

Measure theory

After calculus of variations, we need to divert further to a topic somewhat related to integration. Some pieces will be harder to follow otherwise.

A primer on measure theory

You may have grown accustomed to calling “integration” to something like $\int_a^b f dx$, but that’s actually just the Riemann integration. Ok, it “just” powers 99% of all real-life integration, but there are many other specialized integrals that can handle special situations like discontinuities or holes in f , such as the Riemann-Stieltjes integral, the Henstock-Kurzweil integral, and the Lebesgue integral. The last one is where we will concentrate the most.

We start with a seemingly unrelated problem: the size of a set. Let’s say we have a set of \mathbb{N} and we want to know how big it is: we may, for example, just count how many elements are there in the set:

$$\begin{aligned}\#\{\pi, e, 25\} &= 3 \\ \#\{\} &= 0 \\ \#\{1, 2, 3, \dots\} &= \infty\end{aligned}$$

In a sense, $\#$ is a function that receives a set with a combination of elements from \mathbb{N} and outputs a numeric value. In the literature, the elements belong to a set Ω and the combinations of those elements belong to a set of sets Σ , also called a sigma-algebra. More formally, $\# : \Sigma \rightarrow \mathbb{N}$.

$\#$ is called a measure. Measures are normally referred as μ , and fulfill roughly two conditions:

- $\mu(s) \geq 0, \forall s \in \Sigma$, that is, they are always positive. This condition can be relaxed, though, into having “signed measures”.
- If $A_1 \cap A_2 = \{\}$ $\implies \mu(A_1 \cup A_2) = \mu(A_1) + \mu(A_2)$. This means that the “size” of a set is the same as the sum of the sizes of its subsets.

There are many measures, with different properties, that are defined to operate over “measurable sets”, namely the pair (Ω, Σ) . You have already seen the counting measure. The Borel measure, which we will use interchangeably with the Lebesgue measure¹, can be defined as:

$$\lambda((a, b]) = b - a$$

If the set is a composition of intervals, it can be split into as many sums as we need, so:

$$\lambda([0, 2] \cup (12, 20)) = \lambda([0, 2]) + \lambda((12, 20)) = (2 - 0) + (20 - 12) = 10$$

It can also be reasonably extended to multidimensional cases:

$$\lambda^2((0, \pi] \times (2, 4]) = \lambda((0, \pi]) * \lambda((2, 4]) = (\pi - 0) * (4 - 2) = \pi * 2 = 2\pi$$

Connection between measures and integrals

We will now do something daring: have you noticed that definite integrals work over an interval (a set) and the result is a number? Yes, an integral is a (signed) measure². This will require a bit of rethinking of what integrating really means, as well as changing our notation for the integral.

In the world we all know, we do Riemann integration, $\int_a^b f dx$. This means:

- $I = [a, b]$ will select the portion of the function’s domain, Ω , to integrate. With the language we have used above, $I \subset \Omega$, $I \in \Sigma$
- We split I into partitions, typically using a single partition size Δx , and come up with a list of values $\{x_0 = a, x_1, \dots, x_n = b\} \in I$
- For each partition x_i we evaluate it with $f(x_i)$ to obtain a list of evaluations f_0, f_1, \dots, f_n
- We calculate an approximation of the integral by making each of these values a rectangle of height f_i and predefined width Δx , and summing over all rectangles.
- The integral, if it exists, is the limit of this sum when the partition’s norm tends to zero.

Here’s how it looks like:

¹The Lebesgue outer measure has a more complex definition that avoids some pathological results that the Borel measure has in certain measurable sets. In “reasonable” sets where the Borel measure is defined, the Lebesgue measure coincides.

²The converse, that any (signed) measure is always an integral, is also true under certain conditions. This is the consequence of the Radon-Nikodym theorem.

```

library(ggplot2)
library(broom)

f <- function(x) x^3 - 3*x^2 + x +4

# Define the interval [a, b] for integration
a <- 0
b <- 3

# Number of rectangles for approximation
n <- 24

# Generate x values and corresponding function values
x_values <- seq(a, b, length.out = n+1)
y_values <- f(x_values)

# Create a data frame for ggplot
df <- data.frame(x = x_values, y = y_values)

# Calculate rectangle heights
df$rect_heights <- c(0,f(x_values)[2:length(x_values)])

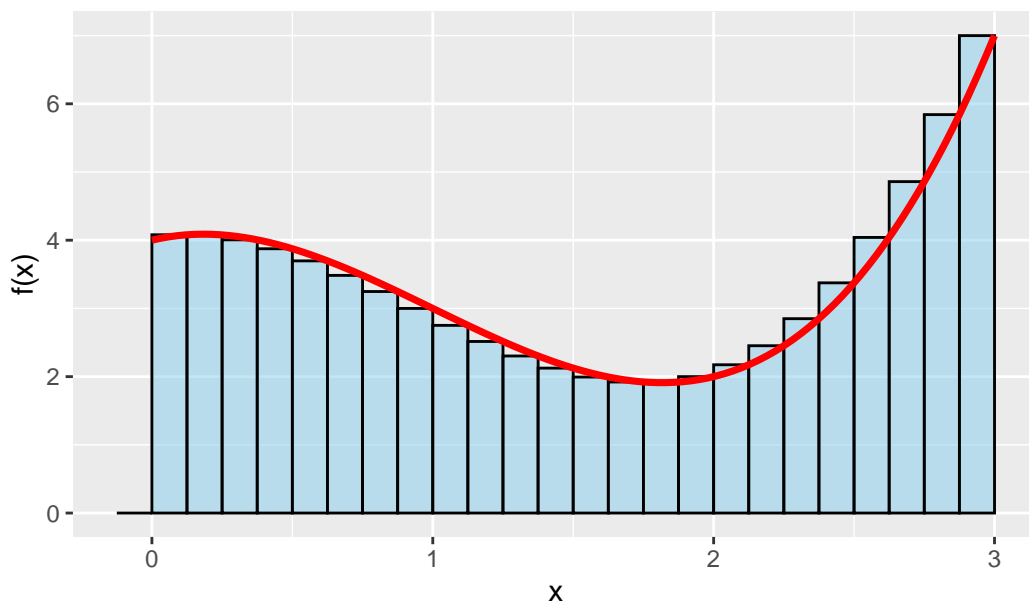
# Calculate total width of the bars to fill the x-axis
total_width <- b - a

# Create a ggplot with rectangles
p <- ggplot(df, aes(x = x, y = rect_heights)) +
  geom_col(width=(total_width/n), fill = "skyblue", color = "black", just = 1, alpha=0.5) +
  geom_function(fun = f, color = "red", linewidth = 1.25) +
  labs(title = "Riemann Integration Demonstration",
       x = "x", y = "f(x)")

# Print the plot
print(p)

```

Riemann Integration Demonstration



Powerful as it is, this can be improved. For starters, we can only integrate intervals over a real domain, $\Omega = \mathbb{R}$. For functions with many variables, we also need to integrate over many real domains, for example $\int_{x_a}^{x_b} \int_{y_a}^{y_b} \int_{z_a}^{z_b} f(x, y, z) dz dy dx$.

Notice that we could get rid of these restrictions if we integrate over the only real line that's guaranteed to exist: the one of the codomain or image. Kind of like a stack of dishes, for each height we measure the domain that falls under it. This is called Lebesgue integration, it's written as $\int_s f d\lambda$, and it means:

- s will select a portion of the function's domain, Ω , to integrate. With the language we have used above, $s \subset \Omega$, $s \in \Sigma$
- We split the image of the function, f^{-1} into partitions, and we obtain the images f_0, f_1, \dots, ∞
- For each of these partitions, we obtain the set of values of the domain that have a function value at least that height: $f_t^{-1} = \{\omega \in s / f(\omega) > t\} \in \Sigma$. Notice that these sets should be measurable and that $f_\infty = \{\}$, so that the integral is bounded.
- We calculate an approximation of the integral by making each of these sets a group of "volumes" of predefined height f_t and base $\mu(f_t^{-1})$, and summing over all rectangles.
- Adding up all the measured sets, all the slices of heights, will yield a final value, which behaves as a measure

- The integral, if it exists, is the supremum of all the sum calculations that can be achieved by splitting the image into smaller and smaller partitions

Here's how this mess looks like:

```

A <- 1
B <- -3
C <- 1
D <- 4
f <- function(x) A*x^3 + B*x^2 + C*x + D
dfdx <- function(x) A*3*x^2 + B*2*x + C

# Define the domain [left, right] for integration
left <- 0
right <- 3

# Number of image partitions
max_y <- 7
n <- 28
delta_y <- max_y / n

# Generate x values and corresponding function values
y_values <- seq(0, max_y, length.out = n + 1)

find_real_roots_of_cubic <- function(A, B, C, D, tolerance = 1e-10) {
  # Find the roots using polyroot()
  roots <- polyroot(c(D, C, B, A))

  # Extract real parts of the roots
  real_parts <- Re(roots)

  # Filter out complex roots (where the imaginary part is close to zero)
  real_roots <- real_parts[abs(Im(roots)) < tolerance]

  return(real_roots)
}

calculate_intervals_for_measure <- function(A, B, C, D, lower_bound, upper_bound) {
  # Find the real roots
  roots <- find_real_roots_of_cubic(A, B, C, D)
  roots <- sort(roots)

```

```

# The first interval should start on the lower bound if the function is decreasing
intervals <- data.frame(a = double(), b = double())
current_left <- lower_bound

# Then, build every pair of (f' > 0, f' < 0).
for (i in (1:length(roots))) {
  # Stop if you have gone beyond the boundary.
  if (roots[i] > upper_bound) {
    break;
  }

  if (!is.na(current_left)) {
    possible_right <- roots[i]
    if (dfdx(possible_right) < 0) {
      if (current_left < possible_right) {
        # Add an interval.
        intervals[nrow(intervals) + 1,] <- c(current_left, possible_right)
        current_left <- NA
      } else {
        # This is closing an interval to the left of the lower bound. Skip.
      }
    } else {
      # Another increasing left would be the start of an interval to the left of the low
      # or to the right of the lower bound. So, just keep the maximum.
      # It could also be a missed root in between, but we will disregard this option.
      current_left <- max(current_left, possible_right)
    }
  } else {
    possible_left <- roots[i]
    if (dfdx(possible_left) > 0) {
      if (possible_left < upper_bound) {
        # The opening of a new interval. As long as it's not beyond the upper bound, use
        current_left <- possible_left
      }
    } else {
      # A decreasing function, without a left boundary, can be discarded
    }
  }
}

# If there's an open interval after processing, close it at the upper bound.

```

```

if (!is.na(current_left)) {
  intervals[nrow(intervals) + 1,] <- c(current_left, upper_bound)
}

return(intervals)
}

# Create a data frame for ggplot
rectangles <- data.frame(
  xmin = double(),
  xmax = double(),
  ymin = double(),
  ymax = double())

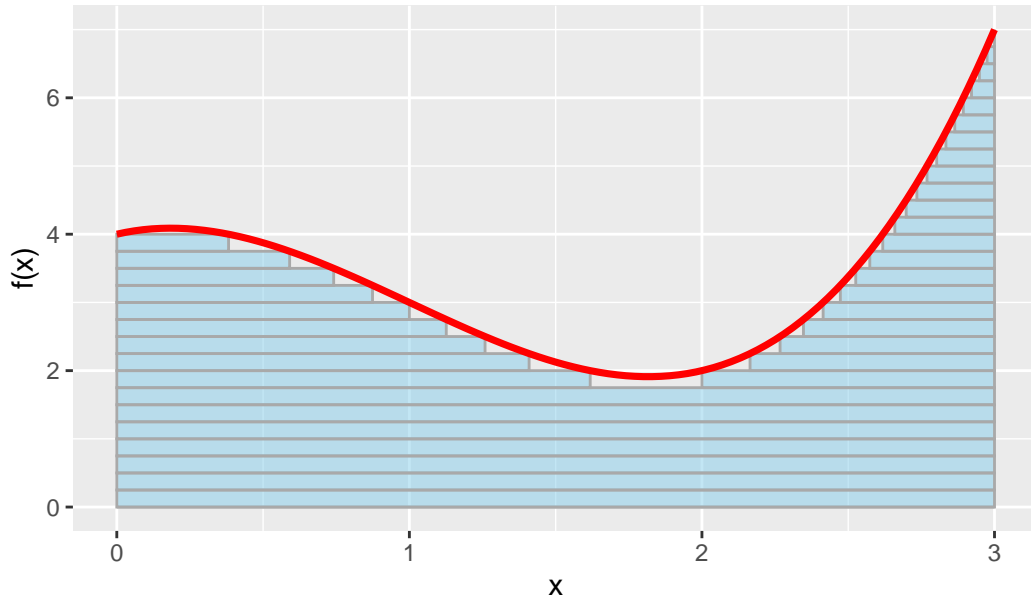
# Calculate all rectangles
for(y in y_values) {
  intervals <- calculate_intervals_for_measure(A, B, C, D - y, left, right)
  for (i in (1:nrow(intervals))) {
    interval <- intervals[i,]
    rectangles[nrow(rectangles) + 1,] <- c(interval[1], interval[2], max(0, y - delta_y),
  }
}

# Create a ggplot with rectangles
p <- ggplot() +
  geom_rect(data = rectangles,
    aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax),
    fill = "skyblue",
    colour = "darkgrey",
    alpha = 0.5) +
  geom_function(fun = f, color = "red", linewidth = 1.25) +
  labs(title = "Lebesgue Integration Demonstration",
    x = "x", y = "f(x)")

# Print the plot
print(p)

```

Lebesgue Integration Demonstration



Gaussian measures

As a final example of a measure, the n -dimensional Gaussian measure, with mean 0 and standard deviation 1, is defined as:

$$\gamma^n(A) = \frac{1}{(2\pi)^{n/2}} \int_A \exp\left(-\frac{1}{2}\|x\|_{\mathbb{R}^n}^2\right) d\lambda^n(x)$$

Again, for every possible value of $x \in A$, we will measure all the image sets that result from those values and we add up all those measure results. This particular function is really great because for $x \rightarrow \pm\infty$, the measure is still a number. It's also called a probability measure because $\gamma^n(\Omega) = 1$, that is, the total measured area under the whole domain is 1 or 100%. If you have done a bit of probability, the function \mathbb{P} was a probability measure all along.

This is a lot to digest, I know. Just read it until it clicks. It took me 3 years of meditating on the subject after all!