Numerical integration

Cecina Babich Morrow

2023-11-07

Integration

Many applications in statistics necessitate the calculation of integrals. Unfortunately many integrals cannot be computed in closed form, necessitating approximation techniques.

Quadrature

"Quadrature rules" refer to the process of approximating integrals, specifically in a deterministic way. Polynomial interpolation is at the heart of quadrature. To use quadrature, we'll start by numerically approximating the definite integral of a function over a finite interval using a polynomial function, since polynomials can be integrated exactly.

Gaussian quadrature

Gaussian quadrature attempts to approximate an integral as a weighted summation. There are many types of Gaussian quadrature, so this portfolio will focus on Gauss-Legendre quadrature. It aims to find an accurate approximation of the integral by selecting specific points (nodes) and associated weights for the integration. In the Gauss-Legendre quadrature, these nodes and weights are chosen in such a way that the method provides exact results for polynomials up to a certain degree. The nodes are typically roots of orthogonal polynomials, and the weights are determined to ensure optimal accuracy. The Gauss-Legendre quadrature is particularly effective for integrating functions over a finite interval.

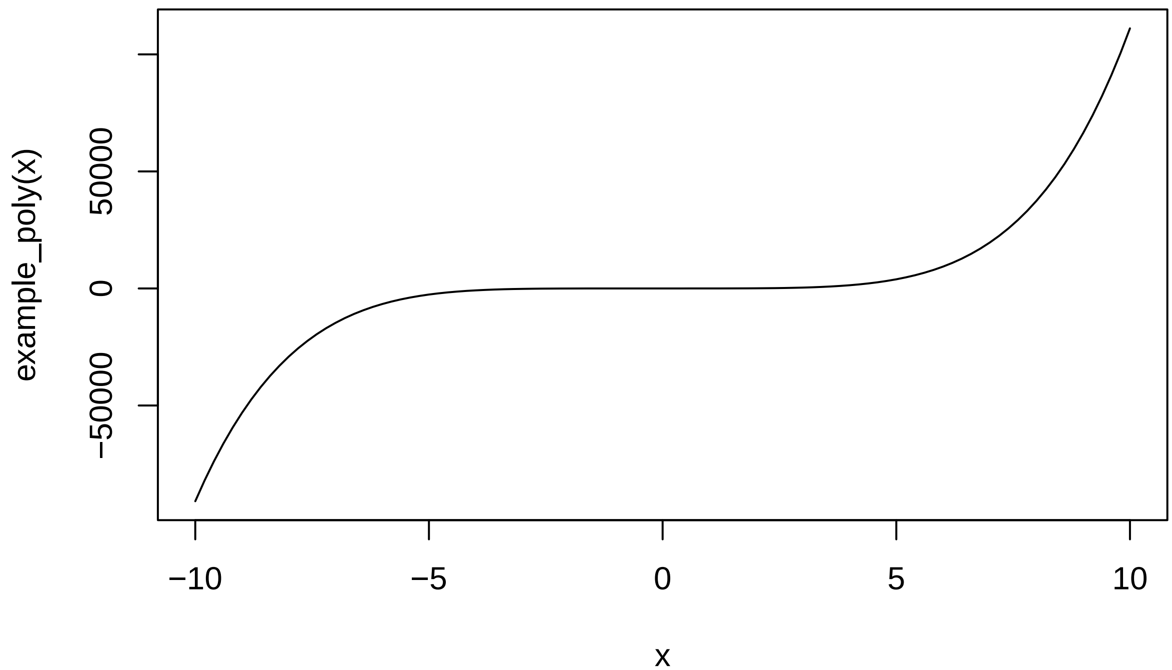
Gauss-Legendre quadrature takes the form

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

for a set of n sample points and w_i weights, where x_i are the roots of the nth Legendre polynomials. This rule can provide exact integration of degree $2n-1$ polynomials.

```r
library(statmod)
# Calculate nodes and weights where n = 3
gauss_3 <- gauss.quad(n = 3, kind = "legendre")
# where n = 8
gauss_8 <- gauss.quad(n = 8, kind = "legendre")
```

```r
example_poly <- function(x) 1 + x + x^2 + x^3 + x^4 + x^5
curve(example_poly, from = -10, to = 10)
```



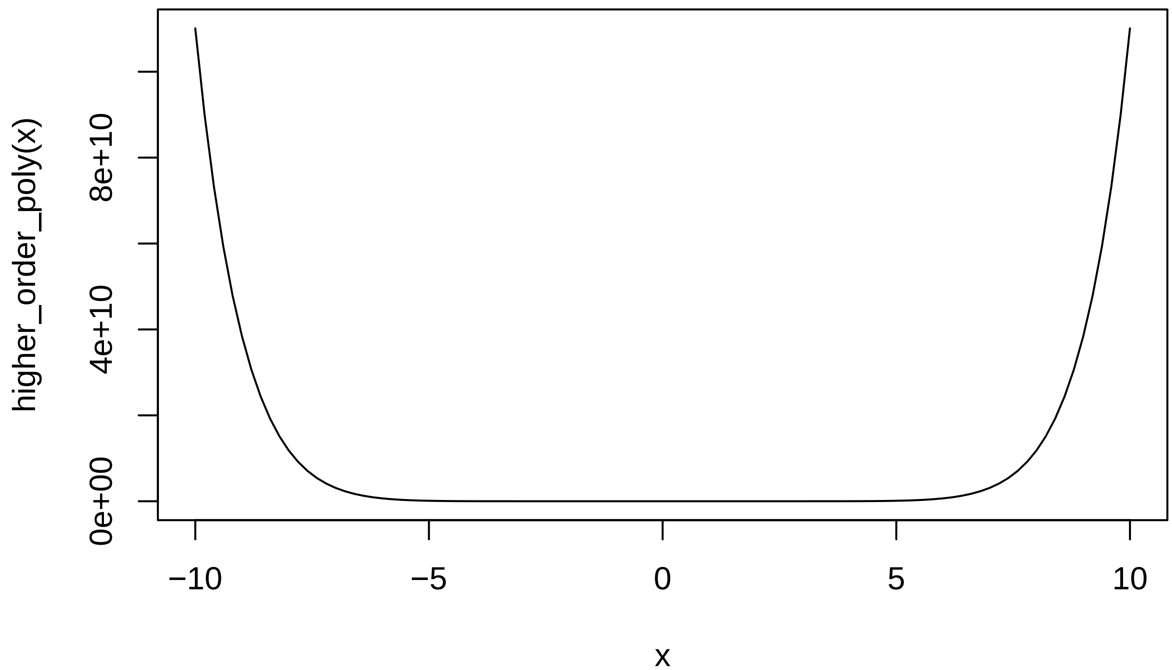

```r
gauss_legendre_approx <- function(f, gauss_quad_output) {
  weights <- gauss_quad_output$weights
  nodes <- gauss_quad_output$nodes
  approx <- sum(weights * f(nodes))
  return(approx)
}

gauss_legendre_approx(example_poly, gauss_3)
```

```
## [1] 3.066667
```

This is a very good approximation of the true integral (which is $\frac{46}{15} \approx 3.0666667$). Let's try a higher order polynomial:

```r
higher_order_poly <- function(x) 1 + x + x^2 + x^3 + x^4 + x^5 + x^8 + 11*x^10
curve(higher_order_poly, from = -10, to = 10)
```




```r
gauss_legendre_approx(higher_order_poly, gauss_3)
```

```
## [1] 4.161067
```

```r
gauss_legendre_approx(higher_order_poly, gauss_8)
```

```
## [1] 5.288889
```

As the number of nodes increases, we get a better approximation of the true integral ($\frac{238}{45} \approx 5.2889$).