Homework 3
Topic

Finn Joel Bjervig,[*] Thomas Herard,[†] and Anton Palm Ekspong[‡]

Computational Physics,
Department of Physics and Astronomy,
Uppsala university, Sweden

Februrary 2, 2021

[*]Electronic address: joelfbjervig@gmail.com
[†]Electronic address: herardthomas@gmail.com
[‡]Electronic address: anton.palm.ekspong@gmail.com

1 Introduction

There are many numerical ways to calculate an integral. Which method to use may wary depending on the problem. Two key things to consider when deciding on a method is accuracy and efficiency. Often there is a trade of between these two. Deciding on how fast it needs to be compared to how precises it need to be. Here we have compared the number of points need to achieve an accuracy and how long time it takes to calculated for Bode's rule, Simpson's rule and Gauss-Legendre quadrature rule.

2 Theory

Gauss Legendre quadrature will integrate exactly the definite a polynomial of degree $N-1$, $f(x)=x^p$ by expressing the integral as a linear combination of weights and function values $f(x_n)$ at optimally chosen abscissae points $x_n \in [-1,1]$.

$$\int_{-1}^{1} x^p dx \approx \sum_{n=1}^{N} \omega_n x_n^p \tag{1}$$

To choose the optimal x_n, consider the Legendre polynomials and take advantage of its orthogonality properties. It can be shown that the integral can be written as

$$I = \sum_{n=1}^{N} \omega_n R(x_n) \tag{2}$$

$R(x_n)$ is a polynomial of degree $N-1$ or less. ω_n need to be found such that R is integrated exactly, meaning we need to find x_n such that $P_N(x_n)=0$. The weight is related to the Legendre polynomial of degree N P_N as

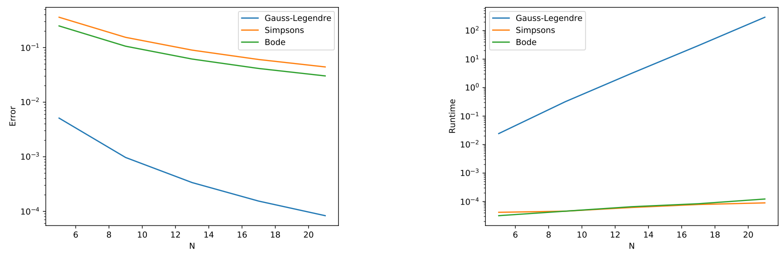
$$\omega_n = \frac{2(1-x_n^2)}{(N+1)^2[P_{N+1}(x_n)]^2} \tag{3}$$

The numerical implementation is thereby a matter of making a Legendre function generator of degree N, a general weight function, and finally implementing the Gauss Legenndre quadrature. The multiple root finder from previous assignments will come in handy since the roots of each Legendre polynomial x_n is required in the quadrature method. See the attached code in appendix for reference.

3 Discussion

The equation to be integrated in this assignment was

$$f(x) = \left(1-x^2\right)^{\frac{1}{2}} \tag{4}$$



(a) Accuracy of each method for increasing number of points N

(b) Run time for each method with increasing number of points

Figure 1

For the function given in this problem the Gauss Legendre quadrature achieves a higher accuracy for fewer number of points (see figure 1 a), but its run time is significantly longer than the other methods (see figure 1 b). The inevitable trade-off in computational science between performance and accuracy is very apparent here, since the Bode rule can achieve the same accuracy, not for fewer points N, but for a shorter run time, which serves as a better measure of performance.

Consider the Gauss Legendre quadrature with a Legendre polynomial of degree $N = 14$ (number of points is the same as the degree of the polynomial). The same accuracy of calculating the integral using Bodes rule is achieved with approximately $N = 489$ number of points (closer to 491, but that number doesn't satisfy Bode's condition of being a multiple of $4p + 1$) , see figure 2 Gauss Legendre quadrature takes 36.793123 seconds with 14 points, and produces an accuracy of $\delta = 0.000271$, while Bode's rule takes 0.001112 seconds with 489 points, and produces an accuracy of 0.000273.

Clearly this specific task favors Bodes rule, and perhaps even the Simpsons rule. For other functions, such as a monomial $f(x) = x^n$ perhaps the Gauss Legendre performs better.
The Legendre polynomial were found recursively. This could be one of the reasons that it was slower than Bode's rule. When finding the roots to the Legendre the bisection method will call the Legendre polynomials will call it self recursively twice for each order of polynomial. This combined with many iterations to find the roots could make it slower than the number of points used would suggest.

Obviously the time taken depends on the computer used so integration periods must be compared on the same machine. On another laptop in our group the Gauss-Legendre method takes above one second to run for N=12, equally, it must around the other of 500 for Bode to match the accuracy of the latter algorithm. We can draw the graph of Bode's runtime to have an idea of how the computing evolves to be able to reach the same accuracy as Gauss-Legendre.

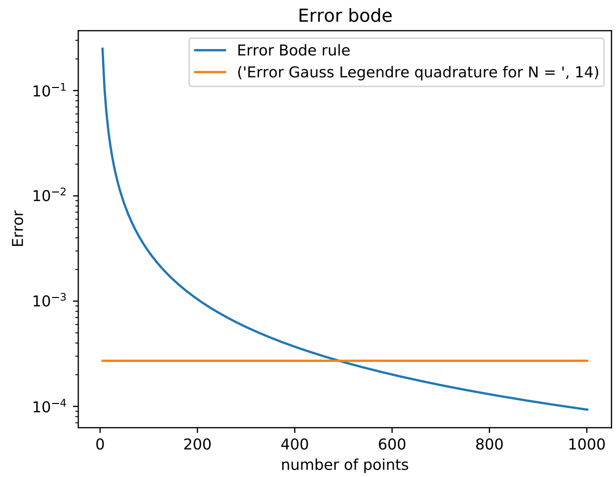


Figure 2: To determine for which number of points the bodes rule achieves the accuracy as the Gauss Legendre quadrature, we look at the following graph: Bodes approximation of the integral w.r.t the number of points for the method, compared to the error from Gauss Legendre quadrature for a fixed number of points $N = 14$.

We can see in Figure 3 that the evolution is linear and not exponential so there's no major issue with increasing N for Bode if we want to increase the accuracy. As we saw in Figure 2, the slope error with respect to the analytical value for large N is small compared to the one for lower values of N but still decent. This is Bode's strength: being able to increase the accuracy decently without making the computing time skyrocket.

Conclusion

This is all about making a compromise between accuracy and efficiency. Gauss-Legendre gives high accuracy immediately for a very small number of roots, however its computing time escalates exponentially. Depending on the problem, Gauss-Legendre is to be preferred when a high accuracy is needed but it can be rather judicious to prefer Bode when a large number of integrations have to be done so as not increase the efficiency of the program in question.

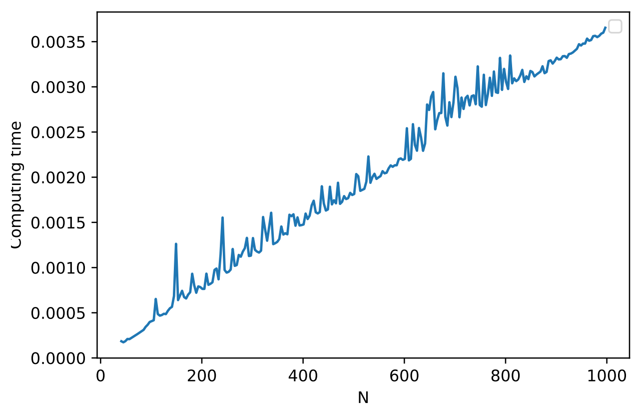


Figure 3: Time taken by Bode's algorithm for different values of N