Institute of Scientific Computing
Technical University Braunschweig
Prof. Hermann G. Matthies, Ph.D.

The solution of the pressure equation is the most time-consuming part of the flow simulation. Thus we want to consider (and implement) in the following a very efficient method for the solution of linear equation systems: the multigrid method (MG).

1 Multigrid methods

The idea of the multigrid method is, to consider the problem on different planes. Thereby values must be carried over by a fine grid on the next coarser grid, this process is called restriction. On the other hand we must carry forward the won approximations on the coarse grid on the finer grid, then we speak of extension. On a solid grid smoothing steps are accomplished in addition, to eliminate the high-frequency parts of the error, i.e. we want to smooth the error.
For an overview we give a list of the operators and spaces, on whom they are defined:

Restriction operator: $R_h^{2h} : \Omega_h \longrightarrow \Omega_{2h}$,
Smoothing operator: $G_h : \Omega_h \longrightarrow \Omega_h$,
Prolongation operator: $P_{2h}^h : \Omega_{2h} \longrightarrow \Omega_h$.

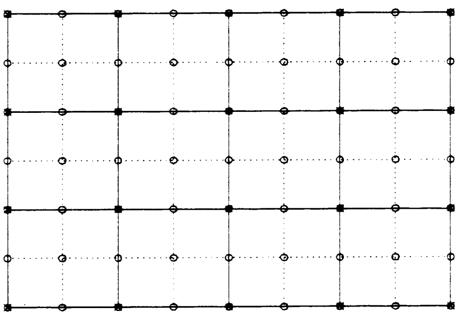

Figure 1: *Grid Ω_h and Ω_{2h}*

In our case we deal with a cell pattern and not with a grid, we are using the following restriction and prolongation (see figure 1):
For the restriction the values of four cells are averaged and used as a value for the coarser cell. Similarly with the prolongation the values are transferred from a large cell to the four daughter cells.

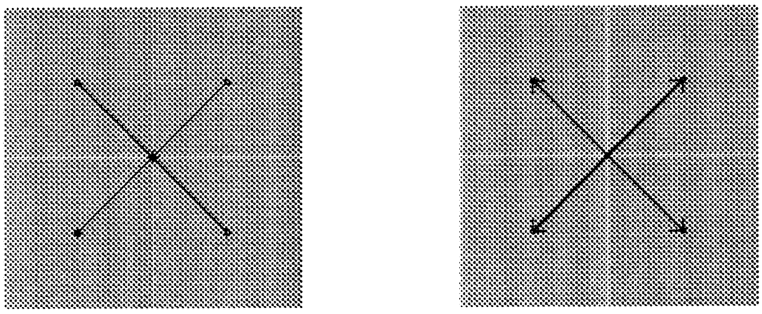

Figure 2: *Restiction and prolongation for the pressure cells*

The basic form of a multigrid step for the solution of $A^h u^h = f^h$ looks thereby as follows:
`Algorithm MG:` $u^h \longleftarrow MG(u^h, f^h)$

smooth $A^h u^h = f^h$ ν_1-times
$r^h = f^h - A^h u^h$
$f^{2h} = R_h^{2h} r^h$
solve $A^{2h} u^{2h} = f^{2h}$
$u^h = u^h + P_{2h}^h u^{2h}$
smooth $A^h u^h = f^h$ ν_2-times

But how should the approximation be won on the coarser grid. This remains still unclear in the algorithm above. One possibility is to solve the linear equation $A^{2h} u^{2h} = f^{2h}$ directly. That is however advisable only if one already concerned at a very coarse grid, i.e. the dimension of the set of equations which can be regarded is relatively small.
If one calls the program recursively instead, one obtain already a first version of the multigrid method, the so-called *V-Multigrid cycle*:
`Algorithm MV:` $u^h \longleftarrow MV(u^h, f^h)$

```
smooth A^h u^h = f^h  ν_1-times
if h < h_1 then

        r^h = f^h - A^h u^h
        f^{2h} = R_h^{2h} r^h
        u^{2h} = MV(0, f^{2h})
        u^h = u^h + P_{2h}^h u^{2h}

end
smooth A^h u^h = f^h  ν_2-times
```

One can illustrate the V-cycle as follows:

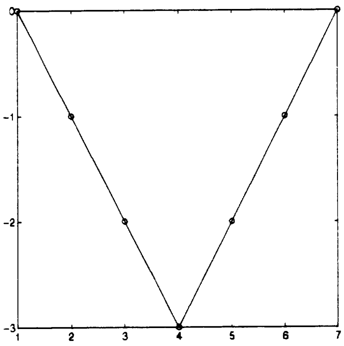

Figure 3: *V-Multigrid Cycle*

If one executes the recursive call twice consecutively, then this procedure is called *W-multigrid cycle*:

`Algorithm MW:` $u^h \longleftarrow MW(u^h, f^h)$

smooth $A^h u^h = f^h$ ν_1-times
`if` $h < h_1$ `then`

$$\begin{aligned} r^h &= f^h - A^h u^h \\ f^{2h} &= R_h^{2h} r^h \\ u^{2h} &= MW(0, f^{2h}) \\ u^{2h} &= MW(u^{2h}, f^{2h}) \\ u^h &= u^h + P_{2h}^h u^{2h} \end{aligned}$$

`end`
smooth $A^h u^h = f^h$ ν_2-times

The following visualization belongs to the W-cycle:

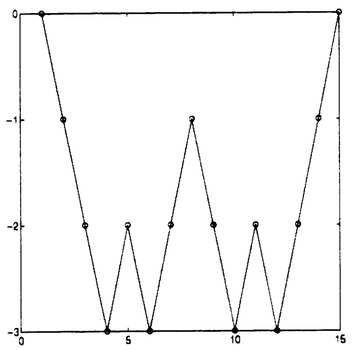

Figure 4: *W-Multigrid Zyklus*

If we start directly on the coarsest grid instead and execute the V-Multigrid cycles, then one speaks of the *Full Multigrid* or *Full Multigrid v-cycle.*
The algorithm looks as follows:
`Algorithm FMV:` $u^h \longleftarrow FMV(u^h, f^h)$

`if` $h < h_1$ `then`

$$\begin{aligned} r^h &= f^h - A^h u^h \\ f^{2h} &= R_h^{2h} r^h \\ u^{2h} &= FMV(0, f^{2h}) \\ u^h &= P_{2h}^h u^{2h} \end{aligned}$$

`end`
$u^h = MV(u^h, f^h)$

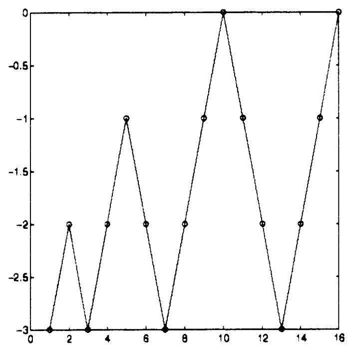

Figure 5: *Full Multigrid Zyklus*

The process of the Full Multigrid cycle can be visualized then with the following figure:
There are still numerous other versions of the multigrid method, which we do not want to enter here in detail (for more exact details look in the literature).
In all algorithms we can do without setting up the iteration stencils throughout, we just need the effect of the operators on the grid functions, in our case the pressure. Also we can execute the restriction and the prolongation, without setting up the matrices explicitly, which depend in the two-dimensional case crucially on the numbering of the unknown quantities.

Task 1 *Replace both functions* `SOR` *and* `PCG` *with*

- *Multigrid with V-cycle,*
- *Multigrid with W-cycle,*
- *and Full Multigrid.*

For the process of smoothing use the `SOR` *method.*
Do the following:

- *Convergence behavior of the methods (compare the results with* `SOR` *and* `PCG` *method).*
- *Time measurement for solving the set of equations,*
- *Variation of the parameters: relaxation parameter, number of smoothing steps, ...*

Apply the multigrid methods to some flow problems and compare the results with the other iterative methods!