Task K1

Emil Ljungskog & Simon Sigurdhsson

This report discusses the solution of the heat equation,

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+b=0,$$

discretized as

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b\Delta x\Delta y \tag{1}$$

and solved using the Gauss-Seidel iterative method. The problem is considered on a plate with $L = H = 1$, $k = 2(1 + 20T/T_1)$ and the heat source $b = 15(c_1 - c_2T^2)$ ($c_1 = 25$, $c_2 = 0.1$) over the whole domain. Boundary conditions on boundaries 1, 2 and 4 (south, east and west) are of the Dirichlet type with $T_1 = 10$, $T_2 = 20$, $T_4 = 10(1 + 2y/H)$. Boundary 3 is governed by a homogeneous Neumann condition, *i.e.* $\frac{\partial T}{\partial y} = 0$.

Implementation

The code, written in MATLAB, consists of a number of functions to implement meshing of the domain, an iterative solver and visualization of the obtained results. The code starts by generating a mesh (described below), which defines the faces of the control volumes. A node was then placed in the center of each cell. As in general for the finite volume method, all values in a cell is stored in its node which means that the values at the faces has to be interpolated.

Generating a clever mesh

Generating a simple mesh is trivial; one simply divides the domain into equally sized intervals in each direction. However, in order to obtain more accurate solutions without "wasting" computational power, the mesh should be finer in those areas where the problem results in a solution with large gradients, and coarser in those areas in which it does not.

Since the given problem has a solution with large gradients near the Dirichlet boundaries, as illustrated by figure 1a on page 4, these areas should have a finer mesh. A mesh with varying resolution can be created by considering a geometric progression with a given sum (the length of the domain in the relevant direction), common ratio λ and a given number of elements n. The size of the smallest element in a mesh created from this geometric progression (here, in the y direction) can be computed as

$$\bar{y} = H\frac{\lambda_y - 1}{\lambda_y^{n_y} - 1}.$$

The position of each node in the mesh can then be iteratively calculated as

$$y_i = y_{i-1} + \bar{y}\lambda_y^{i-1},$$

where $y_0 = 0$ and $y_{n_y} = H$.

In the x direction, $L/2$ is used as the domain length and $n_x/2$ elements are used along with mirroring (*i.e.* $x_i = x_{n_x - i}$) to obtain the fine mesh at both ends.

The source term and face values

By rewriting the source term $b\Delta x\Delta y$ as

$$b\Delta x\Delta y = S_U + S_P T_P = 15c_1 - 15c_2 T_p \cdot T_P,$$

where T_p is the temperature calculated in the previous iteration and T_P is the one to be solved for, the discretized equation (1) can be written as

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + S_U \tag{2}$$

where $S_P = -15c_2 T_P$ has been "hidden" in a_P, as it turns out that

$$a_P = a_E + a_W + a_N + a_S - S_P$$

where $a_E = \frac{k_e \Delta y}{\delta x_e}$ and so forth.

The heat conduction k_e at the face is interpolated using the central differencing scheme as

$$k_e = f_e k_E + (1 - f_e)k_P$$

with the interpolation factor $f_e = \frac{\Delta x}{2\delta x_e}$. The same approach is of course used for the other face values as well.

The Gauss-Seidel solver

The Gauss-Seidel method is a simple, and quite clever, way of solving a system of equations. Applying the method on (2) requires prescribing the temperature on the

Dirichlet boundaries and "guessing" a temperature of 0 for the rest of the domain (the Neumann boundary condition is implemented implicitly by setting $a_N = 0$ on the boundary).

The iterative process then begins with solving for T_P in node 1 using the boundary values and the initial guess, followed by solving for T_P in node 2 using the recently computed value in node 1 together with boundary values and guesses. This process is repeated until the temperature has been solved for in all nodes. A convergence check is then performed (see below); if the convergence criteria is met, the solution has been found and the process is aborted. If not, the process is repeated starting with node 1 using the temperatures obtained in the previous iteration.

Since both the source and the heat conductivity is temperature dependent, the coefficients and sources in (2) have to be recomputed for every node.

Convergence criteria

To determine if the obtained solution is good enough, we need to have some kind of accurate measurement of the error. For this purpose a residual R is introduced:

$$R = \sum_{\text{all nodes}} |a_E T_E + a_W T_W + a_N T_N + a_S T_S + S_U - a_P T_P|.$$

The resudual is normalized using the total heat flux F into the domain,

$$F = \sum_{\substack{\text{boundary} \\ \text{nodes}}} |q_i|.$$

The convergence criteria to be met is then

$$\frac{R}{F} \leq \varepsilon,$$

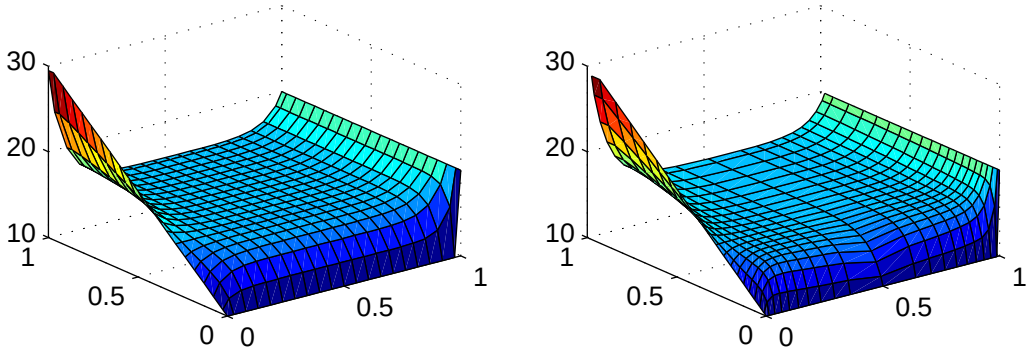
where $10^{-4} \leq \varepsilon \leq 10^{-2}$.

Results

Standard case

For the standard case described in the problem introduction, a surface plot of the solution for a 20×20 mesh can be seen in figure 1b on the next page. Furthermore, a contour plot and a plot of the heat flux can be seen in figures 5 to 6 on page 6.

Clearly, the temperature seems to be constant in the central part of the domain. This is due to the temperature dependency of the source term yielding a state of equilibrium when $15c_1 - 15c_2 T_p^2 = 0 \Leftrightarrow T_P = \sqrt{\frac{c_1}{c_2}} \approx 15.8$, which is very close to the temperature in the center of the domain.



(a) *A simple, linear mesh* **(b)** *A mesh based on geometric progressions*

Figure 1: *Two different types of mesh*

Changing the mesh size

Figure 2 on the next page shows the solution computed with a smaller 10×10 grid and a larger 40×40 grid. Comparing this with the solution in figure 1b which was computed with a 20×20 mesh, it is clear that increasing the mesh size also increases the quality (or rather, resolution) of the solution. Decreasing the mesh size does the opposite.

The number of iterations required to compute the solution increases with the mesh size (25, 30 and 40 iterations), and the time spent in each step as well as the total time scales accordingly (0.1 s, 0.5 s and 3 s).

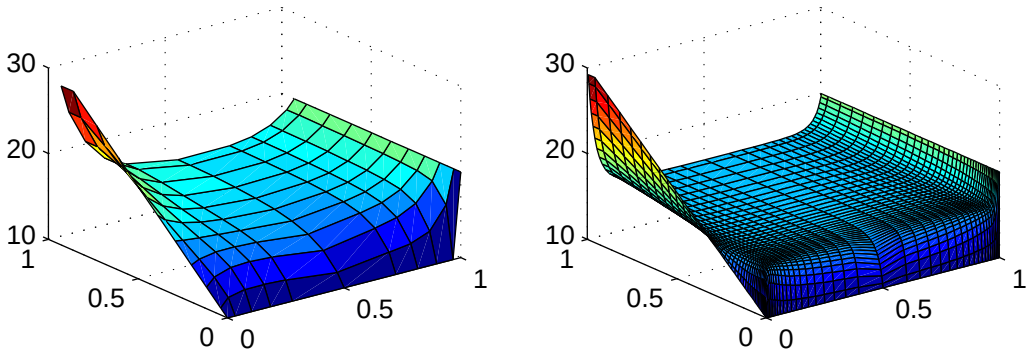
Varying heat conductivity k

If the heat conductivity k is increased or decreased by a factor 100, the solution changes according to figure 3 on the next page. It can be seen that the boundaries seems to have a smaller impact on the solution inside the domain for the lower value of k compared to the standard case, and vice versa for the increased k.

This is no surprise, since a higher k means stronger diffusion and a higher heat flux, which should lead to the obtained results.

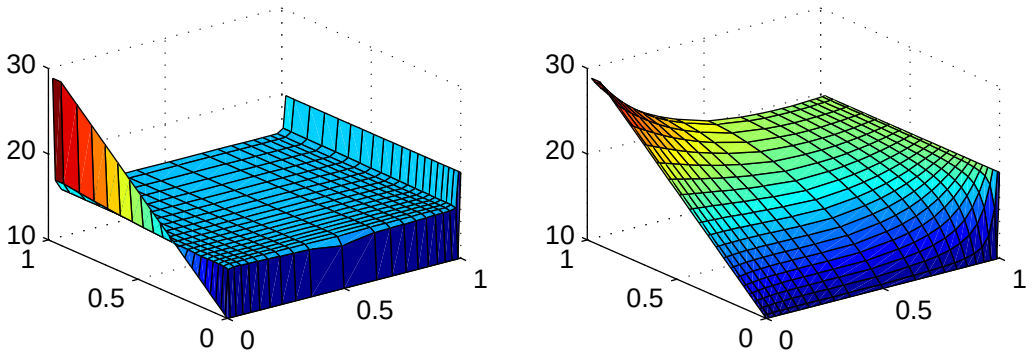
Changing boundary conditions

As can be seen in figure 4 on the following page, the solution behaves as expected when changing the boundary conditions on boundary 1 and 3. By setting a homogeneous Neumann condition on boundary 1, the solution is flat (at a level where the source yields equilibrium) according to figure 4b on the next page.



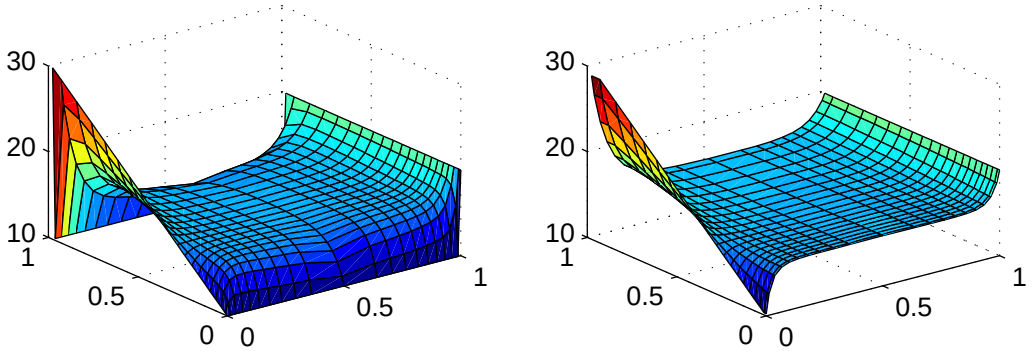
(a) *A coarse mesh,* 10×10 **(b)** *A fine mesh,* 40×40

Figure 2: *Solution with different meshes*



(a) *Smaller* k *(*$k' = 0.01k$*)* **(b)** *Larger* k *(*$k' = 100k$*)*

Figure 3: *Solution for different values of* k



(a) *Dirichlet condition on boundary 3* **(b)** *Neumann condition on boundary 1*

Figure 4: *Solution with different boundary conditions*

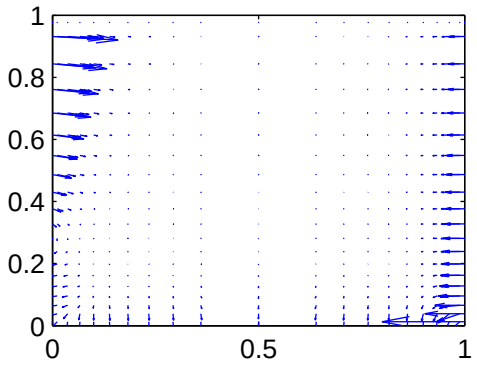


Figure 5: *The flux of the solution*

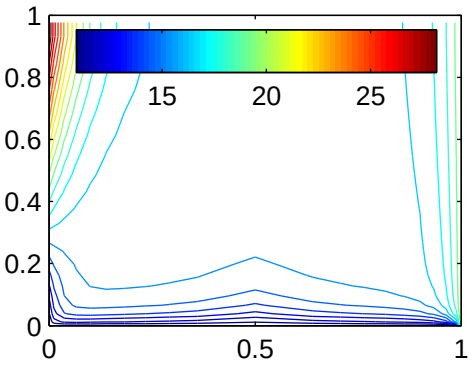


Figure 6: *Contour plot of the solution*

Letting boundary 3 be of Dirichlet type with $T_3 = 10$, the solution in 4a on the preceding page is obtained. Again, this is an expected solution.

Visualizing the flux

To get a view of the heat fluxes in the domain, a vector plot of the flux can be seen in figure 5. Comparing the vector plot with the contour plot in figure 6, it is clear that the arrows representing the flux are always orthogonal to the contour lines and longer where the contour lines are close. Neither of these observations are surprises, since this is a fairly basic result in vector calculus.