

# Trabalho 1

Felipe Pêpe, Gustavo Oliveira, Igor Godinho, Lucas Inocência, Matheus Cardoso

## Implementação do Solver

Aqui, primeiramente projetamos a matriz para resolver o problema:

$$\frac{\partial}{\partial x} \left( k \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial u}{\partial y} \right) = f(x, y), \forall \quad x, y \text{ em } \Omega : (0, 1) \times (0, 1)$$

condições de contorno:  $u(x, y) = 0, \forall \quad x, y \text{ em } \partial\Omega$

```
L <- 1
n <- 25

d <- L / (n - 1)

# Gerando D (Matriz que faz d2u)
solver <- matrix(0, n*n, n*n)

for (i in 1:(n * n)) {
  if (i <= n || i >= (n*n - n) || i %% n <= 1) {
    solver[i, i] <- 1
    next
  }

  solver[i, i - n] <- 1
  solver[i, i - 1] <- 1
  solver[i, i] <- -4
  solver[i, i + 1] <- 1
  solver[i, i + n] <- 1
}

A <- solve(solver)
```

Perceba que o R já possui uma função para calcular a inversa de uma matriz, a função *solve*.  
A partir deste momento, já temos a matriz A.

### Projetando a função $f$ .

Nesse caso, escolhemos que  $f$  seja a função:

$$f(x, y) = -\frac{e^x}{|y| + 1}$$

```
f <- matrix(0, n * n, 1)
X = seq(0, L, d)
for (i in 1:length(X)) {
  for (j in 1:length(X)) {
    f[i + (j - 1) * n] <- - (exp(X[i]) / (abs(X[j]) + 1))
  }
}

solution <- A %*% f
```

Ao final, obtemos a matriz solução.

1. Temos a matriz A.
2. Temos o vetor força  $f$ .

Fazemos:

$$U = Af$$

Em que  $U$  é o vetor solução.

### Aplicando as Condições de Contorno

Apenas definimos a matriz cujos valores serão o do espaço (em  $z$ ). Caso o ponto esteja nas bordas, o valor é zero.

```

x <- seq(0, L, d)
y <- seq(0, L, d)

z <- matrix(0, n, n)
for (i in 1:n) {
  for (j in 1:n) {
    if (i == 1 || j == 1) {
      z[i, j] <- 0
    } else {
      z[i, j] <- solution[i + (j - 1) * n]
    }
  }
}
}

```

## Plotagem do Gráfico

Finalmente, plotamos o gráfico.

```

fig <- plot_ly(
  type = 'surface',
  contours = list(
    x = list(show = TRUE, start = 1.5, end = 2, size = 0.04, color = 'white'),
    z = list(show = TRUE, start = 0.5, end = 0.8, size = 0.05)),
  x = ~x,
  y = ~y,
  z = ~z)

fig <- fig %>% layout(
  scene = list(
    xaxis = list(nticks = 20),
    zaxis = list(nticks = 4),
    camera = list(eye = list(x = 0, y = -1, z = 1)),
    aspectratio = list(x = .9, y = .8, z = .8)))

htmlwidgets::saveWidget(widget = fig, file = "hc.html")
webshot(url = "hc.html", file = "hc.png", delay = 1, zoom = 4, vheight = 500)

```

