NUI Galway
Programming and Tools for AI (CT5132/CT5148)

James McDermott

September 22, 2021

Assignment 1: Numerical Integration

Weighting 15% of the module

Deadline Midnight Sunday 10 October

Note Read this spec carefully. E.g., if you submit the wrong file format or filename, or put your interpretation comment in the wrong place, or fail to write your student name(s)/ID(s), you will lose marks!

Background As we know, integration is one of the fundamental tools of calculus, and it is used in many AI applications. Everything we need to know is in this pdf and the skeleton code. We do not need to revise understanding of integration for this assignment.

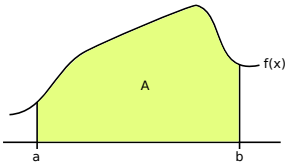

Figure 1: Integration gives area under a curve f between endpoints a and b: $A = \int_a^b f(x)dx$.

Often, it is difficult to calculate the true integral of a complicated function, but *numerical integration* can be used to give an approximation.

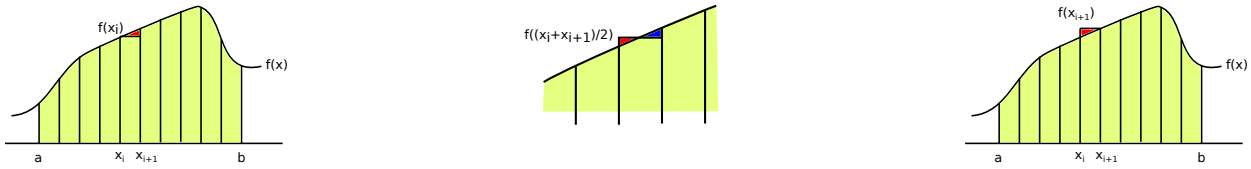

Figure 2: Three schemes for numerical integration. The area under $f(x)$ between a and b is equal to the sum of areas between x_i and x_{i+1} for all i. The area under $f(x)$ between x_i and x_{i+1} can be approximated by a rectangle of height $f(x_i)$, where x_i is always at the left of the rectangle. This is a "left rectangular" scheme, as shown on the left-most image. The error in the approximation is shown in red. Alternatively it can be approximated by a rectangle of height $f(x_{i+1})$, a "right rectangular" scheme, as shown in the right-most image. Zooming in, the area under $f(x)$ between x_i and x_{i+1} can instead be approximated by a rectangle of height $f((x_i + x_{i+1})/2)$, a "midpoint" scheme. In this case there are two areas of error as shown in the centre image.

Requirements Start with the `numint.py` file which is attached. It contains complete implementations of some functions and skeletons for others. If you run `numint.py` (before editing), doctests will run and will report some failures. Fix each of these by adding code to the `numint_py()`, `numint()`, `numint_err()` and `make_table()` functions. (Do it in this order, as they build on each other.) Read the doctests to help you see how the functions fit together and especially to see how `numint` matches the mathematical formula. Do not change the doctests.

Once your code is working correctly, add code in `main()` which calls `make_table()` to print out a table of the true values of the integrals below, and the absolute and relative errors. The "true value" means the area $A = \int_a^b f(x)dx$. In school, we learned how to work out this value on paper, but here we don't need to. The supplied function `true_integral` calculates it using Sympy. The *absolute error* means the absolute difference between A and the approximation given by our numerical integration function. The *relative error* means the absolute error divided by the absolute value of the true value.

The table should consider the following three integrals:

- $\int_0^\pi \cos(x)dx$
- $\int_0^1 \sin(2x)dx$
- $\int_0^1 e^x dx$

It should consider $n = 10, 100, 1000$, and schemes `left` and `midpoint`.

Below `main()`, write a short comment (e.g. 5 lines) interpreting your results: which scheme is better? By how much (with reference to n? Why?

Finally, design and implement an n-dimensional version, `numint_nd`, where f will be a function of multiple arguments. a should be a tuple of m values indicating the lower bound per dimension, and similarly for b and n. You may use pure Python, or Numpy, and any of the three integration schemes, as you prefer. Include some doctests. **Note** this part is more difficult and abstract than the rest of the assignment. If you complete the rest of the assignment and do not attempt this part, a high B grade is still possible.

Submit one file, named `numint.py`. It should include your name and student number as directed at the top.

Marks will be awarded for correct code, for understanding in the interpretation, and for good code style including spacing, variable names, appropriate comments.

Groups Students may work solo or in groups of 2, as desired. Students who wish to form a group but do not yet have a partner may post on the Discussion Board. In groups of 2, only 1 student need submit the file. In any communication concerning the assignment, students must cc all group members.

Plagiarism Students are reminded of the University's policy on plagiarism. Students may discuss the assignment with other students/groups but must not look at other students'/groups' work, or allow others to look at theirs. Any online sources used must be cited with URL and date of access in a comment. Materials from CT5132/CT5148 need not be cited. By writing your name and ID in the .py file and submitting it, you declare that you have abided by these conditions. Students may be called to interview to discuss their submissions. Suspected infringements will be investigated, and may be referred to NUI Galway authorities.