amar-codes/cflo/Source/main.cpp

```cpp
#include <iostream>

#include <AMReX.H>
#include <AMReX_BLProfiler.H>
#include <AMReX_ParallelDescriptor.H>

#include <cflo.H>
#include <userfuncs.H>

using namespace amrex;

int main(int argc, char* argv[])
{
    amrex::Initialize(argc, argv);

    // timer for profiling
    BL_PROFILE_VAR("main()", pmain);

    // wallclock time
    const Real strt_total = amrex::second();
    {
        // constructor - reads in parameters from inputs file
        //             - sizes multilevel arrays and data structures

        cflo_user_funcs::initialize_problem();

        cflo cflo_obj;

        // initialize AMR data
        cflo_obj.InitData();

        // advance solution to final time
        cflo_obj.Evolve();

        // wallclock time
        Real end_total = amrex::second() - strt_total;

        // print wallclock time
        ParallelDescriptor::ReduceRealMax(
            end_total, ParallelDescriptor::IOProcessorNumber());
        if (cflo_obj.Verbose()) {
            amrex::Print() << "\nTotal Time: " << end_total << '\n';
        }
    }

    // destroy timer for profiling
    BL_PROFILE_VAR_STOP(pmain);

    amrex::Finalize();
}
```

→ defined in amar-codes/Source/cflo.cpp

amar-codes/Source/cflo.cpp -> InitData()

```
cflo::~cflo() {}
// initializes multilevel data
void cflo::InitData()
{
    if (restart_chkfile == "") {
        // start simulation from the beginning
        const Real time = 0.0;
        InitFromScratch(time);
        AverageDown();

        if (chk_int > 0) {
            WriteCheckpointFile();
        }

    } else {
        // restart from a checkpoint
        ReadCheckpointFile();
    }

    if (plot_int > 0) {
        WritePlotFile();
    }
}
```

→ defined in amrex/Src/AmrCore/AMReX_AmrCore.cpp (2)

→ defined in cflo/Source/cflo_gridwork.cpp (1)

(1) AverageDown() calls amrex::average_down

↳ defined in amrex/Src/Base/AMReX_MultiFabUtil.cpp

~

- actual amrex averaging functions : amrex_avgdown defined in:

amrex/Src/Base/AMReX_MultiFabUtil_*D_C.H

where * = dim #

(2) InitFromScratch(time) calls MakeNewGrids(time) (only)

↳ defined in

amrex/Src/AmrCore/AMReX_AmrMesh.cpp

(2.1) MakeNewGrids(time) (virtual void amrex)

calls MakeNewLevelFromScratch(level, time, boxarray, dist.map)

- defined in cflo/Source/cflo_gridwork.cpp
- only called upon initialization