Assignment 2: From Integrals to Sums

Submit codes in the directory `/projectnb/ec526/students/yourloginname/HW2` and the written part and output figures in subdirectory `/projectnb/ec526/students/yourloginname/HW2/doc` on your SCC account by Feb 9, 2024, 11:59PM.

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 a test data as a kind of analytical version of *"Machine Learning"*)

1 Background

The relation between integrals and derivatives is give by

$$F(x) = I[f] = \int_0^x f(y)dy \implies D[F(x)] = \frac{dF(y)}{dy}\Big|_x = f(x) \tag{1}$$

(Note we are being careful here to distinquish 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 $\int$ but not vise a versa since $F(x)+c$ is has the same derivative independent of c. $I[\ D[\ F(x)]\ ]$ is define only up to an unknown constant! (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 rectangle of width $h = (b-a)/N$

$$\int_b^a f(x)dx \simeq \sum_{i=0}^{N-1} hf(a + (i + 1/2)h) \tag{3}$$

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

$$\widetilde{I}_h[f(x)] = \sum_{i=0}^{x/h-1} hf(ih) = h[f(x-h) + f(x-2h) + \cdots + f(h) + f(0)] \tag{4}$$

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

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

$$\int_{-h}^{h} f(x)dx \simeq h\,[f(-h/2) + f(h/2)] \tag{7}$$

and the **Trapezoidal** rule as

$$\int_{-h}^{h} f(x)dx \simeq h\left[\frac{1}{2}f(-h) + f(0) + \frac{1}{2}f(h)\right] \tag{8}$$

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

$$\int_{-h}^{h} f(x)dx = h\left[\frac{1}{3}f(-h) + \frac{4}{3}f(0) + \frac{1}{3}f(h)\right] \tag{9}$$

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

$$\int_{-h}^{h} f(x)dx = \frac{h}{9}\left[5f(-h\sqrt{3/5}) + 8f(0) + 5f(h\sqrt{3/5})\right] \tag{10}$$

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

$$\sum_{n=0}^{N/2-1} h[\frac{1}{2}f(a+2nh) + f(a+(2n+1)h) + \frac{1}{2}fa+(2n+2)h)] \tag{11}$$

Now if you want you can change $2h$ to h. This make make the formula look nicer – **Remember this is up to you. The physical distance is always the same $b-a$ in the end! The grid points have with uniform spacing are separated by intervals: $\Delta x = (b-a)/N$.**

2 Written Exercise

Work out and put in `/projectnb/ec526/students/yourloginname/HW2/doc`

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

[1] We are playing with weights and position in preparation for a general Gaussian adaptation,

$$\int_{-1}^{1} f(x)dx \simeq \sum_{i=1}^{N} w_i f(x_i) \tag{6}$$

will choose cleverly N weights w_i and positions x_i in the standard interval $[-1,1]$ form scaled by h.

2. Show the trapezoidal rule is

$$I_h^{trap}[f(x)] = \frac{1}{2}(I_h[f(x)] + \widetilde{I}_h[f(x)]) \tag{12}$$

3. Give you 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:

$$\int_{-h}^{h} x^n dx = h^{n+1}\frac{1-(-1)^{n+1}}{n+1} \quad n = 0, 1, 2, \cdots \tag{13}$$

(Odd n term are zero!) The first 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`.

$$\begin{aligned} \int_{-1}^{1} x^8 \; dx &=? \\ \int_{-1}^{1} \cos(\pi x/2) \; dx &=? \\ \int_{-1}^{1} \frac{1}{x^2+1} \; dx &=? \end{aligned} \tag{14}$$

Write a single main program `test_integrate.c` that does all 4 cases for a range of values of $N = 4$ to large N maybe as large as $N = 2^{20}$. Plot the error to see if you can verify the error estimates above for $h = 2/N$.

You may want to try other functions, but only for your own enjoyment. (A really crazy example that may well fail numerically 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 (exact − approximate)/exact.

Next week in class will help you to write a nicer Mathematica notebook that imports the data form the C code and compares with the analytical results. BUT in this problem you only submit the own C code that does these integrals numerically.

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):

$$\int_{-1}^{1} dy \int_{-1}^{1} dx\; g(x,y) \simeq \sum_{j=1}^{N}\sum_{i=1}^{N} w_{i,j}\, g(x_i, y_j) \tag{15}$$

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.)

$$\int_{-1}^{1} dy \int_{-1}^{1} dx\; [x^8 + y^8 + (y-1)^3(x-3)^5] = ? \tag{16}$$

$$\int_{-1}^{1} dy \int_{-1}^{1} dx\; \sqrt{x^2 - y^2 + 2} = ? \tag{17}$$

$$\int_{-1}^{1} dy \int_{-1}^{1} dx\; [e^{-x^2-\frac{y^2}{8}} \cos(\pi x)\sin(\frac{\pi}{8}x)] = ? \tag{18}$$

$$\tag{19}$$

(**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!**)