Zadanie Obliczeniowe

Piotr Karamon

1 Problem

$$-k(x)\frac{d^2u(x)}{dx^2} = 0$$

$$u(2) = 3$$

$$\frac{d(u)}{dx} + u(0) = 20$$

$$k(x) = \begin{cases} 1 & \text{dla } 0 \leq x \leq 1 \\ 2 & \text{dla } 1 < x \leq 2 \end{cases}$$

2 Sformułowanie wariacyjne

$$\forall x \in \Omega \quad k(x) \neq 0$$

$$\begin{aligned}
-ku'' &= 0 \\
u'' &= 0 \qquad \cdot v \quad v(2) = 0 \\
u''v &= 0 \\
\int_0^2 u''v\,dx &= 0 \\
u'v\Big|_0^2 - \int_0^2 u'v'\,dx &= 0 \\
u'(2)v(2) - u'(0)v(0) - \int_0^2 u'v'\,dx &= 0 \\
-(20 - u(0))v(0) - \int_0^2 u'v'\,dx &= 0 \\
u(0)v(0) - \int_0^2 u'v'\,dx &= 20v(0) \\
B(u, v) &= L(v)
\end{aligned}$$

$$u(2) = 3$$

Czyli musimy wykonać przesunięcie warunku Dirichleta.

$$u = w + 3, \quad w(2) = 0$$

Korzystając z biliniowości B możemy zapisać:

$$
\begin{aligned}
B(w+3,v) &= L(v)\\
B(w,v)+B(3,v) &= L(v)\\
B(w,v) &= L(v)-B(3,v)\\
-\int_0^2 w'v'\,dx + w(0)v(0) &= 20v(0)+\int_0^2 0\cdot v'\,dx - 3v(0)\\
-\int_0^2 w'v'\,dx + w(0)v(0) &= 17v(0)\\
B(w,v) &= L(v)
\end{aligned}
$$

3 Rozwiązanie

Najpierw dzielimy na Ω wyróżniamy n równo odległych od siebie puntków.

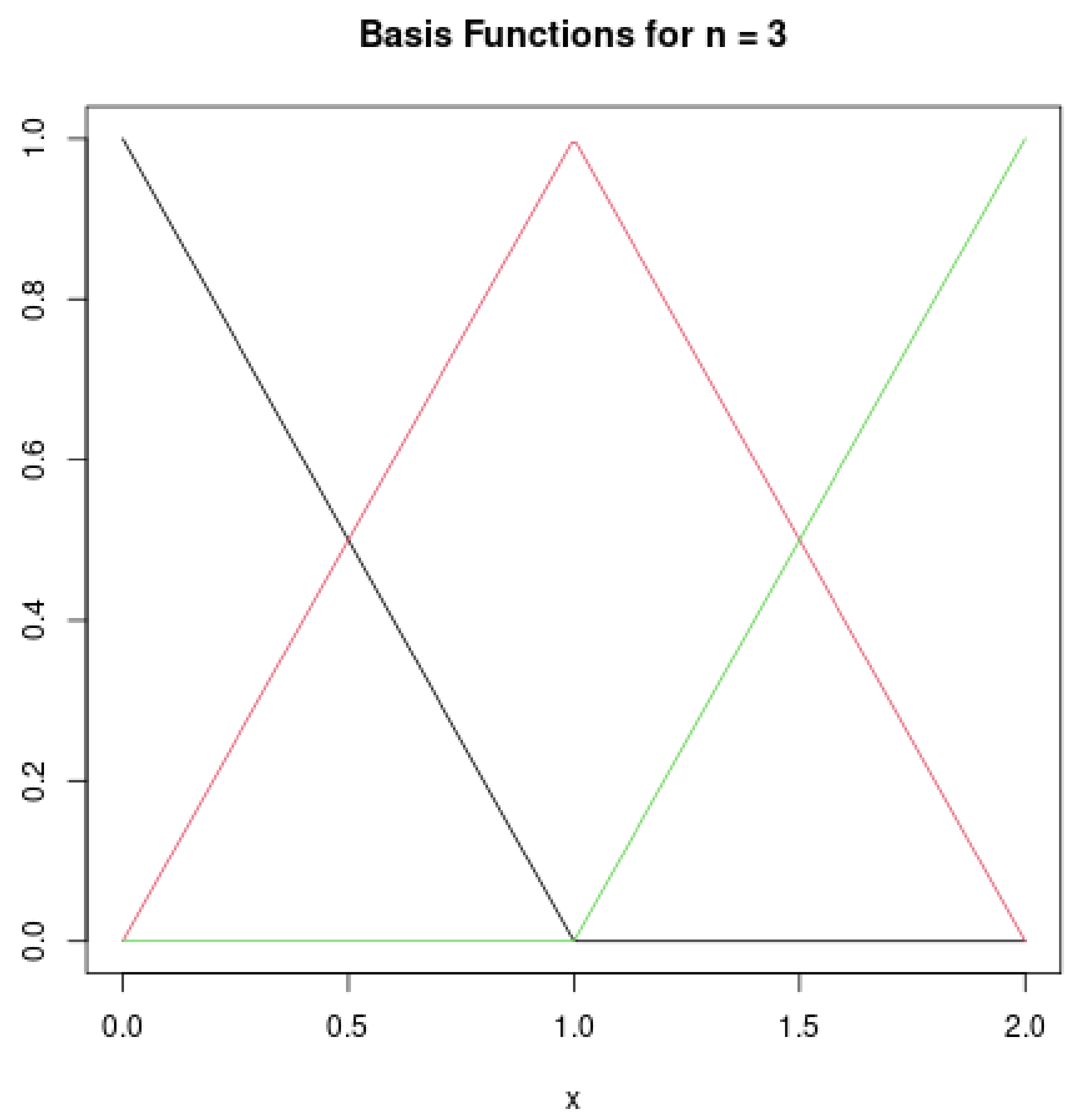
```
a <- 0
b <- 2
n <- 3
h <- 2 / (n - 1)

x_i <- function(i) {
  return((i - 1) * h)
}
```

Kolejno tworzymy funkcje bazowe przestrzeni $V_h \subset V$. Które kształtem przypominają “daszek”.

```
e <- function(i, x) {
  if (x > x_i(i-1) && x < x_i(i)){
    return (1/h*(x - x_i(i-1)))
  }
  else if (x >= x_i(i) && x < x_i(i+1)){
    return (-1/h*(x - x_i(i+1)))
  }
  else {
    return (0)
  }
}

deriv_e <- function(i, x){
  if (x > x_i(i-1) && x < x_i(i)){
    return (1/h)
  }
  else if (x >= x_i(i) && x < x_i(i+1)){
    return (-1/h)
  }
  else {
    return (0)
  }
}
```



W celu obliczenia całek użyjemy kwadratury Gaussa, używając dwóch puntków. Kwadratura normalnie działa na przedziale $[-1, 1]$ my będziemy obliczać całki po przedziałach mieszczących się w $(0, 2)$ konieczne zatem jest odpowiednie przekształcenie naszej funkcji. To przekształecenie oraz obliczenie całki realizuje następująca funkcja.

```
gaussian_quadrature_2p <- function(f, a, b) {
  g <- function(t) {
    f((b - a) / 2 * t + (a + b) / 2)
  }

  integral <- (b-a)/2*(g(-1/sqrt(3)) + g(1/sqrt(3)))
  return(integral)
}
```

Następnie zapisujemy funkcje obliczające $B(e_i, e_j)$ $L(e_j)$

```
B <- function(i, j) {
  under_integral <-  function(x) {
    return (deriv_e(i,x)*deriv_e(j,x))
```

```
  }
  a = max(0, x_i(i-1), x_i(j-1))
  b = min(x_i(i+1), x_i(j+1))
  return (-gaussian_quadrature_2p(under_integral, a,b) + e(i, 0)*e(j,0))
}

L <- function(j) {
  return (17*e(j, 0))
}
```

Tworzymy układ równań

```
create_A_matrix <- function() {
  A <- matrix(0, nrow = n, ncol = n)
  for (i in 1:(n - 2)) {
    A[i, i] <- B(i, i)
    A[i, i + 1] <- B(i, i + 1)
    A[i + 1, i] <- A[i, i + 1]  # Symmetry
  }
  A[n-1, n-1] = B(n-1, n-1)
  A[n, n] <- 1
  return (A)
}

create_b_vector <- function() {
  b <- sapply(1:n, L)
  b[n]<- 0
  return (b)
}
```

Rozwiązujemy i tworzymy przybliżenie funkcji $u(x)$.

```
create_u_function <- function() {
  w_i <- solve(create_A_matrix(), create_b_vector())

  u_tilde <- function(x) {
    return (3)
  }

  return (function(x) {
    total <- 0
    for(i in 1:n) {
      total <- total +  w_i[i]*e(i, x)
    }
    return (total + u_tilde(x))
  })
}
```

I ostatecznie rysujemy jej wykres:

```
plot_f <- function(f) {
  x_vals <- seq(0, 2, by = 0.01)
  y_vals <- sapply(x_vals, f)

  # Create the plot
  plot(x_vals, y_vals, type = 'l', col = 'blue', lwd = 2,
       xlab = 'x', ylab = 'u(x)', main = 'Graph of u(x)',
       xlim = c(0, 2), ylim = c(0,40),
       xaxs = 'i', yaxs = 'i') # 'i' for internal axis

  grid(nx = NULL, ny = NULL, col = "gray", lty = "dotted")
}
```

```
u = create_u_function()
plot_f(u)
```