ECSE 443 – Assignment 3

McGill University

Tristan Bouchard - 260747124

March 27, 2019

Introduction

This assignment required us to implement a few numerical methods to integrate and differentiate. The methods investigated for integration were the Midpoint method, Trapezoidal method and Simpson method, both in one and two variable functions, which are forms of numerical quadrature and cubature, in one and two variables respectively. Some methods are more efficient than others, and they will be explored in this report.

Question 1 – Single variable Integration

This question involved computing the following integral:

$$I = \int_0^{\pi} \ln(5 - 4 * \cos(x))\, dx$$

Using three different methods, with a relative error under 10^{-6} of the true value. The true value of this integral was computed using the built-in MATLAB function *int(),* which yields the following result:

$$I_{true} = 4.355172$$

Part a) Mid-point rule

This question involved performing the integration of the above integral using the mid-point rule. The mid-point method is one of three ways explored in this assignment to perform numerical integration. It consists estimating the value of the function as a box, based at the mid-point of the bounds.

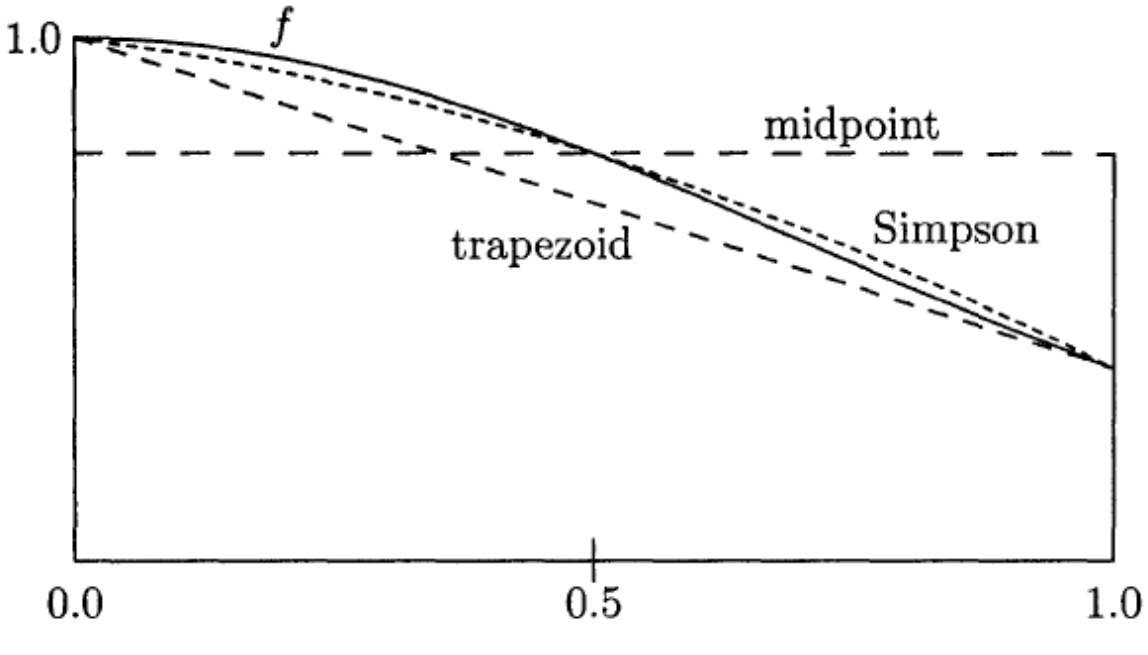

Figure 1 - Numerical Quadrature (taken from textbook)

As can be seen in **Figure 1** above, the crux of the method relies on estimating the integral using a rectangular region, whose height is defined to be the value of the function at the mid-point of the rectangular slice. The mid-point rule, which computed the area of that rectangular region, is thus defined as:

$$M(f) = (b - a) * f(\frac{b + a}{2})$$

To gain higher precision, this method essentially slices up the region to be integrated over into many sub-intervals, which means progressively "skinnier" rectangles which are (hopefully) closer and closer to the actual value of the integral. I have implemented this method in MATLAB, using the mid-point equation above. Computing the value of the integral using this algorithm, to the required relative error of 10^{-6}, yields the following result:

$$I_{midpoint} = 4.3551724802128$$

This result was computed by dividing the integration region into 10 segments of length 0.31416.

Part b) Trapezoid rule

This section makes use of a different method to compute the area under the function. Essentially, it modifies the mid-point method such that the top of the rectangle no longer has a slope of 0, essentially turning the region into a trapezoid, which is where this method gets its name. This can be seen in **Figure 1** above, where the region used to approximate the value of the integral is trapezoidal. The trapezoidal rule is defined as follows:

$$T(f) = \frac{b - a}{2} * (f(b) + f(a))$$

Having implemented this method in MATLAB, the following result is obtained:

$$I_{trapezoid} = 4.35517216500274$$

This result was computed by dividing the integration region into 12 segments of length 0.2618 each.

Part c) Simpson's rule

This section makes use of the Simpson's method to compute the area under the function. It is essentially a combination of the mid-point and trapezoidal rule, but this method adds more weight to the value of the midpoint. This allows the method to more accurately fit the curve of the function, as can be seen in **Figure 1** above. Simpson's rule is defined as follows:

$$S(f) = \frac{b - a}{6}(f(b) + f(a) + 4 * f\left(\frac{b + a}{2}\right))$$

The result I obtain in MATLAB after implementing is:

$$I_{Simpson} = 4.3551722804756$$

This result was computed by dividing the integration region into 10 segments of length 0.31416 each.

Conclusion

In one variable integration, the results for integration are:

True value	Mid-point rule	Trapezoid rule	Simpson's rule
4.355172	4.3551724802128	4.35517216500274	4.3551722804756

These results were computed in approximately the same time, and as such, for now, there is no advantage in selecting one method over the other. However, as can be seen in **Question 2** below, there is a difference when computing the two-variable integral.

Question 2 – Two Variable Integration

This section required us to perform the integration of the two variable function below:

$$\int_{2}^{3}\int_{x}^{2x^3} (x^2 + y)\, dydx$$

Again, we were to use the three methods explored in class, namely the mid-point, trapezoid and Simpson rules, but this time in two variables. Thus, the area being computed in each iteration is now a volume. The strategy to approach this problem is to divide the integration area into a grid of squares, and to compute the volume of the according rectangular prisms formed between the z = 0 plane and the function. Obviously, this rectangular prism will look different based on the method chosen as we will see.

The true value of the integral was computed using the integrated MATLAB function *integral2()*, to which we pass the function handles for the inner integral. The result obtained is the following:

$$I_{trueDouble} = 790.535714$$

Thus, the following sections make use of the numerical cubatures to compute the integral of the region with an absolute error of 10^{-4}.

Part a) Mid-point rule

The method required to approach this problem is very similar to that in one variable. Namely, it is now necessary to use the mid-point of the square area describing the base of the rectangular prism and to compute the value of the function at that point. The mid-point rule is then defined as:

$$M(f) = (b-a)(d-c) * f(\frac{a+b}{2}, \frac{c+d}{2})$$

Where

$$a, b = lower\ and\ upper\ bounds\ in\ x, respectively$$

$$c, d = lower\ and\ upper\ bounds\ in\ y, respectively$$

Which defines the volume of a rectangular prism. Then, it is simply a matter of dividing up the integration area into smaller and smaller squares, until the required precision is obtained. As double

integration can make use of a function for the inner bounds of the function, I made use of function handles in my function to be able to pass the proper bounds to the Y values. My algorithm iterates over divisions in x first, then slices those intervals in y by iterating over them in the nested for loop. The algorithm then computes the value of the integral by adding up the volume of each of the rectangular prisms and compares that value with the true value: If the value is not within the acceptable range, the amount of divisions is increased, and the process restarts. This process continues until the proper precision is obtained.

The result obtained for this integral is:

$$I_{midDouble} = 790.535614364065$$

This result has an absolute error $= -9.9922 * 10^{-5}$, and was computed using 1149 segments in x and y of area 0.00087032 in x. The time to compute the value of the integral for this amount of divisions is equal to 1.332739 seconds. Obviously, it takes quite a bit of time to iterate over the amount of divisions, but also to iteratively increase the divisions in hope of finding the lowest value of divisions. Such, this algorithm to get a good precision of 10^{-4} requires a LOT of time to run.

Part b) Trapezoid rule

The method to compute the integral is the same as in **Part a)**, except that this method makes use of the Trapezoid cubature rule instead. As such, it still iterates over x first, then y, using function handles to properly set the bounds in y. The Trapezoid rule in two-variable is:

$$T(f) = \frac{1}{4} * [(b-a) * (\Delta y_{lo}) * \big(f(a,c) + f(a,c + \Delta y_{lo})\big)$$

$$+ (b-a) * (\Delta y_{hi}) * (f(b,c) + f(b,c + \Delta y_{hi})]$$

Where

$$a, b = lower\ and\ upper\ bounds\ in\ x, respectively$$

$$c, d = lower\ and\ upper\ bounds\ in\ y, respectively$$

$$\Delta y_{lo} = Size\ of\ the\ division\ in\ y\ for\ x = a$$

$$\Delta y_{hi} = Size\ of\ the\ division\ in\ y\ for\ x = b$$

Keep in mind here that here, the limits of the bounds in y for each x are computed using the function handles, namely

$$y_{lower} = x$$

$$y_{upper} = 2x^3$$

and the values between these bounds are divided into the same amount of divisions in x. Because the divisions at x = a are not necessarily the same length as at x = b, it is necessary to define $\Delta y_{lo}\ and\ \Delta y_{hi}$.

Using this method, I obtained the following result:

$$I_{trapDouble} = 790.5358141989278$$

This result was computed in 11.446891 seconds with 1625 segments in x and y of length 0.00061538 in x. Note that the time to compute this value is the time it took ONLY to find the integral with 1625 divisions in x and y, NOT the time it took to iterate from 1 to 1625 divisions, computing the integral at all of those points. Thus, it takes quite a bit of time for this algorithm to converge, which renders it even less practical than the mid-point method.

Part c) Simpson's rule

This section required us to extend Simpson's rule to two variable integration. The approach is identical to the methods explored above, differing only on how the volume of the rectangular prism is computed. Simpson's rule, as in one variable, makes use of both the mid-point and trapezoid rules, placing more weight on the mid-point value. Well, it turns out that the approach is very similar in two dimensions, except that it places more emphasis on the center *slice* in y, which is then also divided into a single variable integration placing more weight on the center of that slice. The weighting of each point is as follows:

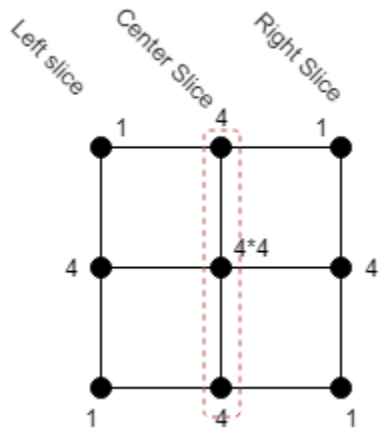

Figure 2 - Simpson's rule weighting

Then, Simpson's rule in two variables is defined as:

$$S(f) = \frac{1}{36}[(\Delta x) * (\Delta y_{lo}) * \left(f(a, c_{lo}) + f(a, c_{lo} + \Delta y_{lo}) + 4 * f\left(a, c_{lo} + \frac{\Delta y_{lo}}{2}\right)\right)$$

$$+ 4 * (\Delta x) * (\Delta y_{mid})\left(f\left(a + \frac{\Delta x}{2}, c_{mid} +\right) + f(a+, c_{mid} + \Delta y_{mid}) + 4 * f\left(a + \frac{\Delta x}{2}, c_{mid} + \frac{\Delta y_{mid}}{2}\right)\right)$$

$$+ (\Delta x) * (\Delta y_{hi}) * \left(f(a + \Delta x, c_{hi}) + f(a + \Delta x, c_{hi} + \Delta y_{hi}) + 4 * f\left(a + \Delta x, c_{lo} + \frac{\Delta y_{hi}}{2}\right)\right)]$$

Where

$$\Delta x = step\ size\ in\ x$$

$$\Delta y_{lo} = step\ size\ in\ y\ for\ x = a$$

$$\Delta y_{mid} = step\ size\ in\ y\ for\ x = a + \frac{\Delta x}{2}\ (aka\ x_{mid})$$

$$\Delta y_{hi} = step\ size\ in\ y\ for\ x = a + \Delta x$$

Where the division by 36 at the end is for balancing the weighting of each point contributing to the volume. This method fits the curvature of the surface best, and as such we expect it to perform the fastest. The result I obtained for this method is:

$$I_{simpDouble} = 790.5358006844086$$

This method was computed with an absolute error of $8.6398 * 10^{-5}$, and computed in 12 segments in x and y, with the x segments having a size of 0.083333. This value was computed, for 12 segments, in 0.019307 seconds. This method is MUCH faster than the other two, not only in the time it takes to compute and individual integral, but with the speed at which it converges to a very low error answer. This is because this method makes use of better weighting to compute the fit of the function.

Conclusion

This second section, where we integrate over two variables, we can very clearly see that the Simpson rule is far superior over the mid-point and the trapezoid method, as it converges to a precise answer with much less divisions (12 instead of 1149 or 1625, respectively).

Question 3 – Finite differentiation

Part a) Finite backward difference

Knowing that the backward difference is defined to be:

$$f'(x) \cong \frac{f(x) - f(x-h)}{h}$$

It is possible to compute the fifth finite difference:

$$f^5(x) \cong \frac{f^4(x) - f^4(x-h)}{h}$$

$$\cong \frac{\frac{f^3(x) - f^3(x-h)}{h} - \frac{f^3(x-h) - f^3(x-2h)}{h}}{h}$$

$$= \frac{f^3(x-2h) - 2f^3(x-h) + f^3(x)}{h^2}$$

$$\cong \frac{\frac{f^2(x-2h) - f^2(x-3h)}{h} - 2 * \frac{f^2(x-h) - f^2(x-2h)}{h} + \frac{f^2(x) - f^2(x-h)}{h}}{h}$$

$$= \frac{-f^2(x-3h) + 3f^2(x-2h) - 3f^2(x-h) + f^2(x)}{h^3}$$

$$= \frac{f^2(x) \; - \; 3f^2(x-h) \; + 3f^2(x-2h) + -f^2(x-3h)}{h^3}$$

$$= \frac{(f^1(x) - f^1(x-h)) - 3\big(f^1(x-h) - f^1(x-2h)\big) + 3\big(f^1(x-2h) - f^1(x-3h)\big) - (f^1(x-3h) - f(x-4h))}{h^4}$$

$$= \frac{f^1(x) - f^1(x-h) - 3f^1(x-h) + 3f^1(x-2h) + 3f^1(x-2h) - 3f^1(x-3h) - f^1(x-3h) + f(x-4h)}{h^4}$$

$$= \frac{f^1(x) - \; 4f^1(x-h) + 6f^1(x-2h) - 4f^1(x-3h) + f(x-4h)}{h^4}$$

$$\cong \frac{f(x) - f(x-h) - 4\big(f(x-h) - f(x-2h)\big) + 6\big(f(x-2h) - f(x-3h)\big) - 4\,\big(f(x-3h) - f(x-4h)\big) + \big(f(x-4h) - f(x-5h)\big)}{h^5}$$

$$= \frac{f(x) - 5f(x-h) + 10f(x-2h) - 10\,f(x-3h) + 5f(x-4h) - f(x-5h)}{h^5}$$

Part b) Finite forward difference

From the Taylor series, we know that:

$$f(x+h) = f(x) + \; f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \cdots$$

$$f(x+2h) = f(x) + \; f'(x)2h + \frac{f''(x)}{2}4h^2 + \frac{f'''(x)}{6}8h^3 + \cdots$$

$$f(x+3h) = f(x) + \; f'(x)3h + \frac{f''(x)}{2}9h^2 + \frac{f'''(x)}{6}27h^3 + \cdots$$

We can solve for $f'(x)$ by summing these equations together by multiplying theme to get a common denominator for f'(x), namely $18 * f(x+h)$, $-9f(x+2h)$, $2 * f(x+3h)$:

$$\begin{aligned} 18f(x+h) - 9f(x+2h) + 2f(x+3h) \\ &= 18f(x) + 18f'(x)h + 9f''(x) + \; 3f'''(x) - 9f(x) - 18f'(x)h - 18f''(x)h^2 \\ &\quad - 12f'''(x)h^3 \; + \; 2f(x) + \; 6f'(x)h + \; 9f''(x) * h^2 + \; 9f'''(x)h^3 \end{aligned}$$

$$= \; 11f(x) + 6f'(x)h$$

Therefore,

$$f'(x) = \frac{18f(x+h) - 9f(x+2h) + 2f(x+3h) - 11f(x)}{6h}$$

Question 4 – Finite forward difference

First, I will derive the equations. As above, it is necessary to use the Taylor series expansion for:

$$f(x+h) = f(x) + \; f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \cdots$$

$$f(x+2h) = f(x) + f'(x)2h + \frac{f''(x)}{2}4h^2 + \frac{f'''(x)}{6}8h^3 + \cdots$$

Adding these equations together, $f(x+2h) - 4 * f(x+h)$, and keeping only the terms in h^2:

$$f(x+2h) - 4 * f(x+h) = f(x) + 2hf'(x) + 2f''(x) - 4f(x) - 4hf'(x) - 2h^2 f''(x)$$

$$= -3f(x) - 2hf'(x)$$

$$f'(x) = \frac{f(x+2h) - 4 * f(x+h) + 3f(x)}{-2h} = \frac{4f(x+h) - f(x+2h) - 3f(x)}{2h}$$

Similarly, using backward differences:

$$f'(x) = \frac{f(x-2h) - 4f(x-h) + 3f(x)}{2h}$$

Then, it is possible to find the second order finite derivative, second order accurate, by performing $f(x+2h) - 8f(x+h)$:

$$f(x+2h) - 8f(x+h)$$

$$= f(x) + 2hf'(x) + 2h^2 f''(x) + \frac{8h^3}{6}f'''(x) - 8f(x) - 4hf'(x) - 4h^2 f''(x) - \frac{8h^3}{6}f'''(x)$$

$$= -7f(x) - 6hf'(x) - 2h^2 f''(x)$$

$$f''(x) = \frac{-f(x+2h) + 8f(x+h) - 7f(x) - 6hf'(x)}{2h^2}$$

Using this equation, we can find the requested derivatives, using MATLAB:

f'(0)	f'(2)	f'(4)	f''(0)
7	-7	-43	-8