On the interval (0,1) we consider a steady-state convection-diffusion-reaction equation, with homogeneous Neumann boundary conditions:

$$-D\frac{d^2u}{dx^2} + \lambda u = f(x), \tag{1}$$

$$-D\frac{du}{dx}(0) = 0, \qquad -D\frac{du}{dx}(1) = 0. \tag{2}$$

1. Assignment 1 Derive the *weak* formulation.
   For this, we multiply Equation (1) by the basis functions ϕ and integrate over the domain.

$$\int_0^1 -D\phi\frac{d^2u}{dx^2} + \lambda\phi u dx = \int_0^1 \phi f(x)dx \tag{3}$$

$$= \int_0^1 -D\left[\frac{du}{dx}\left(\phi\frac{du}{dx}\right) - \frac{d\phi}{dx}\frac{du}{dx} + \lambda\phi u\right]dx =$$
$$\int_0^1 -D\mathbf{n}\cdot\phi\frac{du}{dx}ds + \int_0^1 D\left[\frac{d\phi}{dx}\frac{du}{dx} + \lambda\phi u\right]dx$$

   But using Equation (1) (bc) the first term above is zero, then we have:

$$\int_0^1 \left[D\frac{d\phi}{dx}\frac{du}{dx} + \lambda\phi u\right]dx = \int_0^1 \phi f(x)dx \tag{4}$$

2. Assignment 2 Write the Galerkin formulation of the weak form as derived in the previous assignment for a general number of elements given by n (hence $x_n = 1$). Give the Galerkin equations, that is, the linear system in terms of

$$S\bar{u} = \bar{f}, \tag{5}$$

   all expressed in the basis-functions, $f(x)$, λ and D.
   For the Galerkin formulation we approximate u with the basis functions ϕ_j as:

$$u(x) \sim \sum_{j=1}^{n} a_j\phi_j(x).$$

   Approximating u and substituting $\phi = \phi_i$ in Equation (4) we have:

$$\sum_{j=1}^{n} a_j \int_0^1 \left[D\frac{d\phi_i}{dx}\frac{d\phi_j}{dx} + \lambda\phi_i\phi_j\right]dx = \int_0^1 \phi_i f(x)dx \tag{6}$$

   Then

$$S_{ij}^{e_k} = a_j\int_{e_k}\left[D\frac{d\phi_i}{dx}\frac{d\phi_j}{dx} + \lambda\phi_i\phi_j\right]dx, \qquad S_{ij} = \sum_{k=1}^{nel} S_{ij},$$
$$f_i^{e_k} = \int_{e_k}\phi_i f(x)dx, \qquad f_i = \sum_{k=1}^{nel} f_i^{e_k}.$$

Algorithm 1
function [x]=GenerateMesh(n)
x = linspace(0,1,n);
end

3. Assignment 3 Write a matlab routine, called GenerateMesh.m that generates an equidistant distribution of meshpoints over the interval [0, 1], where $x_1 = 0$ and $x_n = 1$ and $h = \frac{1}{n-1}$. You may use x = linspace(0,1,n).

   Assignment 4 Write a routine, called GenerateTopology.m, that generates a two dimensional array, called elmat, which contains the indices of the ver- tices of each element, that is

$$elmat(i, 1) = i. \qquad for i \in 1, ..., n - 1.$$
$$elmat(i, 2) = i + 1$$

   Next we compute the element matrix S_{elem}. In this case, the matrix is the same for each element, that is, if we consider element e_i .

Algorithm 1
function [elmat] = GenerateTopology(n)
elmat(i, 1) = i;
elmat(i, 2) = i + 1;
end