Numerical Analysis 1 – Class 11

Thursday, April 1st, 2021

Subjects covered

- Numerical integration and Newton-Cotes methods in 1D.
- Gaussian quadrature in 1D.
- Clenshaw-Curtis and other quadrature methods.
- Gaussian quadrature in 2D.
- Integrating odd 2D shapes using a triangular mesh.

Reading

- Kutz, Chapt 4.2 – 4.3.
- Chapter on quadrature from "Numerical Computing with MATLAB", C. Moler (linked on Canvas).
- “The surveyor's area formula”, B. Braden (on Canvas)

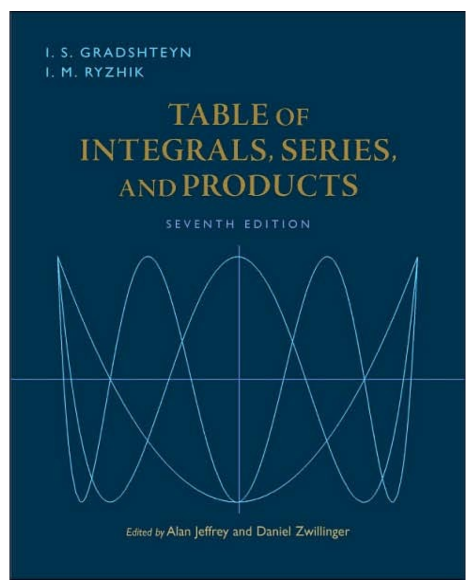
Problems

Most of the following problems require you to write a program. For each program you write, please make sure you also write a test which validates your program. Please use Canvas to upload your submissions under the “Assignments” link for this problem set.

Problem 1

Consider the definite integral

$$\int_0^1 \frac{(\arcsin(x))^2}{x^2} \frac{dx}{\sqrt{1-x^2}} = \pi \ln(2) \qquad (1)$$



This nasty-looking integral appears in the classic reference book, *Table of Integrals, Series, and Products,* by the Russian mathematicians I. S. Gradshteyn and I. M. Ryzhik. The book was first published in the 1940s and has been considered an essential reference by generations of working mathematicians and physicists. It is quite a large book; my copy is 1160 pages long.

In class I mentioned several quadrature methods beyond Gauss-Legendre. Equation (1) is of the form where the integral may be evaluated using Gauss-Chebyshev quadrature. That rule is

$$\int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}} dt \approx \sum_{i=1}^{N} w_i f(t_i)$$

where the weights are given by the rule

$$w_i = \frac{\pi}{N}$$

and the sample points are

$$t_i = \cos\left(\frac{2i-1}{2N}\pi\right) \qquad i = 1 \ldots N$$

Note that these are the sample points we first encountered when using Chebyshev polynomials for interpolation.

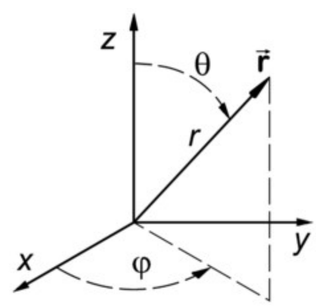
Please write a program to evaluate the above integral using Gauss-Chebyshev quadrature for different values of *N*. To test, please compare your numerical result against the mathematically true result. What is the minimum *N* required to achieve accuracy better than 1e-4?

Problem 2

In class we looked at integrating functions in two dimensions – i.e. surface integrals. The first method we discussed involved Gaussian quadrature in 2D – please use that approach for this problem. Consider integrating a function $f(r, \phi, \theta)$ over the surface of a unit sphere:

$$I = \int_0^{\pi} \int_{-\pi}^{\pi} \sin(\theta) d\phi \, d\theta \;\; f(r, \phi, \theta) \tag{2}$$

The coordinate system is shown at the right. Since we are working on the surface of the unit sphere, $r = 1$ so the integral (2) boils down to integration on the rectangular domain $\phi \in [-\pi, \pi], \; \theta \in [0, \pi]$. Therefore, it's easy to apply the two-dimensional Gaussian quadrature method learned in class.



For this problem, we'll integrate the function $f(\phi, \theta) = \cos^2(\phi)$. Please do the following:

1. Write a program to perform 7th order Gaussian quadrature to compute the surface integral (2) for $f(\phi, \theta) = \cos^2(\phi)$. A good place to get the nodes and the weights is this website: https://pomax.github.io/bezierinfo/legendre-gauss.html. Be careful: The integrand (2) is defined on the domain $\phi \in [-\pi, \pi], \; \theta \in [0, \pi]$, while Gauss quadrature requires a function defined on [-1, 1]. Therefore, you need to do a variable change.
2. Use your favorite method to compute the integral (2) analytically. (I used Wolfram Alpha.)
3. Test your program's result against the analytic result computed in step 2.

Problem 3

In class we looked at using triangular meshes as a way to integrate objects of arbitrary shape in 2 dimensions. In this problem you will integrate over a mesh to find the surface area of some 3 dimensional objects: First, the surface area of a unit sphere, and then the surface area of a cow.

I have placed STL files on Canvas for both the sphere and the cow. An STL file describes a 3D object as a collection of triangles and vertex points similar to the representation described in class. (As an aside, the STL file format was originally developed for CAD systems but is now finding widespread use describing geometry for 3D printing systems. If you have done any 3D printing it is likely you have dealt with an STL file.) Use the following Matlab functions to read in the files and create the triangle and vertex arrays:

```
S = stlread("sphere.stl")
pts = S.Points
tris = S.ConnectivityList
```

Please write a program which does the following:

- Read in an STL file describing a 3-dimensional geometric object using the above functions.
- Plot the object using the Matlab function trisurf(). This step is important to verify you correctly read in the object.
- Loop over all the triangles in the object and compute their area. To compute each triangle's area, compute the vectors describing two of the triangle's edges, $\vec{u}=\vec{p}_2-\vec{p}_1$ and $\vec{v}=\vec{p}_3-\vec{p}_1$, then compute the area from the (norm of the) cross product, $A=(1/2)\|\vec{u}\times\vec{v}\|$.
- Sum the individual triangle areas to get the total surface area of the input object.
- Print out the sum.

Regarding testing, the file "sphere.stl" describes a sphere of unit radius. If your program correctly computes the surface of the unit sphere (within 1% or less) then you can be confident your program is correct. Your test program should report both the surface area of the sphere as well as that of the cow.

With this problem I hope to convince you that although the math is not deep (at least in this case), the technique of representing geometric objects using triangular meshes is extremely powerful and useful.