Exercise 2
Thermal energy in a cavity flow

41319 - Computational Fluid Dynamics

by

Thomas Sørensen s113354

Franz Hastrup-Nielsen s113399

Department of Mechanical Engineering -
Fluid Mechanics, Coastal & Maritime Engineering
Technical University of Denmark
25. October 2015

Contents

1 Development of the finite-volume discretization

In order to complete this report, a two-dimensional finite volume solution for the steady-state transport of thermal energy in a specified flow field through a square cavity has been developed. The value of interest here is the temperature, and the governing equation is the following:

$$Pe \int_S \boldsymbol{n} \cdot (\boldsymbol{u}T) dS = \int_S \boldsymbol{n} \cdot \nabla T dS \tag{1.1}$$

The implementation consists of the two following problems with convective velocity fields and boundary conditions:

Problem 1

- Uniform flow, $[u, v] = [1, 0]$
- North wall: Homogeneous Neumann BC, $\frac{\partial T_n}{\partial y} = 0$
- South wall: Homogeneous Neumann BC, $\frac{\partial T_s}{\partial y} = 0$
- West wall: Homogenous Dirichlet BC, $T_w = 0$
- East wall: Inhomogeneous Dirichlet BC, $T_e = 1$

For a low Peclet number flow, this problem will result in the temperature field seen in figure 1.1.

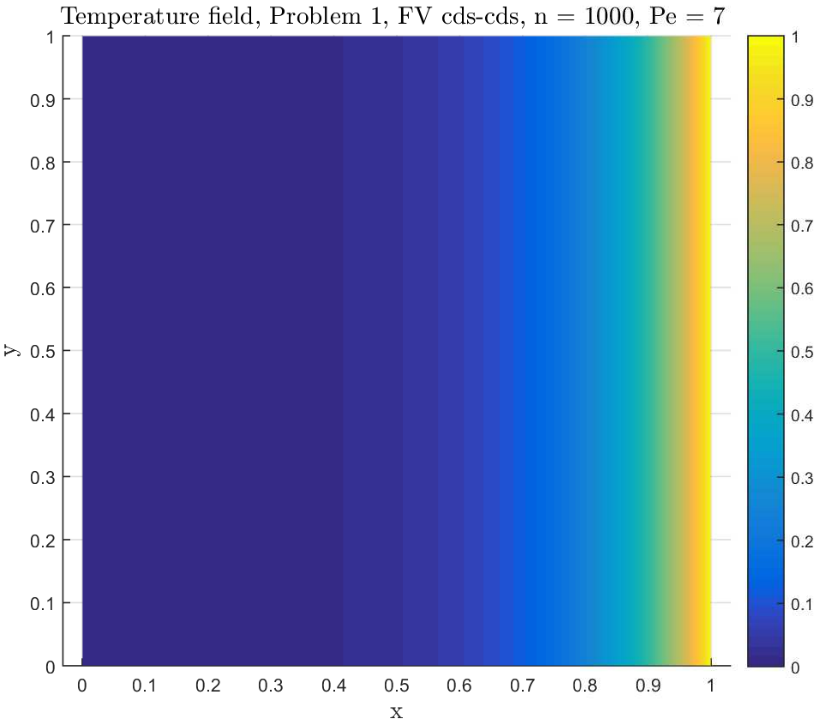


Figure 1.1: The temperature field plotted without boundary conditions for a low peclet number flow

Problem 2

- Corner flow, $[u, v] = [-x, y]$
- North wall: Homogeneous Neumann BC, $\frac{\partial T_n}{\partial y} = 0$
- South wall: Inhomogeneous Dirichlet BC, $T_s = x$
- West wall: Homogenous Dirichlet BC, $T_w = 0$
- East wall: Inhomogeneous Dirichlet BC, $T_e = 1$

For a low Peclet number flow, this problem will result in the temperature field seen in figure 1.1.

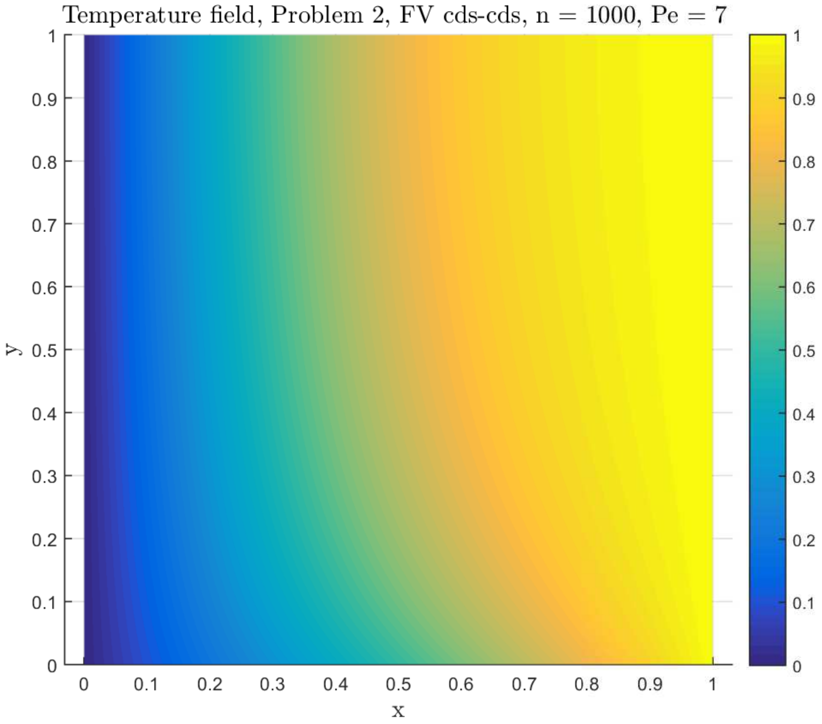


Figure 1.2: The temperature field plotted without boundary conditions for a low peclet number flow

Both these problems will be solved using two different schemes. For diffusion the CDS scheme will be used in all cases and both UDS and CDS schemes will be used for the convection.

2 Solving for uniform flow

This section investigates the uniform flow in problem 1 as described in section 1.

2.1 Convergence

For the uniform flow, the solution will collapse into a 1d-solution. Therefore can the solution be compared to the analytical 1d-solution shown in equation 5.9[1] and a convergence trend can be found. For a Peclet number of 10 the convergence of the cds-cds and the uds-cds schemes are investigated to verify that the solution converges. The results at low n is not weighted in the fit to obtain the expected convergence rates of the two schemes.

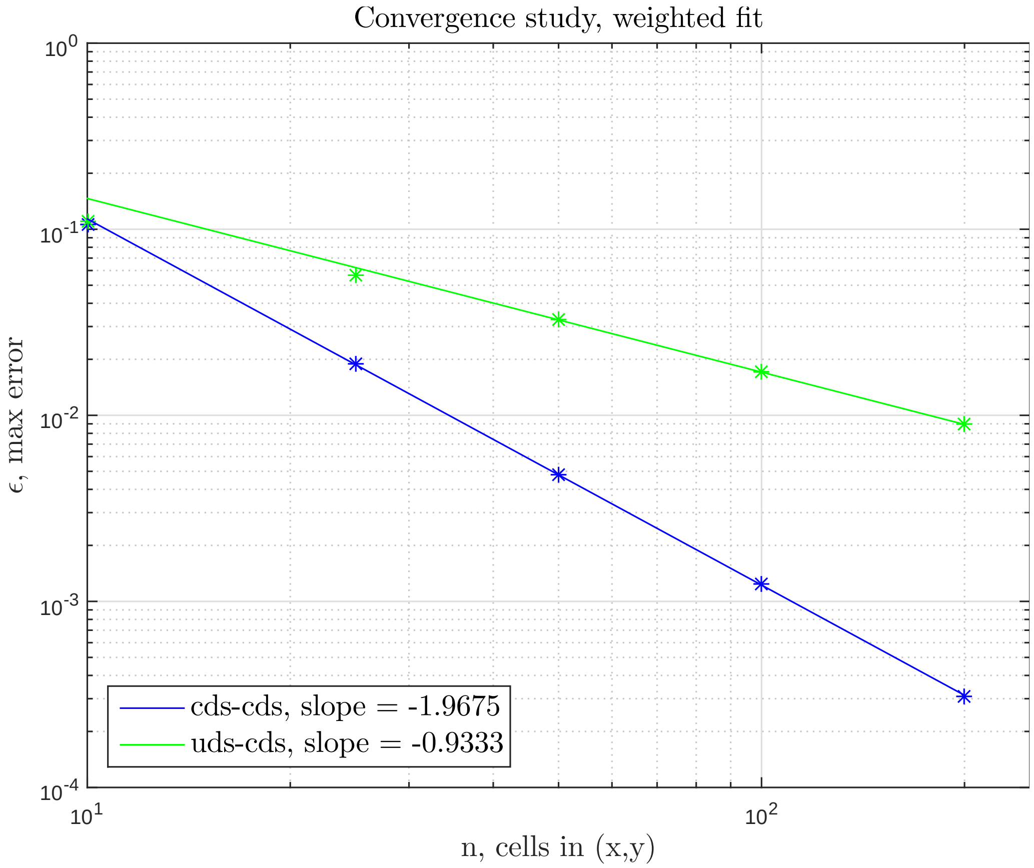


Figure 2.3: Rate of convergence for cds-cds and uds-cds schemes

The convergence rates are very close to the expected values and confirms that the solution converges at a rate of the stencil size minus 1. Hence the cds-cds scheme is second-order and the uds-cds scheme is first order.

2.2 Optimal SOR relaxation method

Through a spectral radius analysis at small values of n an approximated expression for the optimal SOR relaxation factor is found. The function used is:

$$\omega_{opt} \approx \frac{c_1}{1 + c_2 sin\left(\frac{\pi}{n}\right)} \tag{2.2}$$

Where c_1 and c_2 is computed by a weighted least-square method. These coefficients are found for the following three cases:

- i) Pe = 0, cds-cds scheme (no convection so uds-cds scheme can also be used)
- ii) Pe = 10, cds-cds scheme
- iii) Pe = 10, uds-cds scheme

All three cases are solved with the Jacobi method for small numbers of n. The Jacobian spectral radius and the optimal omega from this method are computed. The values, n and $\omega_{opt,Jacobi}$ are obtained and c_1 and c_2 can be computed from the linear relation:

$$\omega_{opt} = \left(-\omega_{opt,Jacobi} sin\left(\frac{\pi}{n}\right)\right) c_2 + c_1 \tag{2.3}$$

The optimal relaxation parameter needs to be below 2 for the SOR method to converge. The function needs to be able to define the ω_{opt} at large n and it is important to make sure that it remains below 2. The c_1 and c_2 values are found by also implementing a value at a larger n with ω_{opt} just below 2, ($n = 10^8$, $\omega_{asymp} = 2 - 10^{-4}$). The result can be seen in figure 2.4

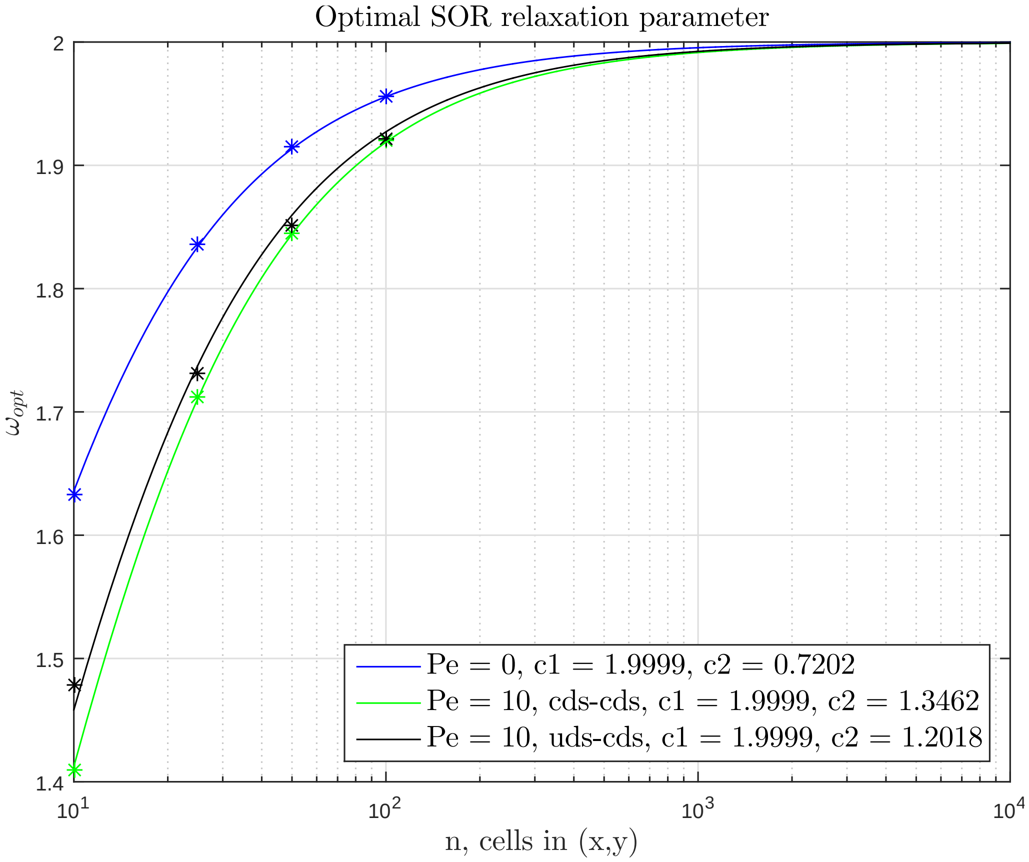


Figure 2.4: The optimal SOR relaxation parameter as function of the number of cells in (x,y) for the three different cases

The c_1 and c_2 values are tested for any reasonable n, to ensure that no ω_{opt} is above 2. The optimal values at different n can be seen in table 2.1.

Table 2.1: Optimal SOR relaxation parameter for varying number of cells

Case / n, cells in (x,y)	10	25	50	100	1e8
ω_{opt}: Pe = 0, CDS-CDS	1.6326	1.8356	1.9145	1.9564	1.9999
ω_{opt}: Pe = 10, CDS-CDS	1.4092	1.7122	1.8453	1.9197	1.9999
ω_{opt}: Pe = 10, UDS-CDS	1.4787	1.7317	1.8514	1.9214	1.9999

2.3 Comparison of the iterative SOR method to MATLAB's direct solver

The performance of the iterative point SOR method with optimal relaxation factor found in section 2.2 are compared to the direct solver in Matlab. The computational costs are the time used for solving the linear system of equations:

$$\mathbf{A} \cdot \vec{T} = \vec{s} \tag{2.4}$$

A scaling of the costs for the different solvers are found with regards to the size of the system solved in n^2.

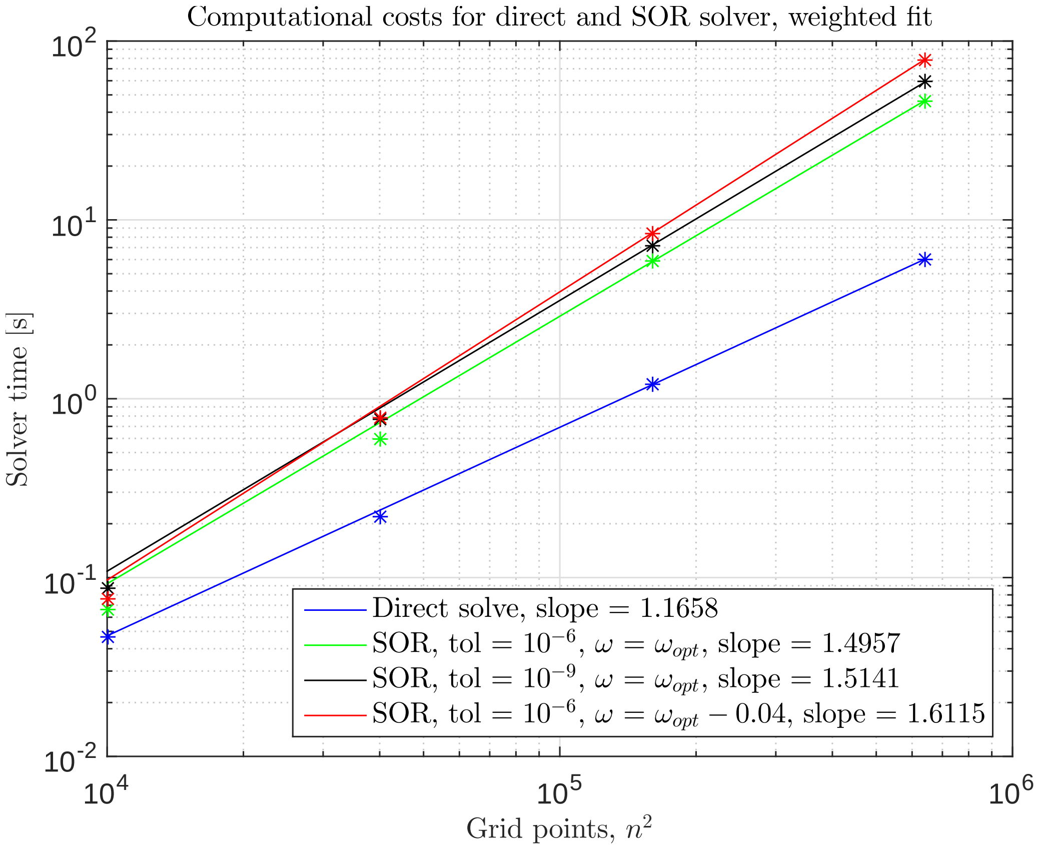


Figure 2.5: Computational cost for different solvers. Different tolerances and ω_{opt} values have been used

The slopes in figure 2.5 shows the computational cost for the different solving methods.
At page 97 in the teaching material[1] is an overview of the computational costs for different

solvers. From this it is clear that the optimal SOR solver has a computational cost of $O(n^{1.5})$. This is the same as in figure 2.5. The computational cost depends on the tolerances used for the SOR solver, which is clear in the figure. As expected there is a small increase in the computational cost when solving for smaller tolerances as the residual error is smaller.
The last case of the SOR solver with the offset of the optimal relaxation factor and large tolerances shows the influence of ω_{opt}. By using an offset value the computational costs increase. Clearly time can be saved in the computations by choosing the correct relaxation factor.
The computational costs using Matlab's direct solver are lower than the SOR method which also is expected as the SOR method is iterative and Matlab solves the problem directly. This direct solution is close to being as effective as the Multigrid solver with a computational cost of $O(n)$. The superiority of Matlab's direct solver is due to the simplicity of this problem, a more complex problem would properly result in a different ranking.

These solutions used CDS convection. UDS convection could also be used and the same results would be expected. The size and the structure of the source term vector, s, and the system matrix, A, will not change when changing the scheme, hence the computational time will be the same.

3 Solving for corner flow

The flow and boundary conditions are now changed, and the conditions described under problem 2 in section 1 are implemented.

3.1 GHC and convergence of diffusive heat flux along the west wall

For the solution to be a valid and consistent finite volume scheme, the global heat conservation criteria must be fulfilled to machine precision. The absolute GHC value is plotted as function of number of grid points for the two schemes in figure 3.6. It can be seen that the error is in the order of machine precision, and that the error increase potential with number of grid points. It should also be noted that the error is smaller for the uds-cds scheme, due to fewer operations being performed for that scheme.

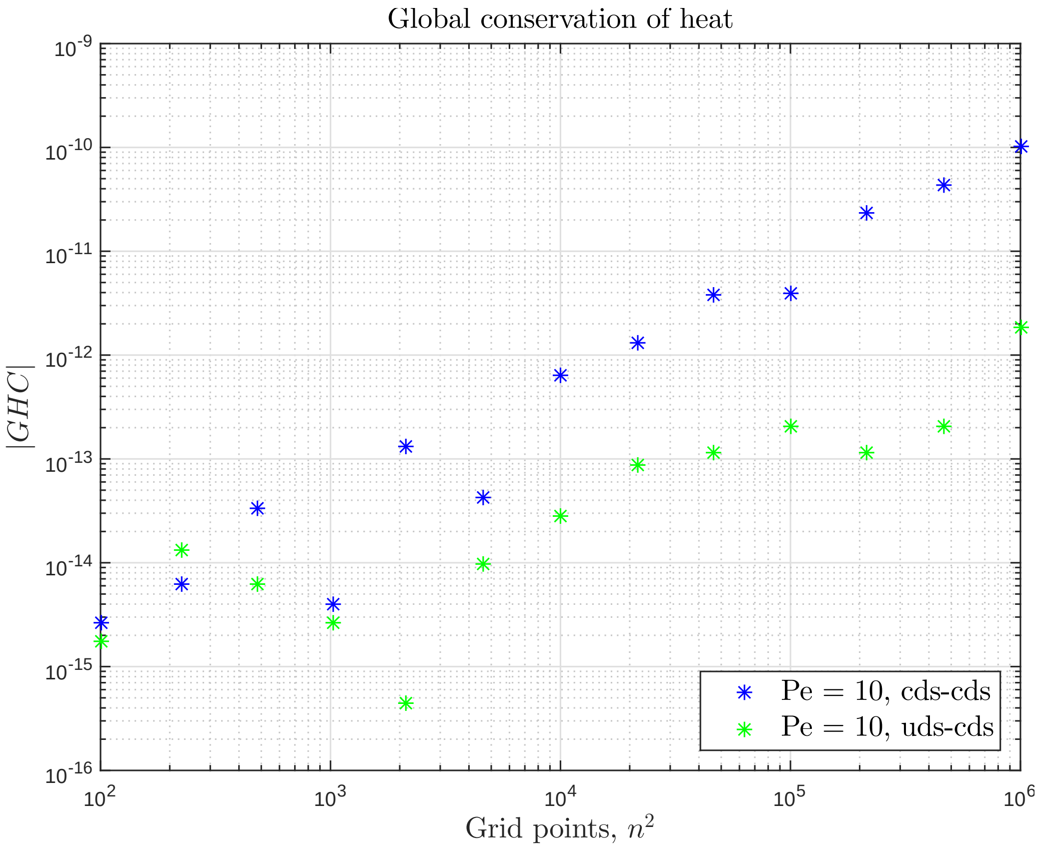


Figure 3.6: The absolute value of global heat conservation as function of number of grid points

The convergence of the diffusive heat flux along the west wall is also checked for both schemes. It can be seen from figure 3.7 that convergence towards the same value is observed for both cds-cds and uds-cds. It is obvious that the cds-cds converge faster. The order of convergence can be estimated by finding the difference between each value and the last value (most "converged" value). This is performed in figure 3.8.

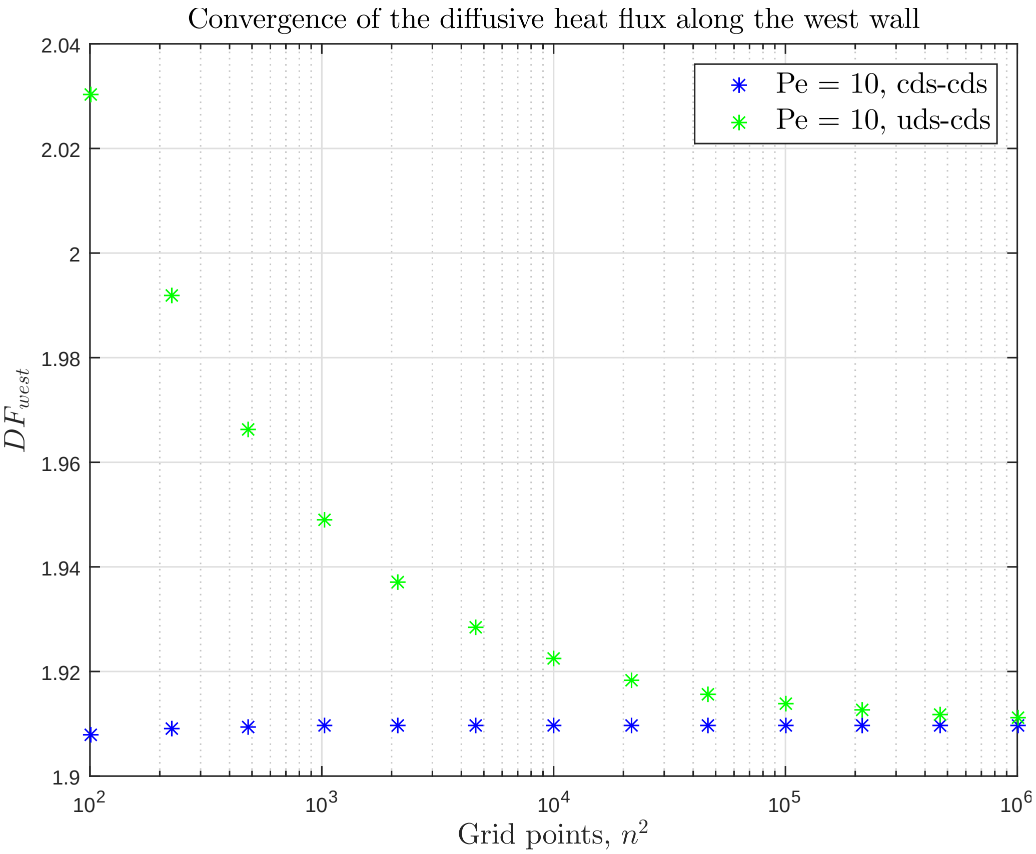


Figure 3.7: Convergence of the diffusive heat flux along the west wall is confirmed for both schemes

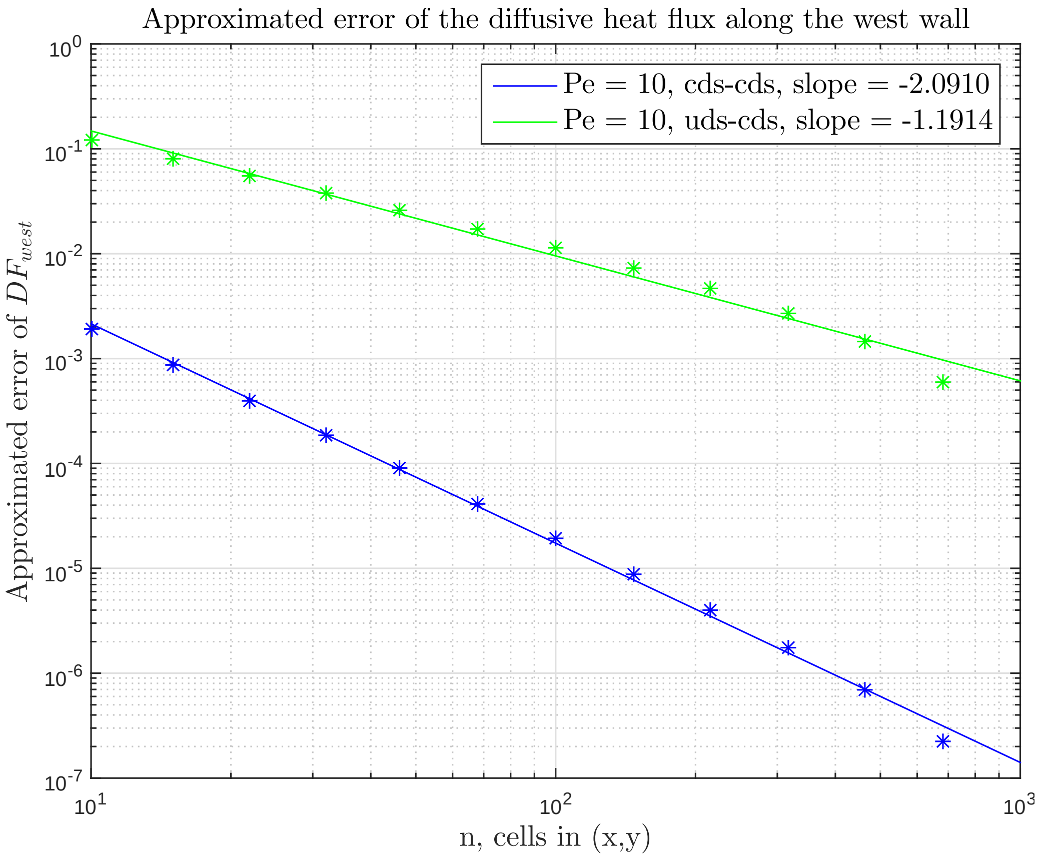


Figure 3.8: The slope of the convergence correlate with the order of the schemes

As described before, the values in figure 3.8 is found the following way:

$$(Approx.Error)_i = |DF_i - DF_n| \tag{3.5}$$

It can be seen that the cds-cds scheme convergence with a power of -2.09, which was expected due to this being a second order scheme. Therefore is is also seen that the first order uds-cds scheme converge with a power of -1.19.

3.2 Examination of the Peclet number's influence

So far in this project, only flows with a very low Peclet number have been considered. In most engineering applications, the Peclet would be much larger, so to examine the Peclet number's influence on the solution, the diffusive heat flux has been plotted for different Peclet numbers at different grid resolutions. It can be seen from figure 3.9 that with an increasing Peclet number, an increasing grid resolution is also needed to properly resolve the gradient at the west wall.

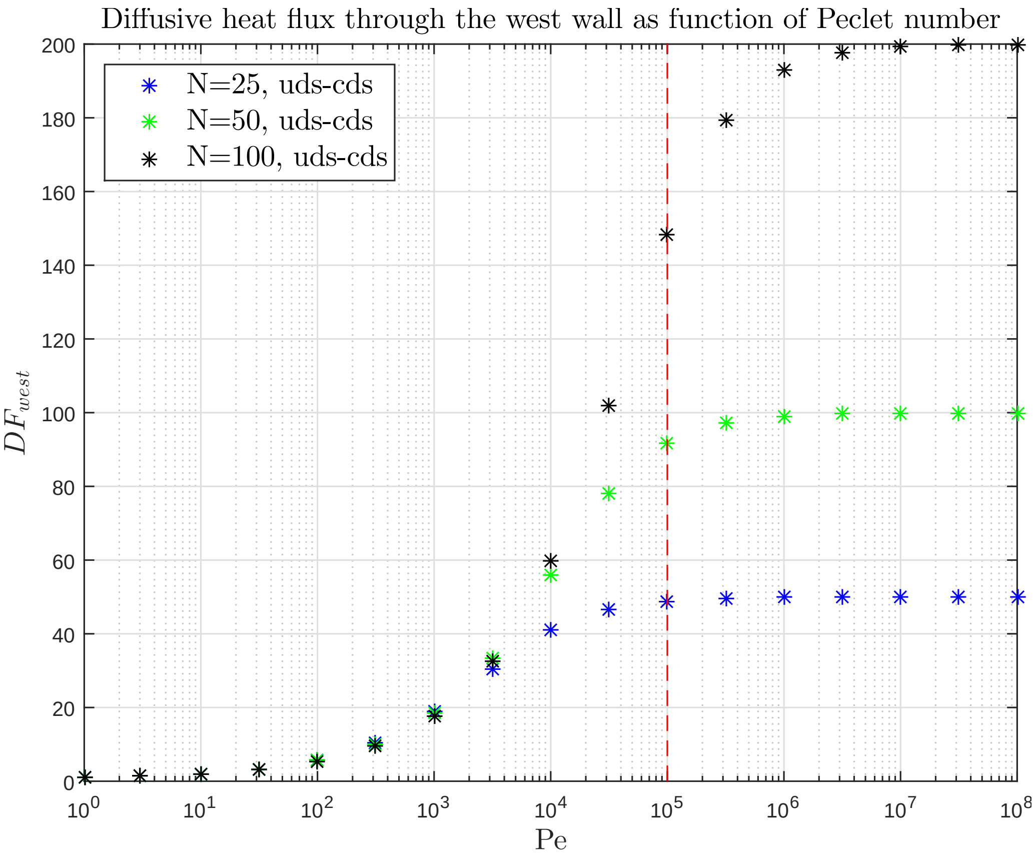
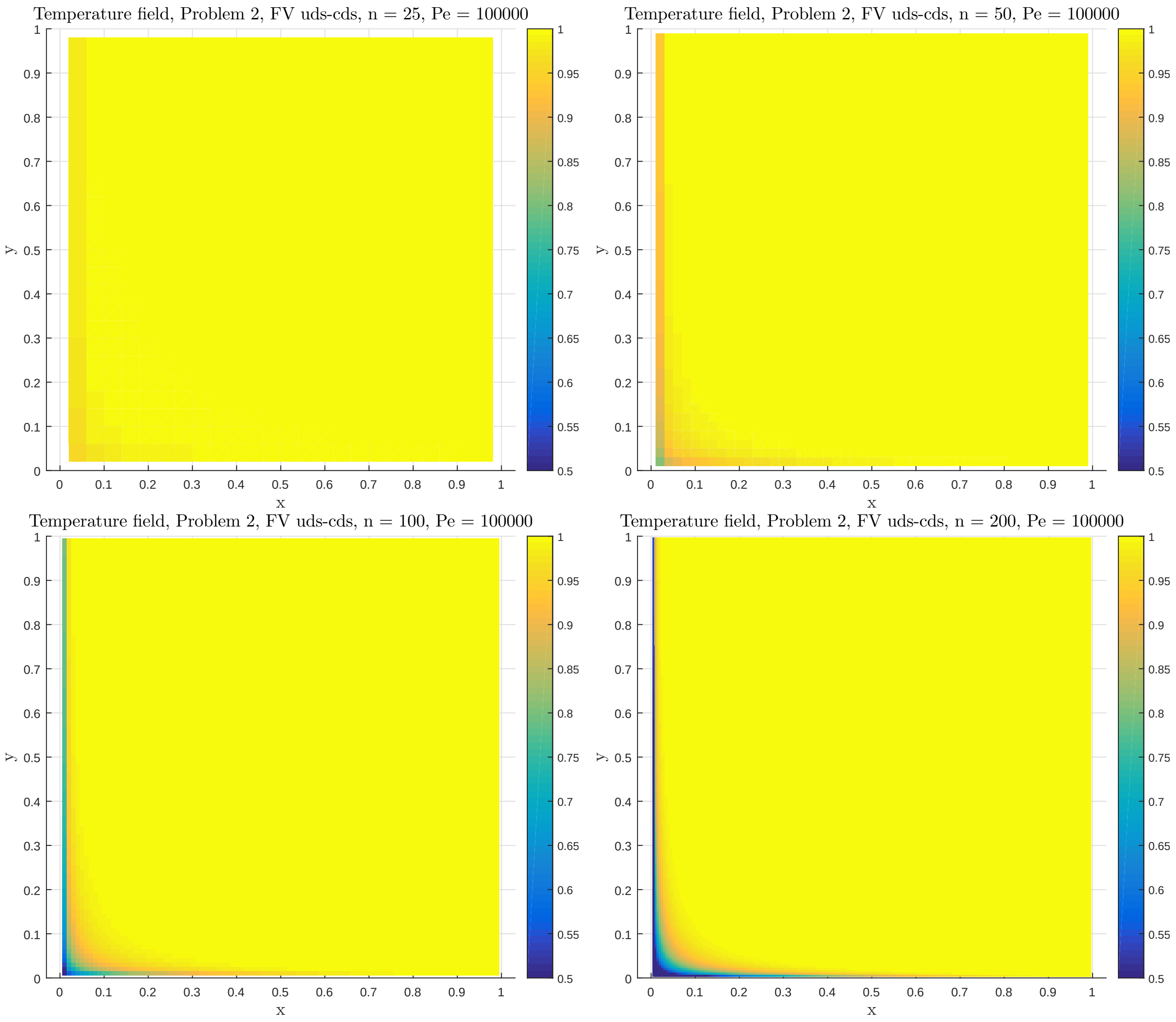


Figure 3.9: The red line illustrate which flow situations have been plotted in figure 3.10

In order to illustrate this, the temperature field for a corner flow with a peclet number of 10^5 is plotted on 4 different grids in figure 3.10. It is clearly seen that the temperature field along the western and southern wall is not properly resolved until n=100 is used, which also correlate with the tendency seen in figure 3.9. The uds-cds scheme is used in this comparison due to its ability to always produce physical solutions. This is seen in the two coarse grids in figure 3.10 where the shape of the temperature fields looks okay, it is just the magnitudes that are way off. If the same coarse grid was used with the cds-convection scheme, some non-physical fluctuation would have been observed.



Temperature field, Problem 2, FV uds-cds, n = 25, Pe = 100000

Temperature field, Problem 2, FV uds-cds, n = 50, Pe = 100000

Temperature field, Problem 2, FV uds-cds, n = 100, Pe = 100000

Temperature field, Problem 2, FV uds-cds, n = 200, Pe = 100000

Figure 3.10: Temperature fields for the uds-cds scheme at a high Peclet number for varying n

References

[1] Harry B. Bingham, Poul S. Larsen, V. Allan Barker, DTU Mekanik, *"Computational Fluid Dynamics"*, 2015