Problem Set 6 - ECON 880

Spring 2022 - University of Kansas

Gunawan, Minh Cao

April 14, 2022

Question 1

We want to approximate $\int_0^1 x^{\frac{1}{3}}\, dx$ using (A) Trapezoid rule, (B) Gauss-Chebyshev quadrature, and (C) Gauss-Legendre quadrature. We use $n \in \{3, 5, 11\}$ nodes with each approach. The true value of the integral is 0.75. The formula for each method is given below:

- The Trapezoid rule: Let $h = (b-a)/n$, $x_i = a + ih$, and $f_j \equiv f(x_j)$, then

$$\int_a^b f(x)\, dx \doteq \frac{h}{2}\left[f_0 + 2f_1 + \cdots + 2f_{n-1} + f_n\right]$$

- The Gauss-Chebyshev quadrature:

$$\int_a^b f(y)\, dy \doteq \frac{\pi(b-a)}{2n} \sum_{i=1}^{n} f\left(\frac{(x_i+1)(b-a)}{2} + a\right)\left(1 - x_i^2\right)^{1/2}$$

  where x_i's are the Gauss-Chebyshev quadrature nodes over $[-1, 1]$:

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, \ldots, n.$$

- The Gauss-Legendre quadrature:

$$\int_a^b f(x)\, dx \doteq \frac{b-a}{2} \sum_{i=1}^{n} \omega_i f\left(\frac{(x_i+1)(b-a)}{2} + a\right),$$

  where the ω_i and x_i are the Gauss-Legendre quadrature weights and nodes over $[-1, 1]^\dagger$.

The relative error of the integral approximations is defined here as $|\hat{I} - I|/I$, where $\hat{I}$ and I is the true approximated and the value of the integral, respectively. See Table 1 for results.

	$n=3$	$n=5$	$n=11$
Trapezoid	0.081359	0.041771	0.01481
Gauss-Chebyshev	0.036964	0.012382	0.0024208
Gauss-Legendre	0.0051406	0.0015098	0.00020904

Table 1: Relative errors of the integral approximations

† See Kenneth L. Judd, 1998. "Numerical Methods in Economics," MIT Press Books, The MIT Press, p.260, Table 7.2

Comment: We see that for the three approaches, the more we add the number of nodes, the lower the relative errors become. Out of the three methods, Gauss-Legendre quadrature performs best with the lowest relative errors, followed by Gauss-Chebyshev, and finally the Trapezoid rule.

Question 2

We want to approximate $\int_{[0,1]^3} e^{x+2y+3z}\,dx\,dy\,dz$ using (A) pseudo-random numbers from Matlab's `rand()`, (B) uniformly spaced grid.

Method (A) relies on the fact that if $X, Y, Z \sim \mathcal{U}_{[0,1]}$, then

$$\mathbb{E}[f(X,Y,Z)] = \int_0^1 \int_0^1 \int_0^1 f(x,y,z)\,dx\,dy\,dz.$$

The crude Monte Carlo method generates n draws from $\mathcal{U}_{[0,1]}$, i.e. $\{x_i, y_i, z_i\}_{i=1}^n$, and takes

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} f(x_i, y_i, z_i)$$

as an estimate for $I = \int_0^1 \int_0^1 \int_0^1 f(x)\,dx\,dy\,dz$. The random draw from $\mathcal{U}_{[0,1]}$ is executed in Matlab using the command `rand()`[†]. The number of draws we choose for this method is of the form $100n$ with $n = 1, 2, \ldots, 30$.

Method (B) relies on the following definition of triple integral

$$\int_0^1 \int_0^1 \int_0^1 f(x,y,z)\,dx\,dy\,dz = \lim_{\Delta x, \Delta y, \Delta z \to 0} \sum_{i=1}^{n} f(x_i, y_i, z_i)\,\Delta x\,\Delta y\,\Delta z,$$

where the number of intervals we divide $[0,1]$ into, n, is chosen such that $\Delta x = \Delta y = \Delta z = 1/n$. In this exercise, we use n equally spaced nodes along each dimension with $n = 5, 6, 7, \ldots, 14$. Thus, the estimated integral for method (B) is given by

$$\hat{I} = \frac{1}{n^3}\sum_{i=1}^{n} f(x_i, y_i, z_i).$$

The true value of the integral is $I = 34.920804$. The relative error of the integral approximations is defined here as $|\hat{I} - I|/I$, where $\hat{I}$ and I is the true approximated and the value of the integral, respectively. See Figure 1 and 2 for relative errors resulting from method A and B, respectively.

Comment: On Figure 1, we see that Method (A) exhibits no clear pattern of diminishing relative error towards zero, however it is bounded by roughly 0.075. This result could potentially be due to the very large variance generated by crude Monte Carlo approximation despite its unbiasedness. On Figure 2, it is clear that uniformly spaced grid method yields relative error that consistently decreases towards zero as n increases. However, with the number of nodes less than 3,000 points, the pseudo-random method performs better than the uniformly spaced grid. Method (B) yields a relative error of roughly 0.8 at $n^3 = 14^3 = 2,744$ nodes, while method (A) yields a relatively lower relative error of 0.75 by using only $n = 1,100$ nodes. From implementation point of view, it is easier to use Matlab's pseudo-random number generator, since we only need to sample a smaller number of nodes to achieve the same level of error that the other method's generates at much higher number of nodes.

[†]We set here a seed number `rng(888)` for reproducibility.

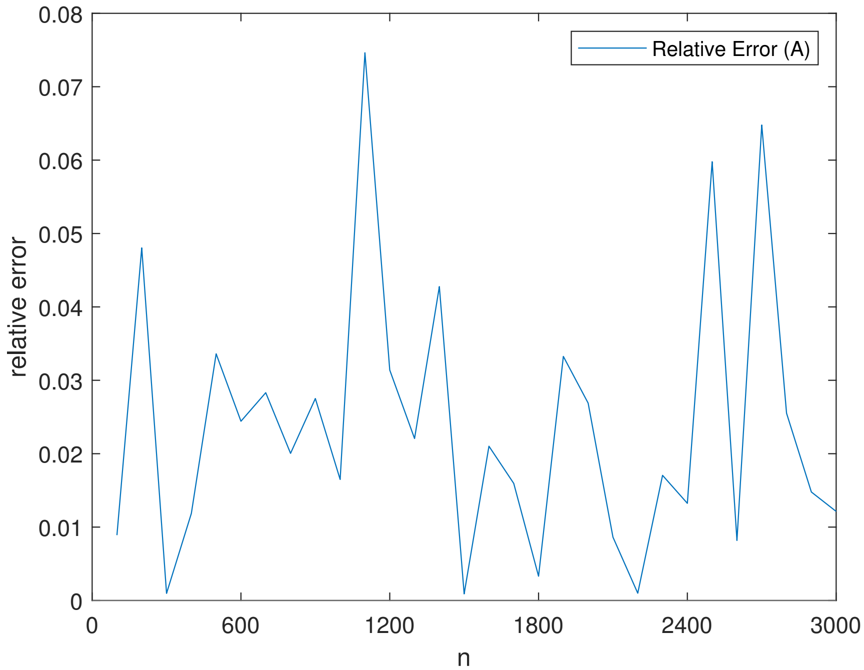

Figure 1: Relative Error for Method (A)

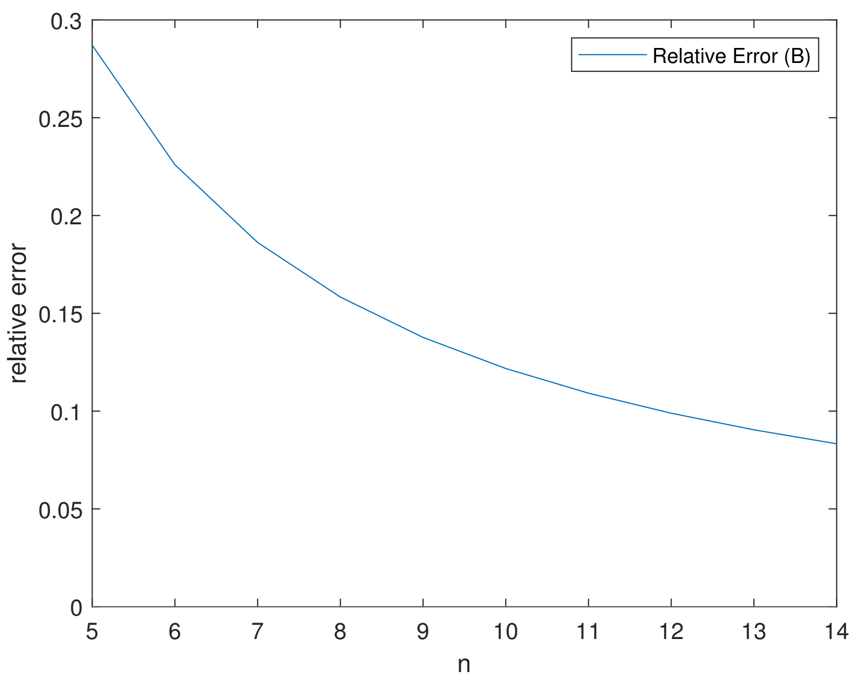

Figure 2: Relative Error for Method (B)