Solution 2

With this exercise you learn how to deal with Matlab.

1. Consider the problem

$$-((1+x)u')' = 0, \quad x \in I = [0,1], \quad u(0) = 0,\ u'(1) = 1$$

Divide the interval I in n subintervals of equal length and let V_h be the corresponding space of continuous piecewise linear functions vanishing at $x = 0$.

After integration by parts we get the weak solution

$$\int_0^2 (1+x)u(x)'v(x)'\,dx = (1+x)u'(x)v(x)|_0^1 = 2v(1).$$

From the book by Larson & Bengzon (p.36) we *modify* their function `StiffnessAssembler1D` (we do not need `kappa`)

```
function A = StiffnessAssembler1D(x,a)

n = length(x)-1;
A = zeros(n+1,n+1);

for i = 1:n
  h = x(i+1) - x(i);
  xmid = (x(i+1) + x(i))/2; % interval mid-point
  amid = a(xmid); % value of a(x) at mid-point
  A(i,i) = A(i,i) + amid/h; % add amid/h to A(i,i)
  A(i,i+1) = A(i,i+1) - amid/h;
  A(i+1,i) = A(i+1,i) - amid/h;
  A(i+1,i+1) = A(i+1,i+1) + amid/h;
end
```

Here is a solution using `StiffnessAssembler1D`:

```
x=[0:.05:1]';
n=length(x)-1;
a=@(x) 1+x;

A=StiffnessAssembler1D(x,a);

% A is singular!  We have to introduce the left boundary condition u(0)=0
A=A(2:n+1,2:n+1);  %  We remove first row/column of A
```

```
b=zeros(n,1); b(n)=2;  %  Construct right hand side
u=A\b;                 %  Solve the system
u=[0;u];               %  introduce left boundary condition u(0)=0

plot(x,f)
(f(end)-f(end-1))/(x(end)-x(end-1))  % notice that the Neumann boundary
                                     % condition is NOT satisfied accurately !
```

2. **Generation of an own geometry and mesh.**

   Generate Swiss cross with `swisscross.m`

```
g = ...
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2;
  0, 0, 1, 1, 2, 2, 1, 1, 0, 0,-1,-1;
  0, 1, 1, 2, 2, 1, 1, 0, 0,-1,-1, 0;
  0,-1,-1, 0, 0, 1, 1, 2, 2, 1, 1, 0;
 -1,-1, 0, 0, 1, 1, 2, 2, 1, 1, 0, 0;
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1;
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
```

   Then,

```
[p,e,t] = initmesh(g,'hmax',0.5); % create mesh
pdemesh(p,e,t), axis square  % look at the mesh

[p,e,t] = refinemesh(g,p,e,t);
p=jigglemesh(p,e,t);            % improve mesh quality

pdemesh(p,e,t), axis square  % look at what has changed
```