SF2520 - Laboratory 5

Goyens Florentin & Weicker David

3th December 2015

Introduction

In this report we present the results of Lab 5. We had to solve an elliptic problem using first Matlab and then Comsol Multiphysics.

$$\Delta T = 0, \qquad (x,y) \in \Omega$$

$$T(0,y) = 300, \quad T(4,y) = 600, \quad 0 \leq y \leq 2 \tag{1}$$

$$\frac{\partial T}{\partial y}(x,0) = 0, \quad \frac{\partial T}{\partial y}(x,2) = 0, \quad 0 < x < 4 \tag{2}$$

1 Matlab finite difference solution

The script that solves the problem is `laplace.m` and is available at the end of the report with the subroutine `sol.m`. The boundary conditions in equation (2) are treated with ghost points. The conditions that the ghost point has the same value as the last point in the *interior* of the domain. This is given by the discrete first derivative set to zero. We have numbered our unknowns vertically on the grid so that the band size is $M+2$ which is smaller than N (the band we obtain when numbering the unknowns horizontally).

Here is the output of our script. With figures 1 and 2.

```
>> laplace
T(2,1) = 450.0000000000003979 for h = 0.2
T(2,1) = 449.9999999999993747 for h = 0.1
```

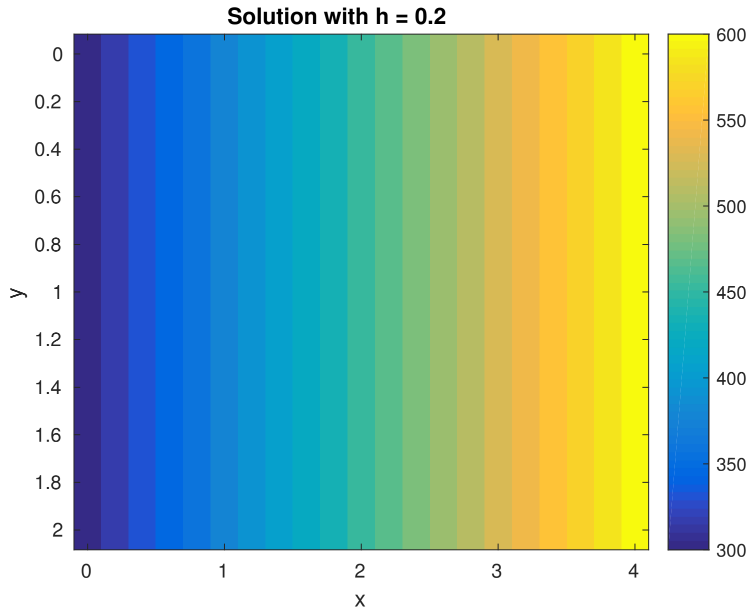

Figure 1: Solution with matlab for $h = 0.2$

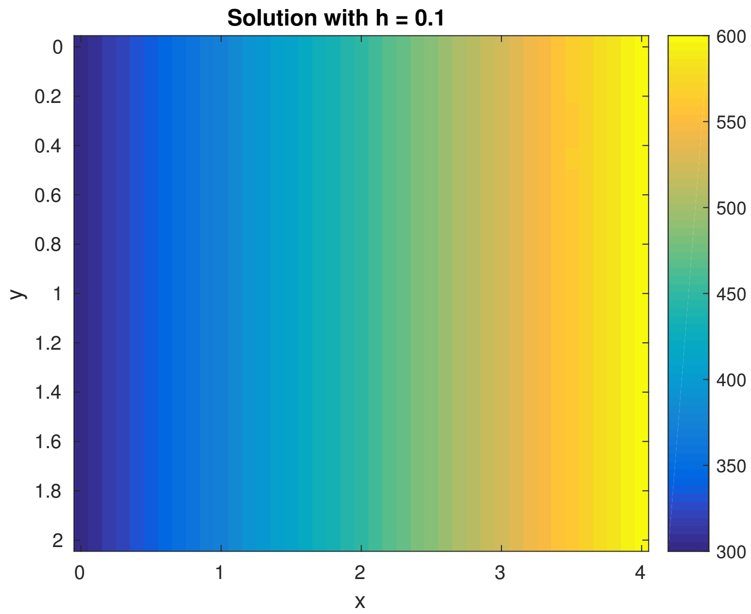

Figure 2: Solution with matlab for $h = 0.1$

We see that the exact value at $(2, 1)$ is close to 450. It is clear by symmetry that the solution does not depend en the y-coordinate. Therefore the equation reduces to

$$\frac{\partial^2 T}{\partial x^2} = 0.$$

The solution is a plane

$$T(x, y) = T(x) = ax + b.$$

The boundary conditions $T(x = 0) = 300$ and $T(x = 4) = 600$ allow to find that $T(x) = 300 +$

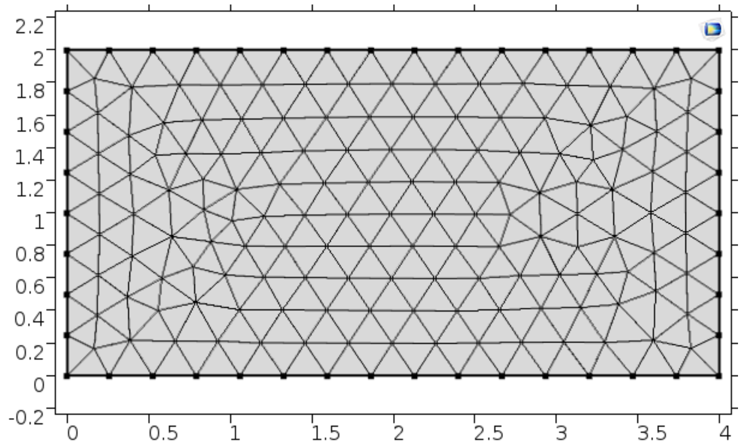

Figure 3: "Normal" mesh for the rectangular geometry

$75x$. The solution respects boundary conditions (1) and (2). It is now clear that $T(2,1) = 450$. The (notably small) errors in the numerical solution are *only* due to floating point computation errors.

2 Comsol Multiphysics solution

In this section, we are going to solve the problem presented above with Comsol Multiphysics.

This is a Laplace equation and Comsol have already a nice setup for us. All we have to do is impose boundary conditions and generate the mesh. By default, Comsol sets zero flux boundary conditions so we just have to specify the Dirichlet conditions on the left and right sides. Once we have done this, we first use a mesh of type "normal". Figure 3 shows what the "normal" mesh looks like for our geometry.

This mesh contains 316 triangles and 679 nodes.mmmmmm....

Figure 4 shows the solution for this choice of mesh. We can see that is it very similar to the one in section 1 which is a linear function between $x = 0$ and $x = 4$. We also used a probe to check the temperature at $(x, y) = (2, 1)$.

$$T(2,1) = 450.0000000001594$$

That is very close to the analytic value and the error is only due to floating point computation.

We are now going to refine the mesh. We switch from "normal" to "fine". There are now 476 triangles and 1022 nodes.je ne te crois pas...

We have another $T - value$ and:

$$T(2,1) = 450.000000000236$$

We can see that the two values are extremely close to each other and as we said, the small difference is due to floating point errors.

We finally restart all over again with a "normal" mesh. After having set all needed values, Comsol plots the solution given in figure 4.

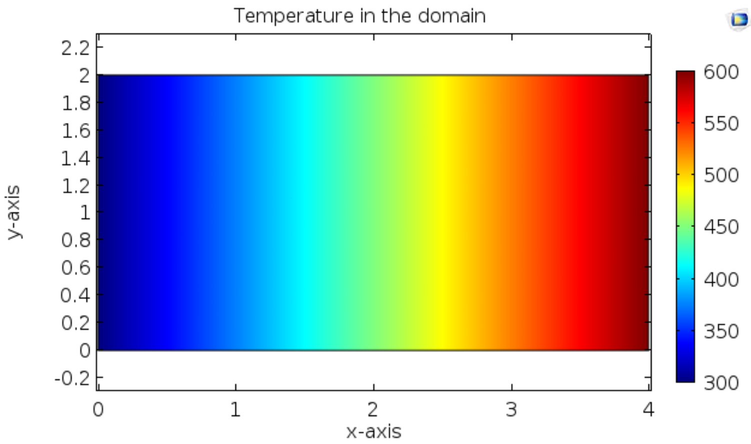

Figure 4: Solution of the given problem with Comsol

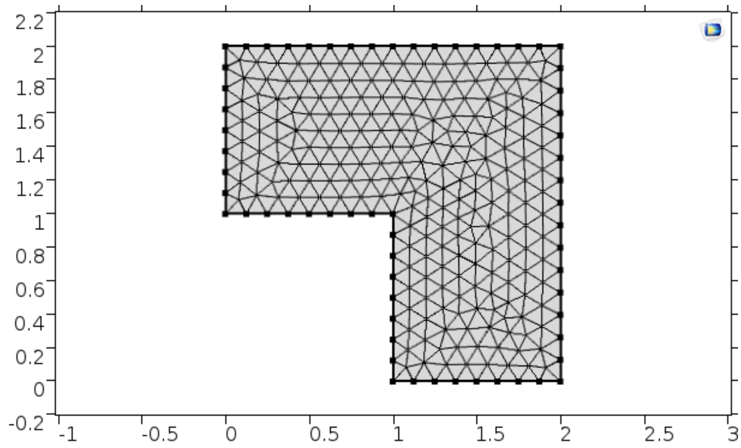

Figure 5: Geometry and mesh for the L-shaped domain

3 More Comsol

We are now going to consider a more complex geometry. The L-shaped area and the corresponding "normal" mesh are given in figure Lmesh.

This mesh contains 489 triangles. The solution computed by Comsol is given in figure 6.

Codes

```
% Script for the first task of homework. Runs the problem for 2 different
% stepsizes.
% N is x-direction. M is y-direction
% Florentin GOYENS & David WEICKER
% LAB 5

close all;
format long;

% Solution with h = 0.2
```

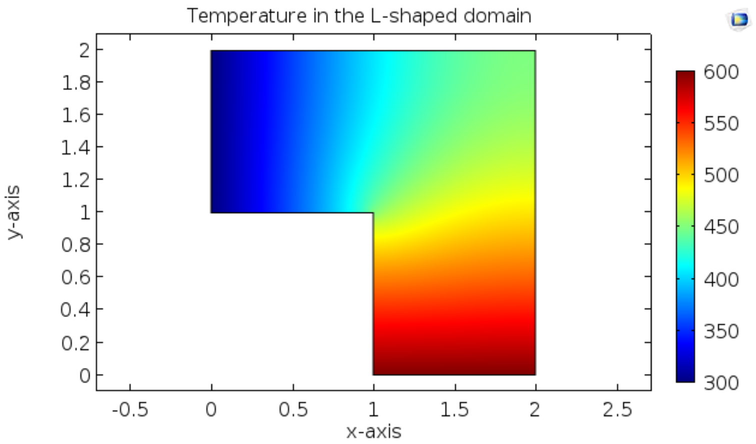

Figure 6: Solution for the L-shaped domain

```
 N = 19; M = 11;
 U02 = sol(M,N);
 X = [0  4]; Y = [0  2];
 figure();
 imagesc(X,Y,U02);  title('Solution with h = 0.2');
 colorbar(); xlabel('x'); ylabel('y');
 Tmiddle02 = U02(7,11); % U is 13x21
 fprintf('T(2,1) = %4.16f for h = 0.2 \n',Tmiddle02);

% Solution with h = 0.1
 N = 39; M = 21;
 U01 = sol(M,N);
 X = [0  4]; Y = [0  2];
 figure();
 imagesc(X,Y,U01);  title('Solution with h = 0.1');
 colorbar(); xlabel('x'); ylabel('y');
 Tmiddle01 = U01(12,21); % U is 23x41
 fprintf('T(2,1) = %4.16f for h = 0.1 \n',Tmiddle01);
```

```
function U = sol(M,N)
% Solves the problem and returns the solution U reshaped as a matrix.
% N is x-direction. M is y-direction
% Florentin GOYENS & David WEICKER
% LAB 5

 n = N*(M+2);% size of system
 e = ones(n,1);
 band = M + 2;
 A = spdiags([-e -e 4*e -e -e],[-band -1 0 1 band],n,n);

 %1) add known values on sides in term b
 b = zeros(n,1);
 b(1:M+2) = 300;
 b(((N-1)*(M+2)+1):end) = 600;

 % 2) Fix ghost points and some coefficients
 for i = 1:N-1
     A(M+2 + (i-1)*(M+2),M+2 + (i-1)*(M+2) +1) = 0; % upper side of domain
     A(M+2 + (i-1)*(M+2),M+2 + (i-1)*(M+2) -1) = -2;% upper side of domain
     A(1+i*(M+2), 1+i*(M+2) -1) = 0;% lower side of domain
     A(1+i*(M+2), 1+i*(M+2) +1) = -2;% lower side of domain
 end
 A(N*(M+2), N*(M+2) -1) = -2;
 A(1,2) = -2;
```

```
U = A\b;
U = reshape(U,M+2,N);
U = [300*ones(M+2,1) U 600*ones(M+2,1)];
end
```