Kacper Augustyn

Rozwiązanie równania $-k(x)u''(x) = 0$ metodą elementów skończonych, przy użyciu języka Julia.

```
$ julia
julia> using Pkg
julia> Pkg.add("Plots")
```

```
$ julia mes_solver.jl 100
```

gdzie 100 to N (liczba podziałów w MES) – tę wartość można zmienić

$$-k(x)u''(x) = 0 \qquad x \epsilon [0,2]$$

$$u(2) = 0$$
$$u'(0) + u(0) = 20$$
$$k(x) = \begin{cases} 1, & x \epsilon [0,1] \\ 2, & x \epsilon (1,2] \end{cases}$$

$$-k(x)u''(x) = 0$$
$$dzielimy\ obustronnie\ przez\ k(x), bo\ k(x) \neq 0$$

$$u''(x) = 0$$
$$mnożymy\ przez\ funkcję\ testową\ v\ i\ całkujemy\ obustronnie\ po\ [0,2]$$
$$\int_0^2 u''(x)v(x)dx = 0$$

$$całkujemy\ przez\ części$$
$$[u'(x)v(x)]_0^2 - \int_0^2 v'(x)u'(x)dx = 0$$
$$u'(2)v(2) - u'(0)v(0) - \int_0^2 v'(x)u'(x)dx = 0$$

$$v(2) = 0, bo\ v\ się\ zeruje\ na\ brzegach\ \Omega\ oraz\ u'(0) = 20 - u(0)$$
$$u(0)v(0) - \int_0^2 v'(x)u'(x)dx\ = 20v(0)$$

$$\boldsymbol{B(u,v) = u(0)v(0) - \int_0^2 v'(x)u'(x)dx}$$
$$\boldsymbol{L(v) = 20v(0)}$$

$$e_i = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x\epsilon(x_{i-1}, x_i) \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & x\epsilon(x_i, x_{i+1}) \end{cases}, \qquad x_i = \frac{2i}{n}$$

$$więc$$

$$e_i = \begin{cases} \dfrac{n}{2}x - i + 1\,, & x\epsilon(x_{i-1}, x_i) \\ -\dfrac{n}{2}x + i + 1\,, & x\epsilon(x_i, x_{i+1}) \end{cases}$$

Funkcja $B(u, v)$ jest symetryczna, więc $B(e_i, e_j) = B(e_j, e_i)$

Funkcje w programie

1. Funkcja $e_i(x)$

```
function e(x, i::Int64)::Float64
  if x > x_i(i-1) && x<=x_i(i)
    return N/2*x - i + 1
  elseif x>x_i(i) && x<x_i(i+1)
    return -N/2*x + i + 1
  else
    return 0.0
  end
end
```

2. Funkcja $e_i{}'(x)$

```
function d_e(x::Float64, i::Int64)::Float64
  if x > x_i(i-1) && x<=x_i(i)
    return N/2
  elseif x>x_i(i) && x<x_i(i+1)
```

```
        return -N/2
    else
        return 0
    end
end
```

3. Funkcja $u'v' = e_i{}'(x)e_j{}'(x)$

```
function du_dv(i::Int64, j::Int64)
    return function(x)
        return d_e(x,i)*d_e(x,j)
    end
end
```

4. Funkcja $B(e_i, e_j)$

```
function B(i::Int64, j::Int64)::Float64
    a = max(0, x_i(i-1), x_i(j-1))
    b = max(x_i(i+1), x_i(j+1))

    return e(0,i)*e(0,j)-integral(du_dv(i,j), a, b)
end
```

5. Funkcja $L(e_i)$

```
function L(i::Int64)::Float64
    return 20*e(0,i)
end
```

6. Funkcja licząca $u(x)$

```
function result_func(x::Float64, v)
    res = 0.0
```

```
  for i=1:N
    res += v[i]*e(x, i-1)
  end
  return res
end
```

7. Funkcja całkująca – zwykła metoda prostokątów. Przedział (a, b) dzielę na 10 przedziałów, bo różnica między a i b już i tak jest bardzo mała

```
function integral(f, a::Float64, b::Float64)::Float64
  div::Int32 = 10
  step::Float64 = (b-a)/div
  res::Float64 = 0.0

  for j in 0:div-step
    res += f(a+(j+0.5)*step)*step
  end

  return res
end
```

8. Funkcja zwracająca x_i

```
function x_i(i::Int64)::Float64
  return 2*i/N
end
```

9. Główna funkcja rozwiązująca równanie

```
function solve()
  # initialize L and B matrix
  L_matrix = zeros(Float64, N+1, 1)
```

```
B_matrix = zeros(Float64, N+1, N+1)


# elements on main diagonal have the same value

B_matrix[2,2]=B(1,1)


# count values in matrix B

for i=1:N+1

    for j=1:i

        if i==j

            #B_matrix[i,j]=B_matrix[2,2]

            B_matrix[i,j]=B(i-1,j-1)

        else

            # B(e_i, e_j)=B(e_j, e_i)

            B_matrix[i,j]=B(i-1,j-1)

            B_matrix[j,i]=B_matrix[i,j]

        end

    end

end


# count values in matrix L

for i=1:N

    L_matrix[i,1] = L(i-1)

end


# u(2)=0 condition

L_matrix[N+1,1]=0

for i=1:N
```

```
    B_matrix[N+1,i]=0

    B_matrix[i,N+1]=0

  end

  B_matrix[N+1, N+1] = 1



  # solution

  X_m = B_matrix\L_matrix



  # x axis

  X_plot = [2*i/N for i=0:N]

  # y axis

  Y_plot = [result_func(X_plot[i+1], X_m) for i=0:N]



  plot(X_plot, Y_plot, label="funkcja u")

  xlabel!("x")

  ylabel!("y")

  savefig("plot.png")

  print("\nPlot of function u is saved in file plot.png.")

end
```

Na początku inicjalizujemy macierze B i L i następnie liczymy wartości w odpowiednich komórkach – liczymy wartości tylko nad i na przekątnej, bo $B(u,v) = B(v,u)$.
Następnie uwzględniamy warunki na końcu przedziału i rozwiązujemy układ równań.
Potem obliczamy funkcję u i rysujemy wykres.