Lab 1: Numerical Integration

South Seattle College
Fall 2019 Math 163 with Instructor Rick Downs
Written by: Andrea Wooley and Renae Ford
October 28, 2019

Contents

Introduction

With many functions it is possible to integrate their value over a period by using their antiderivative. However, non-elementary functions do not have an antiderivative and the value of their integral is found using approximation. This project will evaluate the integral of this function *f(x)* from 0 to 1.

$$f(x) = e^{-x^2}$$

$$\int_0^1 e^{-x^2}dx$$

We are using Simpson's Rule and the integration of Taylor Polynomials. The equation defines a curve that is symmetrical across the y-axis with a horizontal tangent at (0, 1) where the curve turns. The curve slopes down on either side until it approaches zero. The first method for solving this integral using Simpson's Rule. This technique uses quadratic polynomials to approximate area to a desired error based on the number of intervals used to model the area. Another way to estimate the integral uses the Taylor series representation of the function to model the curve with Taylor Polynomials then use series to calculate the area created by this model. Increasing the number of terms used in the polynomial reduces the error in our estimation of the integral of the non-elementary function.

Analysis

We evaluated the integral using two methods of approximation and calculated the error appropriately for each method.

Simpson's Rule

Numerical integration using Simpson's Rule requires that we simplify the function using a series to divide the function into a number of sections that represent the average value over that section. This model is more accurate as the number of intervals increases. To calculate the number of sections needed to estimate f(x) with an error less than 0.0001 we solved the error equation for n after deriving the fourth derivative of f(x).

To calculate the error we used this equation:

$$|Es| \le \frac{K(b-a)^5}{180n^4}$$

Where the fourth derivative of the function is equal to or less than K for a ≤x ≤b and the number of intervals used to calculate the area is the even value, n. We found the fourth derivative,

$$f'(x) = (-2x)e^{-x^2}$$
$$f''(x) = -2e^{-x^2} + 4x^2e^{-x^2}$$
$$f'''(x) = 12e^{-x^2} - 8x^3e^{-x^2}$$
$$f^{(4)}(x) = 16x^4e^{-x^2} - 48x^2e^{-x^2} + 12e^{-x^2}$$

To calculate the maximum error we used the maximum value of the fourth derivative on our x-interval, that is when x = 0.

$$f^{(4)}(0) = 16(0)^4e^{-(0)^2} - 48(0)^2e^{-(0)^2} + 12e^{-(0)^2}$$
$$f^{(4)}(0) = 12$$

Next we used K=12, |Es|<0.0001, on the interval 0 ≤ x ≤ 1 in the error equation to calculate the minimum number of intervals possible to achieve our intended accuracy.

$$n \ge \sqrt[4]{\frac{12(1^5)}{180(0.0001)}}$$

Given that the number of intervals is required to be a whole, even number in order for our model to work. We calculated n needs to be 6, 8, 10 or higher to achieve a sufficiently accurate model.

To analyze the accuracy of this model we wrote an algorithm using Matlab that would evaluate our integral by using Simpson's Rule and an input for the number of intervals. The algorithm defines the lower and upper limits of integration and asks for input to define as the number of intervals. Then uses those variables to calculate the width of each interval using the equation from Simpson's Rule. Next, the algorithm calculates the value of f(x) at the left and right endpoints. Next, the area of each interval is calculated and added to the variable for the sum of each interval area. Last the result is defined as the addition of the partial sums for the left endpoint, right endpoint and the sum of each interval between those.

Taylor Polynomials

A Taylor series models a function as a power series, an infinite sum of terms that are calculated from the values of the function's derivatives at a single point. The Taylor series expansion is given by

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

The infinite series should perfectly model the equation however to approximate the integral we use a finite number of terms referred to partial sums which are nth degree Taylor polynomial.

We used Taylor's inequality to find the number of terms by calculating the remainder for the Taylor polynomial.

$$|R_n(x)| \leq \frac{M}{(n+1)!}|x-a|^{n+1}$$

Discussion and Results

Simpson's Rule

After using the algorithm to find approximations of the integral using Simpson's Rule with different numbers of intervals we calculated the error to better understand the approximation. To calculate the error of the resulting value of integral from each Simpson's equation, we recorded the difference between the actual value of the integral, 0.746824132812427, and each resulting value.

The data shows that the error when estimating the integral using 2 intervals is 0.00036, and that error decreases by one tenth when the number of intervals is increased by 2. Using the Simpson's Rule calculation for error we expected the error to be greater than 0.0001 when the number of intervals was less than 6. The algorithms resulting integral using 4 intervals had the error 0.000031. That is less than the error we anticipated. This data shows that the Simpson's approximation of the integral does not have as high an error as we had calculated.

***n* (number of intervals)**	**value of integral**	**error**
2	0.747180428909510	0.0003562960971
4	0.746855379790987	0.00003124697856
6	0.746830391489345	0.000006258676918
8	0.746826120527466	0.000001987715039
10	0.746824948254444	0.000000815442017

Taylor Polynomials

To calculate the error of the resulting value of the integral using each taylor expression, we recorded the difference between the actual value of the integral, 0.746824132812427, and each resulting value.

The data shows that by increasing the number of terms of the Taylor polynomial the error becomes more accurate. Using 4 terms term the error calculated to 0.08015746615 by adding 2 more terms to the Taylor expression our error decreases by approximately 0.06. Using 14 number of terms the error calculated to 0.00009493608323 which is less than what we expected.

# of terms	Taylor expression	Value of integral	Error
4	$1-x^2$	0.666666666666667	0.08015746615
6	$\frac{x^4}{2}-x^2+1$	0.766666666666667	0.01984253385
8	$\frac{-x^6}{6}+\frac{x^4}{2}-x^2+1$	0.742857142857143	0.003966989955
10	$\frac{x^8}{24}-\frac{x^6}{6}+\frac{x^4}{2}-x^2+1$	0.747486772486773	0.0006626396743
12	$\frac{-x^{10}}{120}+\frac{x^8}{24}-\frac{x^6}{6}+\frac{x^4}{2}-x^2+1$	0.746729196729197	0.00009493608323
14	$\frac{x^{12}}{720}-\frac{x^{10}}{120}+\frac{x^8}{24}-\frac{x^6}{6}+\frac{x^4}{2}-x^2+1$	0.746836034336034	0.00001190152361
16	$\frac{-x^{14}}{5040}+\frac{x^{12}}{720}-\frac{x^{10}}{120}+\frac{x^8}{24}-\frac{x^6}{6}+\frac{x^4}{2}-x^2+1$	0.746822806822807	0.00000132598962

Conclusion

The approximation calculated by our algorithm for evaluating integrals using Simpson's Rule resulted in smaller error than we had calculated. The approximation using 4 intervals was as accurate as we had calculated the approximation using 6 intervals would be. Showing that using Simpson's Rule to approximate the integral of a non-elementary function is reasonably accurate without having to use incredibly small intervals. It proves to be a simple way to calculate an area of a shape with a complex curve.

The approximations calculated by our algorithm for evaluating integrals using Taylor Polynomials resulted in complex polynomials with sufficient margins of error. The most simple polynomial within our preferred error had 12 terms.

When comparing the accuracy of our model using Simpson's Rule and our model using Taylor Polynomials it is clear that the more accurate model for our function is that using Simpson's Rule. When approximating a curve it is preferred that the model used is accurate and simple. In the case of integrating our function, the approximation using quadratic polynomials with Simpson's Rule is the more effective model.

References

1. Stewart, James, Calculus Early Transcendentals, 7th ed., Brooks/Cole Cengage Learning, 2012.

Appendix

MATLAB code for calculating the approximate value of the integral using Simpson's Rule.

```
% simpsonint.m
% Evaluates the integral of a function over an interval
% input: function f(x) = exp(-(x)^2)
%           limits of integration: a – lower limit; b – upper limit
% output: the value of the integral

format long  %
clear         %
a=0;           % lower limit of integration
b=1;          %  upper limit of integration
% enter the number of intervals from the keyboard
% Note: for Simpson's rule n must be even
n=input('input number of intervals (n - an even number) ->');

h = (b-a)/n;  %  change in the width of intervals using the endpoints of integration and the
number of intervals
```

```
fa = exp(-(a)^2); % f(a)- function value at left endpoint
fb = exp(-(b)^2); % f(b)-  function value at right endpoint

ff=0;  % the area of each interval will be stored in ff. Start with 0 area

for i = 2:2:n;  % all 4*f(a+nh) terms to f(b) h = (1,3,5,7,...,n-1)
    x = (a+(i-1)*h);    % define x as the initial value of the interval being evaluated
    fx = exp(-(x)^2);   % f(x)- function value at x
    ff = ff + 4*fx;     % the area of the interval is added to the sum of each interval area
end

for i = 3:2:n;  % all 2*f(a+nh) terms to f(b) h = (2,4,6,...,n-2)
    x = (a+(i-1)*h);    %  define x as the initial value of the interval being evaluated
    fx = exp(-(x)^2);   % f(x)- function value at x
    ff = ff + 2*fx;     % the area of the interval is added to the sum of each interval area
end

result = (h/3)*(fa+fb+ff)  % approximated area using partial sums
```

MATLAB code for calculating the approximate value of the integral using Taylor Polynomials.

```
syms x
expr = exp(-(x)^2); % define expression
texpr = taylor(expr, 'Order', 16) % creates taylor polynomial for the expression; change order
number as needed
int(texpr,x,0,1)    % integral of taylor polynomial
```

MATLAB code for calculating the value of the integral

```
xmin=0; %left endpoint of interval
xmax=1; %right endpoint of interval
```

```
fun= @(x)exp(-(x).^2);  %function to be integrated with variable, x
q=integral(fun,xmin,xmax) %integrate function on set interval
```