



KTH Engineering Sciences

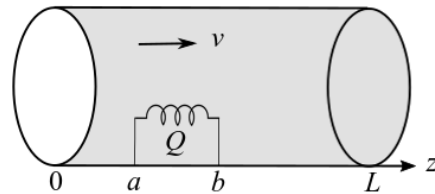
## Computer Exercise 2

### Elliptic equations

In this exercise you will solve elliptic PDEs in one and two dimensions. In the first two parts you will use the finite difference method on a simple geometry (an interval and a rectangle). In the third part you will use Comsol Multiphysics to find solutions in more complicated geometries.

#### Part 1: Finite difference approximation in 1D

Consider a pipe of length  $L$  with a small cylindrical cross section. In the pipe there is a fluid heated by an electric coil. The heat is spreading along the pipe and the temperature  $T(z)$  at steady state is determined by the convection–diffusion equation



$$-\frac{d}{dz} \left( \kappa \frac{dT}{dz} \right) + v\rho C \frac{dT}{dz} = Q(z),$$

where all parameters are constant:  $\kappa$  is the heat conduction coefficient,  $v$  is the fluid velocity in the  $z$ -direction through the pipe,  $\rho$  is the fluid density and  $C$  is the heat capacity of the fluid. The driving function  $Q(z)$ , modeling the electric coil, is defined as

$$Q(z) = \begin{cases} 0, & 0 \leq z < a, \\ Q_0 \sin \left( \frac{(z-a)\pi}{b-a} \right), & a \leq z \leq b, \\ 0, & b < z \leq L. \end{cases}$$

At  $z = 0$  the fluid has the inlet temperature  $T_0$ ,

$$T(0) = T_0.$$

Beyond  $z = L$  the pipe is poorly insulated and the liquid is cooled down since heat is leaking out to the exterior, which has the temperature  $T_{\text{out}}$ . This assumption can be modeled by the following boundary condition (BC) at  $z = L$ :

$$-\kappa \frac{dT}{dz}(L) = \alpha(v)(T(L) - T_{\text{out}}), \quad \alpha(v) = \sqrt{\frac{v^2 \rho^2 C^2}{4} + \alpha_0^2} - \frac{v\rho C}{2},$$

where  $\alpha_0$  is a heat convection coefficient for the non-convective ( $v = 0$ ) case. Use the following values of the parameters:  $L = 10$ ,  $a = 2$ ,  $b = 4$ ,  $Q_0 = 10$ ,  $\kappa = 0.5$ ,  $\alpha_0 = 8$ ,  $\rho = 1$ ,  $C = 0.75$ ,  $T_{\text{out}} = 20$  and  $T_0 = 50$ .

- (a) Consider first the case  $v = 1$ . Solve the boundary value problem with the finite difference method using MATLAB. Discretize the  $z$ -interval  $[0, L]$  with grid points  $z_j = jh$  where  $h$  is a constant stepsize. Use at least second order accurate approximations for the differential equation and the BCs. Plot the solution  $T(z)$  for  $h = 1, 0.5, 0.25$  and  $0.125$  in the same graph. Note the convergence of the curves in the graph. Verify that it is second order. (You can do this by checking the convergence of the approximation in just one  $z$ -point, for instance at the end of the pipe,  $z = L$ . You may need to take even smaller  $h$  for this.)
- (b) Now solve the problem for  $v = 0.1, 0.5, 1, 10$ , and plot the solutions in the same graph. Interpret the curves based on the physical problem they model. Why do they look like they do? Are the results as expected? Note: For higher  $v$  you will need to take a smaller  $h$  to get the correct behavior of the solution close to  $z = L$ . Why is this? Report how large  $h$  you take for the different  $v$ .

OBS! For full credit your implementation should exhibit second order accuracy when the grid is refined as in (a).

*Hints:* If you have used a second order stencil and second order approximation of the boundary conditions but still only observe a first order convergence rate it is quite likely that you have made a small mistake somewhere. (This is very easy to make!) Here are two common minor mistakes that will turn a second order method into a first order method:

- Your implementation of the number of unknowns  $N$ , the step length  $h$  and the pipe length  $L$  are not exactly compatible. You may have used  $h = L/N$  when your definition of  $N$  would require  $h = L/(N + 1)$  for instance, or chosen  $h$  such that  $L/h$  is not an integer.
- You may evaluate the source function  $Q$  in slightly the wrong points. You might use  $Q(z_j)$  when it should actually be  $Q(z_{j+1})$  or something similar.

Additionally, even if you have a second order method, you may observe first order convergence if you do not read off the  $T$  value in exactly the same point for the different grids. Instead of  $z = L$  you might be looking at  $z = L - h$  for instance.

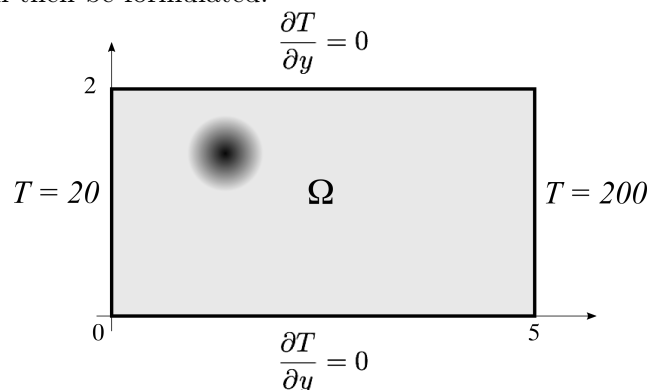
## Part 2: Finite difference approximation in 2D

In this part we consider heat conducted through a 2D rectangular metal block occupying the region  $\Omega = [0 \leq x \leq 5, 0 \leq y \leq 2]$  in the  $xy$ -plane. The block is kept at constant room temperature  $T = 20$  at  $x = 0$  and at  $T = 200$  at  $x = 5$ . It is insulated at the other two sides. An external source modeled by the function  $f(x, y)$  heats the block. The following elliptic problem for the temperature distribution  $T(x, y)$  can then be formulated:

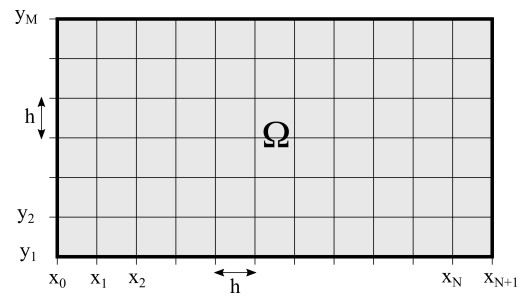
$$-\Delta T = f, \quad (x, y) \in \Omega,$$

with boundary conditions

$$\begin{aligned} T(0, y) &= 20, & 0 \leq y \leq 2, \\ T(5, y) &= 200, & 0 \leq y \leq 2, \\ \frac{\partial T}{\partial y}(x, 0) &= 0, & 0 < x < 5, \\ \frac{\partial T}{\partial y}(x, 2) &= 0, & 0 < x < 5. \end{aligned}$$



Use the finite difference method to solve this problem. The approximation should be of at least order two. Discretize the rectangular domain into a quadratic mesh with the same, uniform, stepsize  $h$  in the  $x$ - and  $y$ -directions. A suggested numbering of the grid points is given in the figure on the right, where for instance,  $N = 49$  and  $M = 21$  if  $h = 0.1$ . Solve the resulting linear system with backslash in MATLAB.



- Compute the solution  $T(x, y)$  with  $f \equiv 0$  and  $h = 0.1$ . Visualize  $T(x, y)$  with colors, using the Matlab function `imagesc`. What is the  $T$ -value in the block at  $(x, y) = (3, 1)$ ?
- Derive the analytic solution when  $f \equiv 0$  and prove that it indeed solves the PDE. Explain why the numerical method gives the *exact* solution for this problem.
- Solve the problem (numerically) with

$$f(x, y) = 50 + 500 \exp(-2(x - 1)^2 - (y - 1.5)^2).$$

Report the  $T$ -values obtained at  $(x, y) = (3, 1)$  when you use  $h = 0.1$ ,  $h = 0.05$  and  $h = 0.025$  (three  $T$ -values). (You may want to check additional values to verify your convergence rate.) Here it is important to use MATLAB's `sparse` format for the matrices. Otherwise the problems with small  $h$  take very long time. Visualize the solution as above for  $h = 0.05$ . This time also plot it with the `mesh` and `contour` commands. (You do not need to find the analytic solution for this case!) Comment on your results. Are they as expected?

OBS! For full credit your implementation should exhibit second order accuracy when the grid is refined.

*Hint:* If you have problem getting second order accuracy, please look at the hints for the 1D case above. In 2D it is even easier to make small mistakes of the types mentioned there. Be particularly careful to read off the value in the right point,  $(x, y) = (3, 1)$  and not something like  $(x, y) = (3 + h/2, 1 - h/2)$ . (This is not possible for any choice of  $N$  and  $h$ !)

### Part 3: Comsol Multiphysics

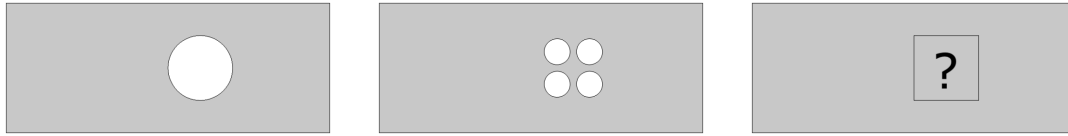
- (Warm-up.) Solve the problem in Part 2c (when  $f \neq 0$ ) with Comsol Multiphysics. Draw the geometry. Set the PDE coefficients. Specify boundary conditions. Generate the mesh. Compute the solution and plot a 2D-graph of the solution.

Check your numerical result: what is the  $T$ -value at the point<sup>1</sup>  $(3, 1)$ ? How many elements (triangles) have been generated in the mesh? Make a refinement of the grid and find again  $T(3, 1)$ . How many elements are there now? Continue to refine until you are sure that the value has 3 correct decimals. Compare with your own solutions above.

- Instead of keeping the rightmost side at 200 degrees we now insulate it. This means that the temperature in the block will increase. Mathematically, we should have the following new boundary condition at  $x = 5$ ,

$$\frac{\partial T}{\partial x}(5, y) = 0.$$

<sup>1</sup>To read off the value at a point  $(x, y)$  in Comsol you can use a Domain Point Probe. Right click on **Definition** in the Model builder and select **Probes/DomainPointProbe** in the popup menu.



**Figure 1.** Configuration in (c) with one hole (left), in (d) with four holes (middle) and your own design (right).

Change this in Comsol. The remaining boundary conditions should not change.

Plot the solution and compute the average temperature on the boundary  $x = 5$ . Use a Boundary Probe, which you find next to the Domain Point Probe used above.

- (c) To cool down the right boundary of the block, a hole is drilled in it and cool liquid is allowed to flow through it. The hole is centered at  $(x, y) = (3, 1)$  and has radius 0.5. See left figure above. The cooling is modeled by the following boundary condition on the edge of the hole

$$\frac{\partial T}{\partial n} = T_0 - T,$$

where  $\partial/\partial n = \hat{n} \cdot \nabla$  is the normal derivative and  $T_0 = 20$  is the liquid temperature. Use the boundary condition **Flux/Source** in Comsol to set this. (Keep all other boundary conditions as before in (b).)

Again, plot the solution and compute the average temperature on the boundary  $x = 5$ . How much is the temperature reduced?

- (d) The cooling will be more efficient if the single hole is replaced by four smaller holes. See middle figure. Those holes have radius 0.2 and are centered in  $(x, y) = (3 \pm 0.25, 1 \pm 0.25)$ . Solve this case and, as in (c), plot the solution and compute the average temperature on the boundary  $x = 5$ . How much is the temperature reduced?

Also plot the mesh for this case and comment on the triangle sizes: Why are they big in some parts and small in others?

- (e) Finally, design a configuration of your own which reduces the temperature even more: The average boundary temperature of your design should be less than 75% of the temperature you observed in (c). You are only allowed to make changes to the geometry inside the square  $[3 \pm 0.5, 1 \pm 0.5]$ <sup>2</sup>. The Comsol feature **Transforms/Array** (right click on Geometry) can be useful.

Plot the solution of your configuration and report the average boundary temperature. Make sure that the temperature has at least one correct decimal.

Comment on the configuration, the result and its limitations, e.g.: How did you come up with it? How can it be further improved? Are the results physically relevant? Is it possible to manufacture? Are there any numerical limitations for simulating it?

*Hint:* Keep in mind the result from thermal physics that the surface area to volume ratio is an important determinant of the cooling rate.

<sup>2</sup>It is quite possible even to come down to below 50% with this limitation, if you put in some work.