phi_i(x)dx = -\int_0^1 x\phi_i(x)dx \quad \text{for } i = 1, \ldots, n. \tag{4}$$

The above system unnecessarily uses the second derivative of the basis functions. We can rewrite the coefficients of the above system to use only the first derivative of the basis functions. To see this, note that we can rewrite the differential equation (1) in the form

$$(p(x)y')' + q(x)y' + r(x)y = f(x) \tag{5}$$

by choosing $p(x) = 1$, $q(x) = 0$, $r(x) = -1$, and $f(x) = -x$. With this form of the differential equation we would require the approximation (3) to satisfy the following equations

$$\int_0^1 ((p(x)y_n')' + r(x)y_n)\phi_i(x)dx = \int_0^1 f(x)\phi_i(x)dx \quad \text{for } i = 1, \ldots, n.$$

Making use of the fact that the basis functions are 0 on the boundary we see that

$$\begin{aligned}\int_0^1 (p(x)y_n')'\phi_i(x)dx &= \phi_i(x)p(x)y_n'|_0^1 - \int_0^1 p(x)y_n'\phi_i'(x)dx \\ &= -\int_0^1 p(x)y_n'\phi_i'(x)dx.\end{aligned}$$

With this and the definitions of the functions $p(x)$, $r(x)$, and $f(x)$, the system of equations (4) becomes

$$\sum_{j=1}^{n} a_j \int_0^1 -\phi_j'(x)\phi_i'(x) - \phi_j(x)\phi_i(x)dx = -\int_0^1 x\phi_i(x)dx \quad \text{for } i = 1, \ldots, n. \tag{6}$$

Finding the solution to the system of equations (6) identifies the coefficients a_j that define our approximation.

In this instance, we have chosen the basis $\{\phi_j(x)\}_{j=1}^n$ where $\phi_j(x) = x^j(1-x)^j$. Thus,

$$\begin{aligned}\phi_j'(x) &= \left(x^j\right)'(1-x)^j + x^j\left((1-x)^j\right)' \\ &= jx^{j-1}(1-x)^j - jx^j(1-x)^{j-1}\end{aligned}$$

for $j = 1, \ldots, n$.

Using the MATLAB function `approximation.m`, we construct the above system of equations and solve them arriving at approximations to the exact solution for $n = 2, 3, 4$. The tables comparing the exact solution to these approximations at the points $x = 0.25, 0.50, 0.75$ can be found below.

x	$y(x)$	$y_2(x)$	$\lvert y(x) - y_2(x)\rvert$	$\frac{100\lvert y(x)-y_2(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	4.266210e-02	7.614504e-03	2.172618e 01
0.50	5.659056e-02	5.659010e-02	4.598883e-07	8.126590e-04
0.75	5.027579e-02	4.266210e-02	7.613681e-03	1.514383e 01

Table 1: Comparison of approximation y_2 to solution y using basis $\phi_j = x^j(1-x)^j$. All computations are rounded to 6 significant digits.

We also provide the graphs of these comparisons in Figure 1.

The first value of n such that the relative error of the approximation at each of the points $x_0 = 0.25, x_1 = 0.50, x_2 = 0.75$ is less than 5.0e-01% is a number larger than 50. The

x	$y(x)$	$y_3(x)$	$\lvert y(x) - y_3(x)\rvert$	$\frac{100\lvert y(x)-y_3(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	4.266169e-02	7.614092e-03	2.172500e 01
0.50	5.659056e-02	5.659056e-02	5.029993e-10	8.888397e-07
0.75	5.027579e-02	4.266169e-02	7.614093e-03	1.514465e 01

Table 2: Comparison of approximation y_3 to solution y using basis $\phi_j = x^j(1-x)^j$. All computations are rounded to 6 significant digits.

x	$y(x)$	$y_4(x)$	$\lvert y(x) - y_4(x)\rvert$	$\frac{100\lvert y(x)-y_4(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	4.266169e-02	7.614093e-03	2.172500e 01
0.50	5.659056e-02	5.659056e-02	3.451059e-13	6.098294e-10
0.75	5.027579e-02	4.266169e-02	7.614092e-03	1.514465e 01

Table 3: Comparison of approximation y_4 to solution y using basis $\phi_j = x^j(1-x)^j$. All computations are rounded to 6 significant digits.

relative error percents r_{x_i} for the approximation y_{50} at the above points are given by $r_{x_0} =$ 2.172500e01%, $r_{x_1} =$ 3.576125e-09%, and $r_{x_2} =$ 1.514465e01%. You can see that the relative errors at the points x_0 and x_2 are no better than they were for $n = 4$. It also appears that these errors do not monotonically decrease towards 0 so that the approximation does not practically converge to the exact solution.

As the coefficient matrix in the system (6) is dense for this basis, we elect not to go any higher than $n = 50$ as the computational power needed to solve linear systems for dense matrices of large size is higher than the author's available hardware can accommodate.

This suggests that this basis does not give a practical approximation to the exact solution at all points of the interval of definition. However, as the relative error of the approximation is very small for $n = 2$ at the point $x = 0.50$, this suggests that the approximation would be useful for neighborhoods with small radius centered at 0.50.

All programming code used to create the approximations, tables, and graphs for this basis can be found here:

https://github.com/gammadistribution/gradschool/tree/master/MATH635/final/programs □

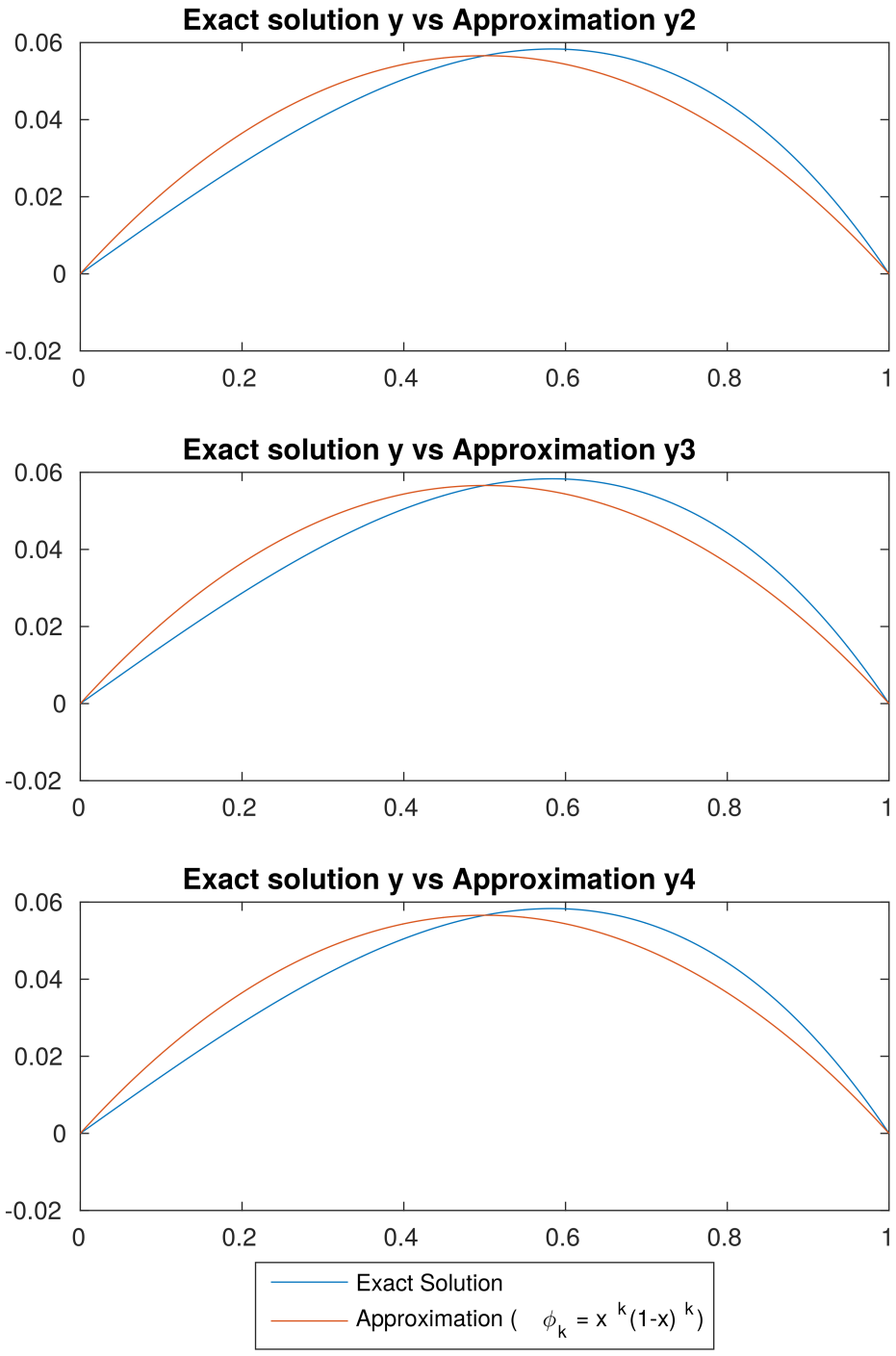

Figure 1: Plots of exact solution y and approximation y_n over the interval $[0, 1]$ using basis $\phi_j = x^j(1-x)^j$.

Problem 3. Repeat problem 2 with the basis $\{\sin(j\pi x)\}$.

Solution. We use the same methods outlined in the solution to problem 2 to obtain an approximation using the basis $\{\phi_j(x)\}$ with $\phi_j(x) = \sin(j\pi x)$. Now we see see that,

$$\phi_j'(x) = j\pi \cos(j\pi x)$$

for $j = 1, \ldots, n$.

Using the MATLAB function `approximation.m`, we construct the above system of equations and solve them arriving at approximations to the exact solution for $n = 2, 3, 4$. The tables comparing the exact solution to these approximations at the points $x = 0.25, 0.50, 0.75$ can be found below.

x	$y(x)$	$y_2(x)$	$\lvert y(x) - y_2(x)\rvert$	$\frac{100\lvert y(x)-y_2(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.355071e-02	1.496893e-03	4.271028
0.50	5.659056e-02	5.856881e-02	1.978250e-03	3.495724
0.75	5.027579e-02	4.927810e-02	9.976905e-04	1.984435

Table 4: Comparison of approximation y_2 to solution y using basis $\phi_j = \sin(j\pi x)$. All computations are rounded to 6 significant digits.

x	$y(x)$	$y_3(x)$	$\lvert y(x) - y_3(x)\rvert$	$\frac{100\lvert y(x)-y_3(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.522118e-02	1.735810e-04	0.495272
0.50	5.659056e-02	5.620640e-02	3.841569e-04	0.678836
0.75	5.027579e-02	5.094857e-02	6.727834e-04	1.338186

Table 5: Comparison of approximation y_3 to solution y using basis $\phi_j = \sin(j\pi x)$. All computations are rounded to 6 significant digits.

We also provide the graphs of these comparisons in Figure 2.

The first value of n such that the relative error of the approximation at each of the points $x_0 = 0.25, x_1 = 0.50, x_2 = 0.75$ is less than 5.0e-01% is given by $n = 6$. The relative errors r_{x_i} at the above points for the approximation y_6 are $r_{x_0} = 3.080284\text{e-}01\%$, $r_{x_1} = 2.293399\text{e-}01\%$, and $r_{x_2} = 2.304205\text{e-}02\%$. This value of n needed for the relative error percents to be less than 5.0e-01% is quite low.

The general method of computing the coefficients suits our needs well enough if the goal is to obtain an approximation with a relative error less than 5.0e-01% at these points. It will be mentioned, however, that the entries a_{ij} of the coefficient matrix found in (6) admit a special structure due to the choice of basis. Namely,

$$a_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ \frac{-j^2\pi^2}{2} - \frac{1}{2} & \text{if } i = j \end{cases}$$

x	$y(x)$	$y_4(x)$	$\lvert y(x) - y_4(x)\rvert$	$\frac{100\lvert y(x)-y_4(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.522118e-02	1.735810e-04	0.495272
0.50	5.659056e-02	5.620640e-02	3.841569e-04	0.678836
0.75	5.027579e-02	5.094857e-02	6.727834e-04	1.338186

Table 6: Comparison of approximation y_4 to solution y using basis $\phi_j = \sin(j\pi x)$. All computations are rounded to 6 significant digits.

This suggests that the coefficient matrix is actually a diagonal matrix. Finding the solution to the system (6) is therefore trivial and reduces the computational complexity of finding the approximation immensely as we only need to compute the entries of the coefficient matrix along the diagonal and the entries of the column vector in the system. Additionally, a similar calculation shows that the integral is not needed for the column vector either as $b_i = (-1)^i/(i\pi)$.

In order to accommodate this special structure we have created conditional creations of the coefficient matrices and column vectors in the code mentioned earlier. □

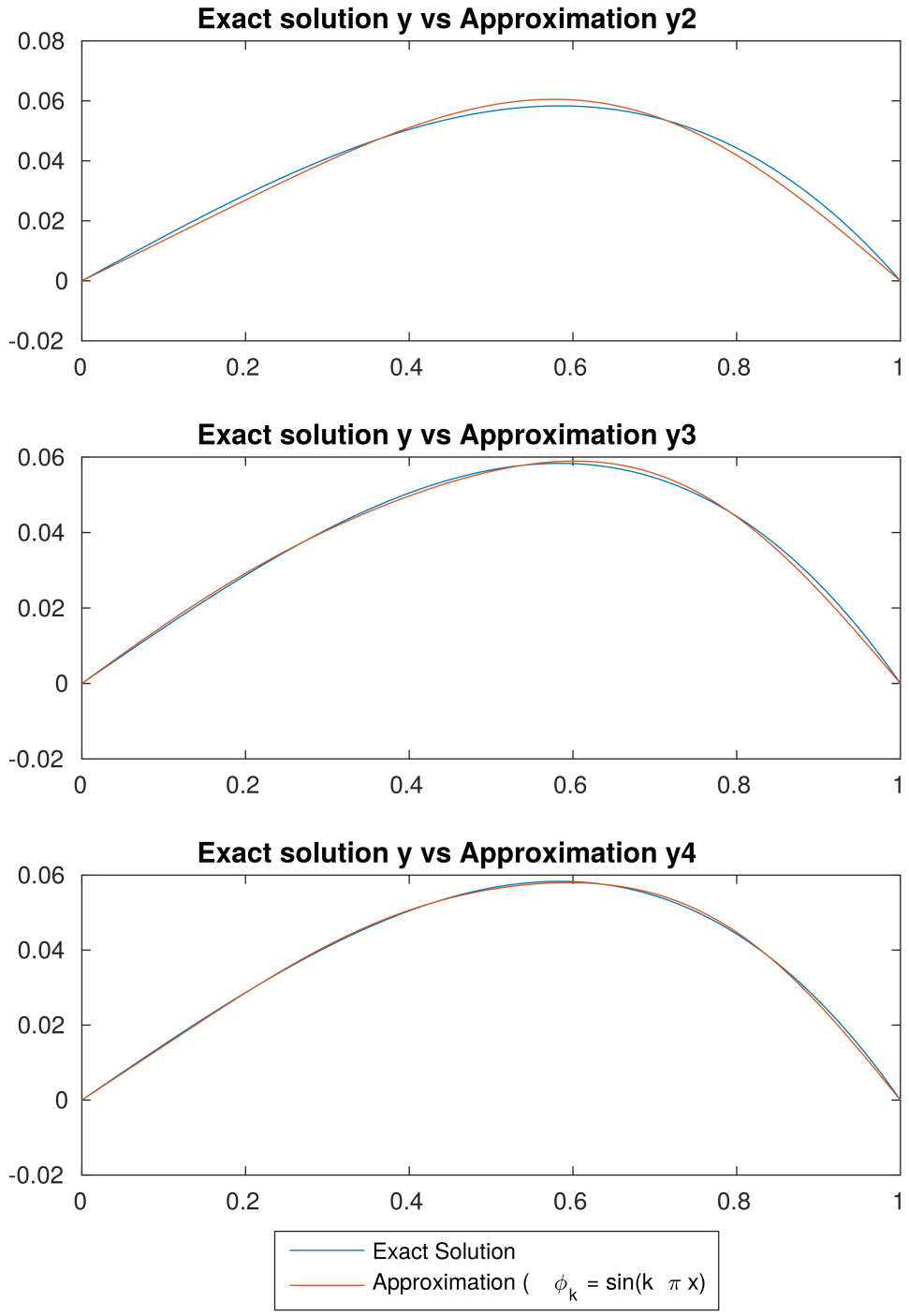

Figure 2: Plots of exact solution y and approximation y_n over the interval $[0, 1]$ using basis $\phi_j = \sin(j\pi x)$.

Problem 4. Repeat the previous problem with the hat function basis (15.51) on p. 502.

Solution. We use the same methods outlined in the solution to problem 2 to obtain an approximation using the basis $\{\phi_j(x)\}$. If n is the number of basis functions to be used in the approximation, define $h = \frac{1}{n+1}$. Then define the basis function as

$$\phi_j(x) := \begin{cases} \frac{x-h(j-1)}{h} & \text{if } (j-1)h \leq x \leq jh \\ -\frac{x-h(j+1)}{h} & \text{if } jh \leq x \leq (j+1)h \\ 0 & \text{otherwise.} \end{cases}$$

Now we see see that,

$$\phi_j(x) := \begin{cases} \frac{1}{h} & \text{if } (j-1)h < x < jh \\ -\frac{1}{h} & \text{if } jh < x < (j+1)h \\ 0 & \text{otherwise.} \end{cases}$$

for $j = 1, \ldots, n$.

Note that from our definition of the basis functions that the entry a_{ij} in the coefficient matrix in the system (6) is 0 if $|i-j| > 1$. This shows that the coefficient matrix is a tri-diagonal matrix and only the sub-diagonal, main diagonal, and super-diagonal need to be computed. Additionally, as it is clear from the way the entries are defined, the coefficient matrix is symmetric and only either the sub-diagonal or super-diagonal needs to be computed.

Thus, there are only two cases two consider, $i = j$ and $i = j+1$. In the first case, we see that for any i,

$$a_{ii} = \int_{(i-1)h}^{ih} -\frac{1}{h^2} - \frac{(x-h(j-1))}{h}dx + \int_{ih}^{(i+1)h} -\frac{1}{(-h)^2} - \frac{(x-h(j+1))}{h}dx \quad = \frac{-2h}{3} - \frac{2}{h}.$$

In the second case, we similarly see that

$$\begin{aligned} a_{i(i-1)} &= \int_0^1 -\phi_i'(x)\phi_{i-1}'(x) - \phi_i(x)\phi_{i-1}(x)dx \\ &= \int_{(i-1)h}^{ih} -\left(\frac{1}{h}\right)\left(\frac{-1}{h}\right) - \left(\frac{x-h(j-1)}{h}\right)\left(\frac{-(x-hj)}{h}\right)dx = \frac{1}{h} - \frac{h}{6} \end{aligned}$$

Therefore,

$$a_{ij} = \begin{cases} \frac{-2h}{3} - \frac{2}{h} & \text{if } i = j \\ \frac{1}{h} - \frac{h}{6} & \text{if } |i-j| = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The column vector entry is computed very similarly and we see that $b_i = \frac{-i}{(n+1)^2}$. These definitions greatly reduce the number of computations needed to compute the coefficient matrix and column vector and the integral is no longer needed.

Using the MATLAB function `approximation.m`, we construct the above system of equations and solve them arriving at approximations to the exact solution for $n = 2, 3, 4$.

The tables comparing the exact solution to these approximations at the points $x = 0.25, 0.50, 0.75$ can be found in Table 7, Table 8, and Table 9. As can be seen from the tables, the first n such that the relative error percent is less than 0.5% at each of the points $0.25, 0.50$, and 0.75 is $n = 3$. Note, however, that for $n = 4$ the relative error percent increases at these points compared to their values for $n = 3$.

x	$y(x)$	$y_2(x)$	$\lvert y(x) - y_2(x)\rvert$	$\frac{100\lvert y(x)-y_2(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.359014e-02	1.457462e-03	4.158520
0.50	5.659056e-02	5.084746e-02	5.743100e-03	1.014851e01
0.75	5.027579e-02	4.268105e-02	7.594738e-03	1.510616e01

Table 7: Comparison of approximation y_2 to solution y using the hat basis. All computations are rounded to 6 significant digits.

x	$y(x)$	$y_3(x)$	$\lvert y(x) - y_3(x)\rvert$	$\frac{100\lvert y(x)-y_3(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.521250e-02	1.648988e-03	4.704996e-01
0.50	5.659056e-02	5.685947e-02	2.689137e-03	4.751917e-01
0.75	5.027579e-02	5.051862e-02	2.428358e-03	4.830075e-01

Table 8: Comparison of approximation y_3 to solution y using the hat basis. All computations are rounded to 6 significant digits.

x	$y(x)$	$y_4(x)$	$\lvert y(x) - y_4(x)\rvert$	$\frac{100\lvert y(x)-y_4(x)\rvert}{\lvert y(x)\rvert}$
0.25	3.504760e-02	3.423311	8.144879e-04	2.323948
0.50	5.659056e-02	5.453670	2.053859e-03	3.629332
0.75	5.027579e-02	4.793311	2.342679e-03	4.659657

Table 9: Comparison of approximation y_4 to solution y using the hat basis. All computations are rounded to 6 significant digits.

We also provide the graphs of these comparisons in Figure 3.

The tri-diagonal nature of the coefficient matrix shows that the matrix is sparse. MATLAB provides optimized functions for solving systems involving sparse matrices. Even though this is not as quick as finding the inverse of a diagonal matrix, it is still quite quick

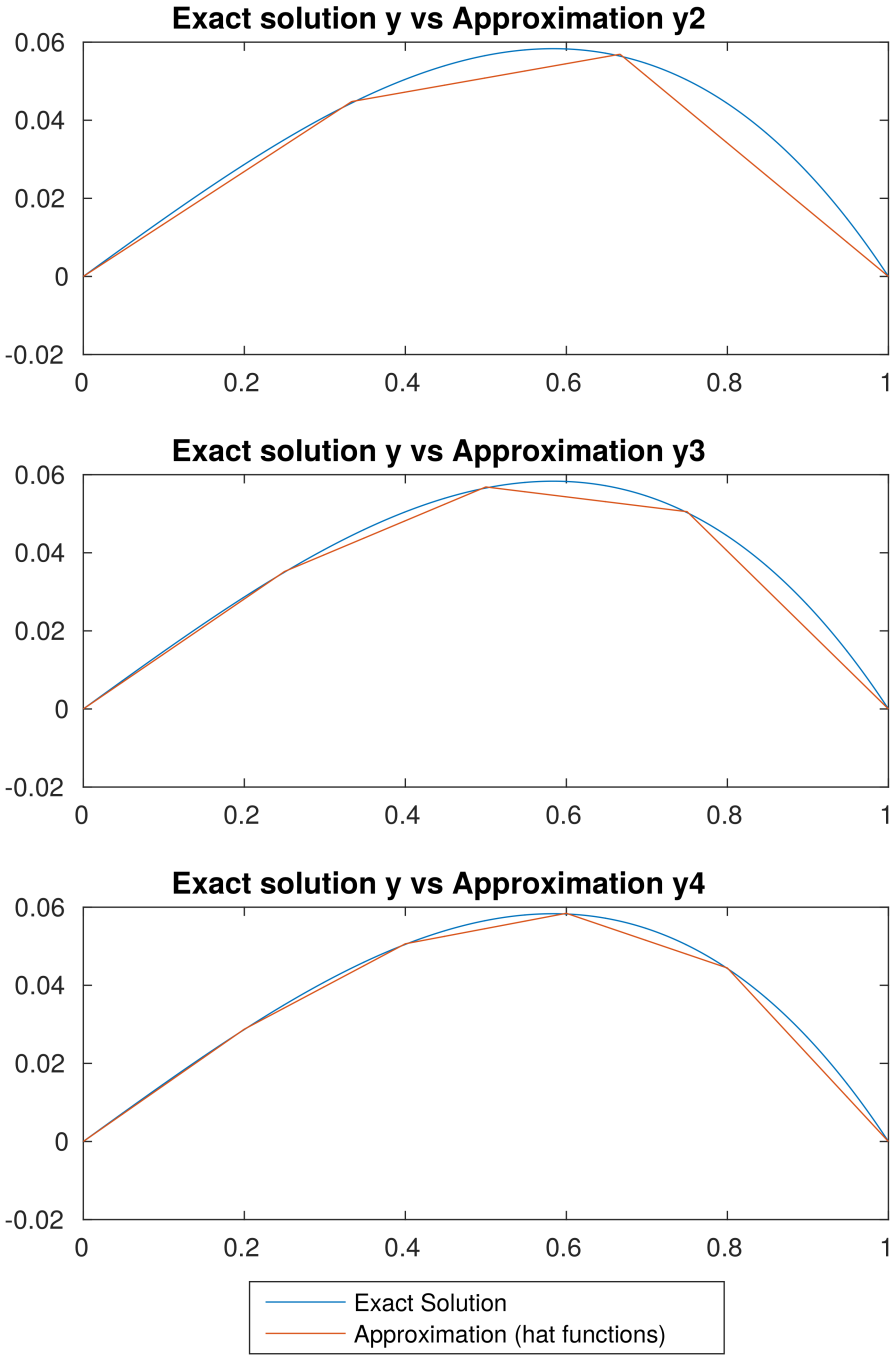

Figure 3: Plots of exact solution y and approximation y_n over the interval $[0, 1]$ using the hat basis.

in MATLAB. This, coupled with the fact that only at most two of the evaluations of the basis functions are needed to compute a point on the approximation, shows that this may

indeed be the best method for approximating the exact solution to this particular differential equation. As such, we have implemented special cases of the algorithm for the creation of the coefficient matrix and column vector as in the trigonometric basis case.

□

Problem 5. Compare the results in problems 2,3,4 and make your recommendation on which method is best.

Solution. We now summarize our findings in problems 2, 3, and 4. There are three things of interest to us when considering which of these approximations is best:

a. If the approximation attains a relative error percent less than 0.5% at the points 0.25, 0.50, and 0.75 as well as throughout the interval [0, 1].

b. Number of basis functions needed to attain a relative error percent less than 0.5% at each of the points 0.25, 0.50, and 0.75 as well as throughout the interval [0, 1].

c. Time and computational power needed to attain a relative error percent smaller than 0.5% at the points 0.25, 0.50, and 0.75 as well as throughout the interval [0, 1].

We will investigate these properties for each of the three basis functions.

From problem 2, we see that the polynomial basis does not practically approximate the exact solution for any point outside of small neighborhoods of 0.5. As such, we cannot recommend this solution as it cannot be used throughout the whole interval of definition in any practical way.

The trigonometric basis provides great approximations with relative error percents less than 0.5% at the points 0.25, 0.50, and 0.75 as well as throughout the interval [0, 1] while also requiring only the first 6 basis functions in order to achieve that accuracy. It also appears that the convergence is uniform in the interval of definition suggesting that this level of accuracy is a global property and the relative error percent will be in close to 0.5% for places not near the boundary. It is also computationally inexpensive to calculate the entries of the column vector and the coefficients of the approximation as the coefficient matrix is a diagonal matrix.

The hat basis only requires 3 basis functions in order to achieve the given relative error percent for the three points. However, as these are linear splines, that accuracy is not attained throughout the interval of defintion and as such it is a local property of convergence. These errors increase on the next iteration and the errors are not uniform in the interval, but it is computationally cheap to compute these approximations. The reason for this is that the coefficient matrix associated to these basis functions is a sparse matrix and by leveraging MATLAB's system of equations solver for sparse coefficient matrices we are able to quickly obtain the coefficients needed for the approximation. Additionally only two of the basis functions are needed to compute a given point of the approximation as all other basis functions will be zero for any given point, reducing the number of computations needed to evaluate an approximation.

In order to obtain more accuracy using the trigonometric basis, you must increase the number of basis functions used in the approximation. This has almost no effect on solving the system of equations, as there is nothing to solve with a diagonal matrix, but computing the values at each of the trigonometric basis functions increases the computational cost of using this approximation as you must involve an increasing number of sine functions. In contrast, the hat basis allows us to quickly obtain more and more accurate approximations requiring more basis functions, but not more computational complexity as only two of those functions are needed to provide an approximation for a given point.

To illustrate this, suppose we are interested in a relative error of 0.0005% at each of the three points 0.25, 0.50, and 0.75. With the trigonometric basis, it only took the first 57 basis functions to attain the desired accuracy but it took 22.012 seconds to check those basis functions and compute the approximations. In contrast, the hat basis functions required the first 123 basis functions, but only took 15.581 seconds to check all other basis functions for this degree of accuracy. Once the number of basis functions needed for the desired accuracy is determined, it is faster to use the basis functions. It only took 2.233 seconds to obtain the approximation and compute a point using the basis function compared to the 3.633 seconds required to obtain the approximation and compute a point. This suggests that using the hat basis will be faster if interested in obtaining an approximation obtaining a relative error less than 0.5%.

In summary, if you are interested in only obtaining a maximum relative error of 0.5% at the given points, you should use the hat basis functions. If you are interested in only obtaining a maximum relative error of 0.5% throughout the interval, you should use the trigonometric basis. To obtain more accurate approximations using the least amount of time and computational power, it is suggested to use the hat basis as it is very cheap computationally to solve the system of equations associated to the problem and to evaluate the approximation for higher numbers of included basis functions. □