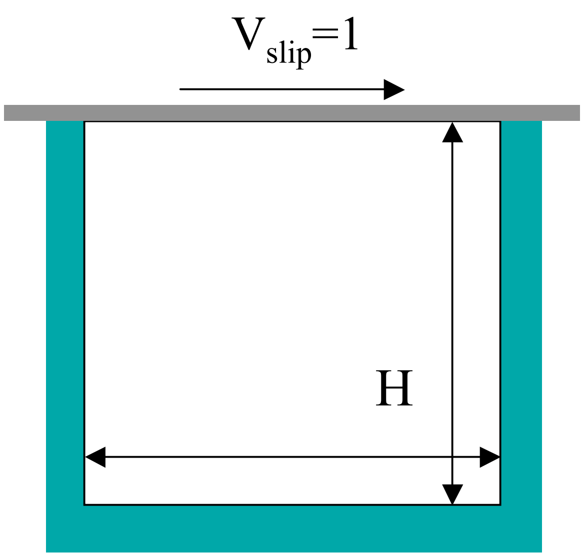
The Flow in Lid-Driven Cavity

Example – Driven cavity

Classical test-case for incompressible flow solvers



Problem set-up

Material Properties:
$\rho = 1 kg/m^3$
$\mu = 0.001 kg/ms$

Reynolds number:
$H = 1m$, $V_{slip} = 1 m/s$
$Re = \rho V_{slip} H/\mu = 1{,}000$

Boundary Conditions:
Slip wall ($u = V_{slip}$) on top
No-slip walls the others

Initial Conditions:
$u = v = p = 0$

Convergence Monitors:
Averaged pressure and friction on the no-slip walls

Solver Set-Up

Segregated Solver

Discretization:
2nd order upwind
SIMPLE

Multigrid
V-Cycle

An example of User Defined Programming

This is a simple workaround to compute grid-to-grid errors in Fluent

We will use the “interpolate” option in Fluent. This allows to write solutions computed on a given grid and read (interpolate) them on any other grid (finer/coarser).

Problem: only saves pressure/velocity/scalar data…

Workaround: save the solution (let’s say the x-velocity) in a “dummy” scalar field using a UDF.

An example of User Defined Programming

ADJUST routine to save the x-velocity in a scalar…

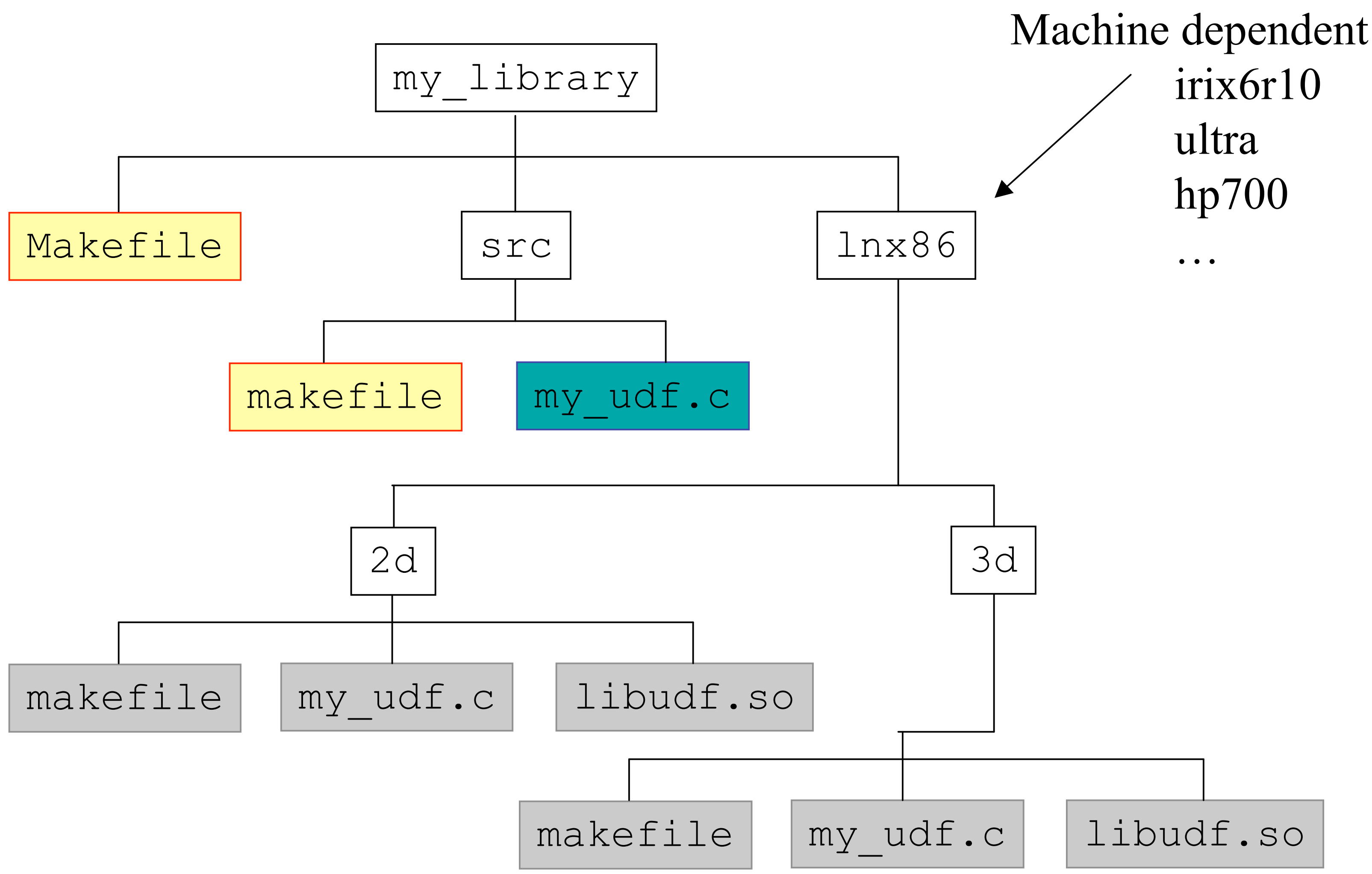
```
#include "udf.h"

enum
{ ONE, N_REQUIRED_UDS };

DEFINE_ADJUST(one_adjust, domain)
{
  Thread *t;
  cell_t c;
  thread_loop_c (t, domain)
        begin_c_loop(c,t)
          {

            C_UDSI(c,t,ONE)=C_U(c,t);
          }
        end_c_loop(c,t)
}
```

Directory tree for compiled UDFs



Makefiles for UDFs

In the directory

```
/usr/local/Fluent.Inc/fluent6.2.16/src
```

There are two files

`makefile2.udf` to be copied in the directory `my_library`
`makefile.udf` to be copied in the directory `my_library/src`

The first one does not require modifications.
In the second one two macros MUST be modified

```
SOURCE = my_udf.c
FLUENT_INC = /usr/local/Fluent.Inc
```