Assignment 2: Numerical Integration Part 1
due: Feb 11, 2021

GOAL: This is the introduction to numerical integration using some simple methods based on equal length integrals. They will prepare for future Gaussian integration (or parameter tuning) against test data as a primitive of *"Machine Learning"* . Additional background material is in Chapter 3 of the class notes.

1 Background

The relation between integrals and derivatives is give by

$$F(x) = I[f] = \int^x f(y)dy \implies D[F(x)] = \frac{dF(y)}{dy}\Big|_x = f(x) \tag{1}$$

Note we are being careful her to disguise the **dummy** integration variables/differentiation variables for evaluations. This is really correct even though this is often written in the notation,

$$F(x) = I[f] = \int^x f(x)dx \implies D[F(x)] = \frac{dF(x)}{dx} = f(x) \tag{2}$$

This is exactly the same as using the same name for a dummy variable in declaration of a function in a C/C++/Fortran versus it use in a calling it– referred to as scoping rules!

Both are linear relations. So formally this is $D[\ I[\ fx)]\ ] = f(x)$ on a function $f(x)$, so $D[..]$ is a left inverse of $I[..]$. Why do I say left inverse because strictly speaking d/dx is the inverse of the integral $\int dx$ but not vise a versa since $F(x) + c$ has the same derivative independent of c. $I[\ D[\ F(x)]\ ]$ is define only up to an unknown constant. To arbitrary fix the constant (assuming the area under $f(y)$ is finite in an interval starting at $y = a$ up to x, we can introduce a definite integral

$$F_a(x) = I[f, a] = \int_a^x f(y)dy \implies D[F(x)] = \frac{dF(y)}{dy}\Big|_x = f(x) \tag{3}$$

(To impress you it is called the **Fundamental Theorem of the Calculus. Big deal !** : https://en.wikipedia.org/wiki/Fundamental_theorem_of_calculus Now we begin to formulate discrete version of this theorem .

The simplest approximation is the (central) Riemann sum over N rectangles of width $h = (b-a)/N$

$$\int_b^a f(x)dx \simeq \sum^{N-1} hf(a + (i + 1/2)h) \tag{4}$$

Let's also take $a = x = Nh, b = 0$ and consider the left and right sides of the rectangles:

$$\widetilde{I}_h[f(x)] = \sum_{i=0}^{x/h-1} hf(ih) = h[f(x-h) + f(x-2h) + \cdots + f(h) + f(0)] \tag{5}$$

$$I_h[f(x)] = \sum_{i=1}^{x/h} hf(ih) = h[f(x) + f(x-h) + \cdots + f(2h) + f(h)] \tag{6}$$

respectively.

Now we can do a modest improvement [1].We do this considering a $2h$ interval to find a better approximation on a each 2h interval (for simplicity now we always assume even number for N.) In this form on a 2h interval, the **central Riemann** is expressed as

$$\int_{-h}^{h} f(x)dx \simeq h\left[f(-h/2) + f(h/2)\right] \tag{8}$$

and the **Trapezoidal** rule as

$$\int_{-h}^{h} f(x)dx \simeq h\left[\frac{1}{2}f(-h) + f(0) + \frac{1}{2}f(h)\right] \tag{9}$$

respectively. Now **Simpson** rule gives a different weight

$$\int_{-h}^{h} f(x)dx = h\left[\frac{1}{3}f(-h) + \frac{4}{3}f(0) + \frac{1}{3}f(h)\right] \tag{10}$$

and a **3 term Gaussian** form another,

$$\int_{-h}^{h} f(x)dx = \frac{h}{9}\left[5f(-h\sqrt{3/5}) + 8f(0) + 5f(h\sqrt{3/5})\right] \tag{11}$$

Each of the $2h$ forms can be repeated $N/2$ time to fill and arbitrary interval $[a, b]$ with $h = (b-a)/N$. For example the trapezoidal rule is

$$\sum_{n=0}^{N/2-1} h[\frac{1}{2}f(a + 2nh) + f(a + (2n+1)h) + \frac{1}{2}fa + (2n+2)h)] \tag{12}$$

2 Written Exercise

Work out and pass in on paper:

1. Show that $\Delta_h \widetilde{I}_h[f(x)] = f(x)$. Show as well that $\widetilde{\Delta}_h I_h[f(x)] = f(x)$.

2. Show the trapezoidal rule is

$$I_h^{trap}[f(x)] = \frac{1}{2}(I_h[f(x)] + \widetilde{I}_h[f(x)]) \tag{13}$$

[1] We are playing with weights and position in the integrand as warm up for a general Gaussian adaptation,

$$\int_{-1}^{1} f(x)dx \simeq \sum_{i=1}^{N} w_i f(x_i) + O(h^{2N}) \tag{7}$$

method that cleverly chooses N weights w_i and positions x_i to optimize convergence. In the above equation we mapped $[a, b]$ limits for convenience to a standard interval $[-1, 1]$.

3. Give your best estimate for the size of the error term $O(h^k)$ for the two $2h$ interval for central Riemann, Trapezoidal, Simpson and Gaussian 3 term rule.

HINT: On last question above we can test them against each term Taylor expansion $1, x, x^2, x^3, \cdots$ on the 2h interval against the exact integrals:

$$\int_{-h}^{h} x^n dx = h^{n+1} \frac{1-(-1)^{n+1}}{n+1} \qquad n = 0, 1, 2, \cdots \tag{14}$$

(Odd n term are zero!) The frist term that fails is the error.

3 Coding Exercises

3.1 Exercise #1 – One Dimensional Integrals

Integrate a few function with all four methods (Riemann, Trapezoidal, Simpson and 2 term Gaussian). Write a single c code called, `integrate_examples.cpp`.

$$\begin{aligned} \int_{-1}^{1} x^8 \; dx &= ? \\ \int_{-1}^{1} \cos(\pi x/2) \; dx &= ? \\ \int_{-1}^{1} \frac{1}{x^2+1} \; dx &= ? \end{aligned} \tag{15}$$

Write a single main program `test_integrate.c` that does all 4 cases for a range of values of $N = 4$ to large $N = O(2^{20})$. Plot the error to see if you can verify the error estimates above for $h = 2/N$. **NOTE: In this exercise the integrals are all done by a sequence of intervals of $2h$ so the number of intervals N of h length h is even.**

You may want to try other functions, but only for your own enjoyment. (A really crazy example that may well fail dramatically is $\int_{-1}^{1} \cos(1/x)\, dx$, although the value is -0.168821901119148 according to Mathematica.)

There is an example code `integrate_sin.cpp` to get you started. The code should put out tables so that it is easy to plot the results. You should plot all 3 integrals together against $log(N)$ so there are only 4 plots in all.

While you are required to submit the code, the important deliverable is a plots of the relative error between approximate and the "exact" answer given by Mathematica:

- `Integrate[x^8, {x, -1, 1}]]` `N[Integrate[x^8, {x, -1, 1}],15]`
- `Integrate[Cos[Pi x/2], {x, -1, 1}]` `N[Integrate[Cos[PI x/2], {x, -1, 1}],15]`
- `Integrate[1/(x^2 + 1), {x, -1, 1}]` `N[Integrate[1/(x^2 + 1), {x, -1, 1}],15]`

as a function of the number of points N. (Don't confuse this N with the `N` in the Mathematica commands—the latter forces Mathematica to print a numerical solution! You may share these exact result with other students.) Remember, the relative error is defined as the ratio:

$$(\text{exact} - \text{approximate})/\text{exact}$$

See the Mathematica Notebook on `GitHub` in `HW2code/n04Integration` that does this for $f(x) = \sin(x)$ on the interval $x \in [0,1]$. You can play with this BUT in this problem you must submit your own C code that does these integrals.

3.2 Exercise #2 – Two Dimension Integrals

Now it is easy to generalize to two dimension integrals. For simplicity they have all been mapped into a square. The result is a double sum (or nested for loops):

$$\int_{-1}^{1} dy \int_{-1}^{1} dx \; g(x,y) \simeq \sum_{j=1}^{N} w_j \sum_{i=1}^{N} w_i g(x_i, y_j) \tag{16}$$

Write a main program `test_integrate_2d.c` that performs the following three integrals using the Trapezoidal rule and Gaussian integration rule above for $N = 2$ to 2^{10} on each axis. For this example you only need to report the best estimate for each. (Figures are optional.)

$$\int_{-1}^{1} dy \int_{-1}^{1} dx \; [x^8 + y^8 + (y-1)^3(x-3)^5] = ? \tag{17}$$

$$\int_{-1}^{1} dy \int_{-1}^{1} dx \; \sqrt{x^2 - y^2 + 2} = ? \tag{18}$$

$$\int_{-1}^{1} dy \int_{-1}^{1} dx \; [e^{-x^2 - \frac{y^2}{8}} \cos(\pi x)/(1 + x^2 + y^2)] = ? \tag{19}$$

$$\tag{20}$$

(**Notation-Notation!** Many people (myself included) think it is nicer to write integration as $\int dx[...]$ rather than the awkward convention $\int [...]dx$. Easier to see it as **operator on the left on functions and naturally converted to nested for loops in code!**)

3.3 Submitting Your Assignment

Submit written part as a pdf file and the source coding assignment with the Makefile and output analysis to the directory tt /projectnb/username/HW2 on your CCS account. Please check that source code actually compiles and runs properly on the CCS before placing it in this directory. Have some output but not necessarily all you use do the analysis.