

# Métodos Numéricos y Optimización

## Práctica 1

Mariano Villafuerte González

Una función implícita es una función de la forma:

$$F(x, y) = 0$$

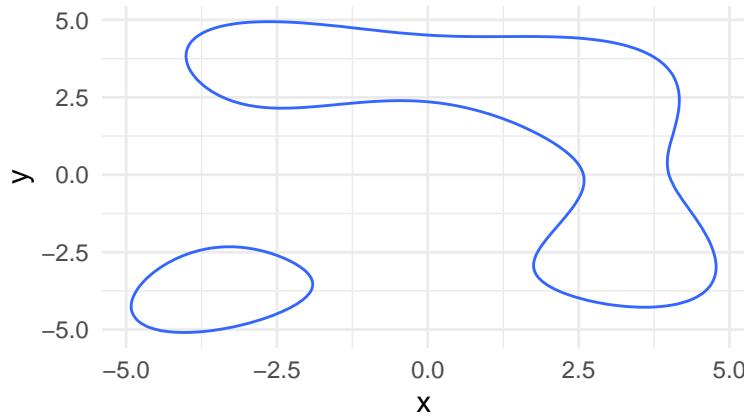
Las funciones implícitas se pueden utilizar para modelar objetos geométricos. Así, un objeto geométrico puede estar definido en términos de  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  como:

$$G = \{(x, y) \in \mathbb{R}^2 \mid F(x, y) \leq 0\} \subset \mathbb{R}^2$$

Para este caso, se considera la región  $G$  determinada por la función:

$$F(x, y) = (x^2 + y - 10)^2 + (x + y^2 - 12)^2 - 100$$

El primer paso consiste en graficar la función implícita. Al hacerlo, se puede observar que el área contenida por la función implícita no es una figura geométrica de la que se conozca su fórmula para estimar el valor del área.



## Cálculo del Área $G$

Para calcular el área de  $G$ , podemos usar el método de Monte Carlo. Para esto, se genera una muestra aleatoria de  $x_i$ . Despues, se evalúa la proporción de los puntos que se encuentran dentro del área deseada y se multiplica por un área conocida delimitada por un rango (dentro del cual se distribuyen las  $x_i$  aleatorias)

Por ejemplo, definimos el área conocida por el cuadrado formado por  $(x, y) \in \{-6, 6\}$  y se procede a calcular  $N$  puntos aleatorios suponiendo que se distribuyen  $U(-6, 6)$ . El área del cuadrado  $A = bh = 144$ .

- *Ejemplo con  $N = 10^2$*

Se procede a generar 100 puntos aleatorios  $(x, y)$ . Una vez hecho eso, se calcula cuántos de estos cumplen la condición:

$$F(x, y) \leq 0$$

En este caso, se obtuvieron 25 puntos que cumplen esta condición, por lo que se estima que área de  $G$  es:

$$A(G) = A * \frac{n}{N}$$

Donde  $A$  es el área total seleccionada donde viven los  $x_i$ ,  $n$  son la cantidad de puntos generados que están dentro de  $G$  y  $N$  es el número total de  $x_i$  generados. Por lo que, para este caso, el cálculo queda de la siguiente manera:

$$A(G) = 144 * \frac{25}{100}$$

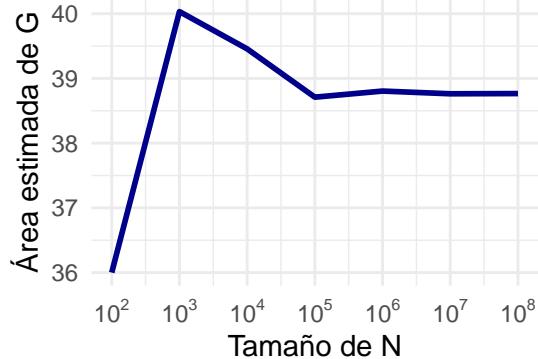
$$A(G) = 36$$

Ahora bien, el cuadrado que fue seleccionado aquí se denomina el *Bounding Box*, sin embargo, no se realizaron esfuerzos por encontrar el *Best Bounding Box*. Ese tema va más allá de esta práctica, sin embargo, debe aclararse que la selección de este Bounding Box se hizo de manera visual de tal modo que toda el área  $G$  estuviera contenida

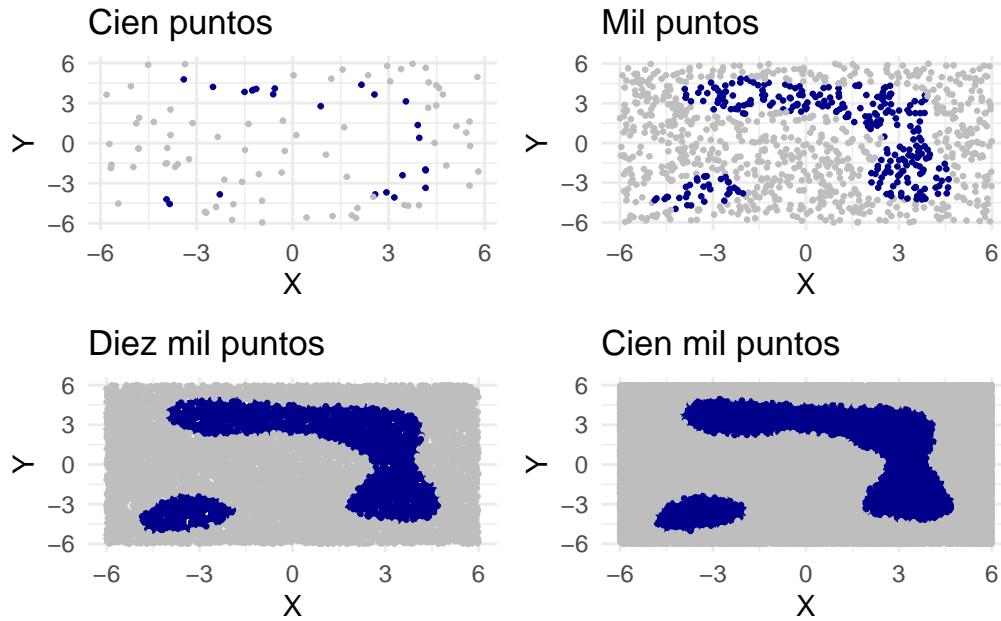
## Cálculo para distintos valores de $N$

Este cálculo se puede realizar para distintos valores de  $N$ . El hacer  $N$  más grande debería entregar valores más cercanos al valor verdadero. Para más información se puede consultar el libro de [Optimización](#) de Erick Palacios.

Por ejemplo, si se calcula con  $N = 10^2, 10^3, \dots, 10^8$ , se obtienen los siguientes resultados:



num_sample	result
100	36
1000	40.032
10000	39.456
$1e+05$	38.71008
$1e+06$	38.804544
$1e+07$	38.7623808
$1e+08$	38.76562944



Como se puede apreciar, el área calculada puede fluctuar más conforme se usan menos puntos para su cálculo, pero conforme se aumenta el la  $N$  se observan valores más estables. Además, se muestra visualmente cómo Monte Carlo aproxima el valor del área con este muestreo aleatorio de puntos dentro de un conjunto.

## Área de $G$ con Muestreo Adaptativo

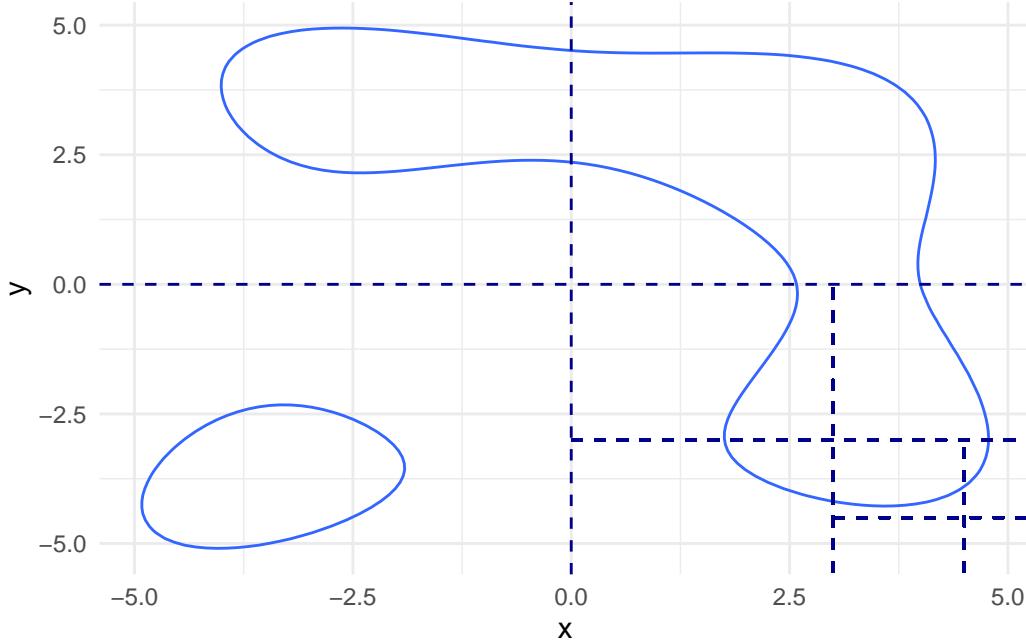
Otra idea que se puede desarrollar es la de aproximar el área de  $G$  con cuadrados. Es decir, se pueden trazar  $n$  cuadrados dentro de  $G$  y se considera que el área es igual a la de los cuadrados. Para hacer esto, se empieza con un cuadrado que contenga todo, en este caso usaremos el cuadrado formado por

$$(x, y) \in \{-6, 6\}$$

El siguiente paso es evaluar si el cuadrado se encuentra contenido en  $G$ . Esto lo hacemos evaluando las esquinas y si  $F(x, y) \leq 0$ , entonces el cuadrado se encuentra contenido en  $G$  y se puede sumar al estimado del área total. En otro caso, si el cuadrado no se encuentra contenido por  $G$  partimos el cuadrado en 4, partiendo por mitades el cuadrado. En este caso serían los cuadrados con vértices en

$$(x, y) \in \{-6, -3, 3, 6\}$$

Se pueden ejemplificar algunos cortes. Nos enfocamos en el cuadrante inferior derecho:



Se puede notar que el último corte produce un cuadrado (en la esquina inferior derecha) en el que ya no se itera, ya que este cuadrado ya no contiene parte del área  $G$ .

Para el cálculo del área con esta idea, se utiliza la siguiente función:

```
rec_int <- function(f,
                      x_range = c(-5, 5),
                      y_range = c(-5, 5),
                      k = 1,
                      kmin = 5,
                      kmax = 20) {

  # Extraemos valores del rango
  x1 <- x_range[1]
  x2 <- x_range[2]
  y1 <- y_range[1]
  y2 <- y_range[2]

  # Calculamos la función implícita en las esquinas
  e_1 <- f(x1, y1) <= 0
  e_2 <- f(x1, y2) <= 0
  e_3 <- f(x2, y1) <= 0
  e_4 <- f(x2, y2) <= 0

  # Checamos si aún se permiten iteraciones
  if (k < kmax) {

    # Checamos si todas las esquinas están contenidas en G
    if (e_1 && e_2 && e_3 && e_4) {

      # Si está en G, calculamos su área
      return ((x2 - x1) * (y2 - y1))
    } else {

      # Checamos si alguno de los puntos está en G
      # 0 si no hemos hecho las iteraciones mínimas
      if ((e_1 || e_2 || e_3 || e_4) || k <= kmin) {

        # Calculamos los nuevos vértices
        half_x <- (x2 + x1) / 2
        half_y <- (y2 + y1) / 2

        # Llamamos a la función con los nuevos vértices
        return(
          # Primer cuadrado
```

```

        rec_int(f,
                  c(x1, half_x),
                  c(y1, half_y),
                  k + 1, kmin, kmax) +
        # Segundo cuadrado
        rec_int(f,
                  c(half_x, x2),
                  c(y1, half_y),
                  k + 1, kmin, kmax) +
        # Tercer cuadrado
        rec_int(f,
                  c(x1, half_x),
                  c(half_y, y2),
                  k + 1, kmin, kmax) +
        # Cuarto cuadrado
        rec_int(f,
                  c(half_x, x2),
                  c(half_y, y2),
                  k + 1, kmin, kmax))

    } else {
        # Si ninguno de sus puntos está en G,
        # la aportación del cuadrado es 0
        return(0)
    }
}
} else {
    # si llegamos a kmax, regresamos el área de los
    #cuadrados que tienen cuando menos un punto en G
    if (e_1 || e_2 || e_3 || e_4) {
        return((x2 - x1) * (y2 - y1))
    } else {
        return(0)
    }
}
}
}

}

```

Lo cual, produce un resultado igual a 38.7048

Cabe mencionar que en esta función, se incluyen los cuadrados con al menos un vértice contenido dentro de  $G$  cuando se llega a el máximo de iteraciones, sin embargo, se puede omitir ese paso y únicamente considerar los cuadrados completamente contenidos en  $G$ , en ambos casos, a mayor número de cuadrados, la estimación del área de  $G$  será más precisa.

## Probabilidad de pertenecer a $G$

Otro tema de interés puede ser el estimar cuál es la probabilidad de que un punto aleatorio  $(x, y)$  esté en la región  $G$ . Es decir, se desea calcular:

$$\mathbb{P}[(x, y) \in G]$$

En específico, consideremos el caso en que

$$(x, y) \in \mathbb{R}^2$$

se distribuyen como normal bivariada con

$$\mu_x = \mu_y \in \{-5, -4, \dots, 4, 5\}$$

$$\sigma_x = \sigma_y = 2$$

$$\rho_{xy} = 0$$

Para este caso, la integral para calcular  $\mathbb{P}[(x, y) \in G]$  esta dada por:

$$P = \iint_G \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)} \left[ \frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2} \right]} dx dy$$

$$P = \iint_G \frac{1}{8\pi} e^{-\frac{1}{2} \left[ \frac{(x-\mu_x)^2}{4} + \frac{(y-\mu_y)^2}{4} \right]} dx dy$$

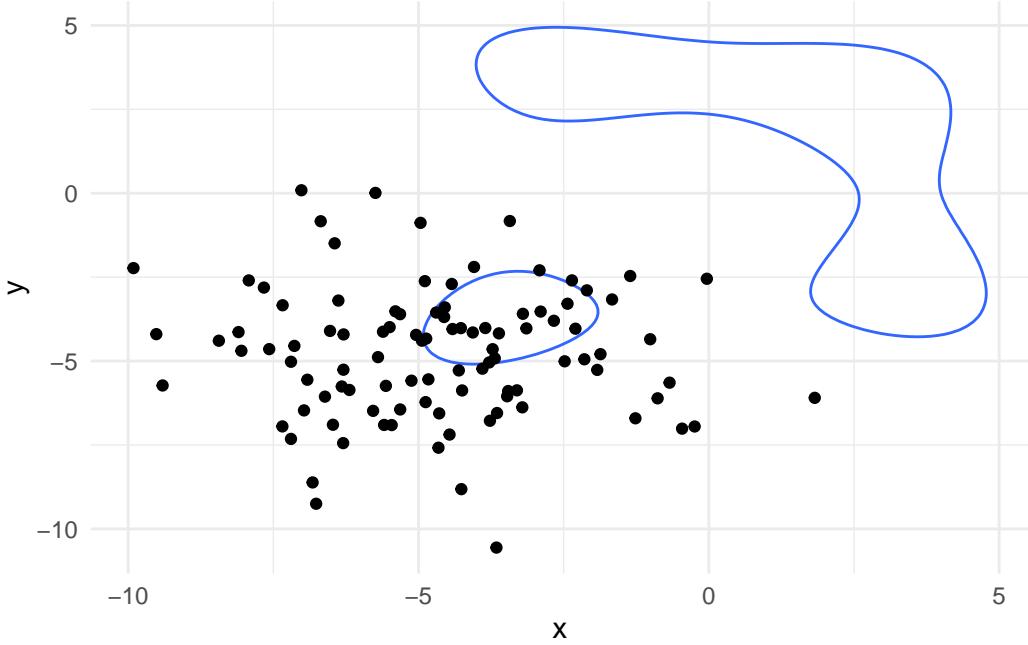
pero de igual manera, se puede utilizar Monte-Carlo. La integral puede parecer muy complicada, sin embargo, Monte Carlo puede facilitar este cálculo.

Lo primero que haremos es hacer muestras aleatorias de  $(x, y)$ , y procedemos a calcular la proporción de los puntos que pertenecen a la región.

En este caso, no buscamos calcular un área, por eso se omite el paso de multiplicar la proporción por el área conocida del *Bounding Box*.

Ahora bien, se puede observar que  $\mu_x = \mu_y$  pueden tomar valores, distintos, por lo que se procede a hacer las distintas combinaciones posibles y se hace el cálculo que ya se ha mencionado.

Por ejemplo, si tenemos  $\mu_x = \mu_y = -5$ , se puede observar una muestra de cien puntos. En este ejemplo, alrededor de 18 puntos caen dentro de  $G$  por lo que diríamos que la probabilidad de que  $(x, y)$  estén en  $G$  es aproximadamente 18%. Ahora bien, se repite esta idea para todas las combinaciones de  $\mu_x = \mu_y$  y con muestras más grandes.



Con esta misma idea, se calcula la probabilidad para todos los valores de  $\mu_x = \mu_y$  y con distintos  $n$  puntos aleatorios.

$\mu_x = \mu_y$	Número de puntos						
	$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$	$10^8$
-5	0.09	0.154	0.1450	0.14190	0.14128	0.14131	0.14134
-4	0.16	0.241	0.2064	0.20735	0.20838	0.20853	0.20846
-3	0.30	0.208	0.2050	0.20705	0.20588	0.20618	0.20595
-2	0.23	0.160	0.1501	0.15517	0.15451	0.15440	0.15442
-1	0.16	0.146	0.1453	0.14503	0.14492	0.14506	0.14495
0	0.31	0.244	0.2376	0.23608	0.23531	0.23482	0.23509
1	0.33	0.413	0.3979	0.38707	0.38700	0.38652	0.38649
2	0.45	0.471	0.4794	0.48116	0.48043	0.48042	0.48037
3	0.37	0.409	0.4178	0.41983	0.42305	0.42312	0.42295
4	0.29	0.254	0.2571	0.25815	0.25794	0.25749	0.25758
5	0.10	0.131	0.1081	0.10597	0.10625	0.10610	0.10635

Y se puede hacer una gráfica de la probabilidad para cada valor de  $\mu_x = \mu_y$ . Se puede apreciar que el valor se estabiliza conforme se utiliza un mayor número de puntos.

