

DYNAMICO

Table of Content

DYNAMICO	1
1. Introduction	2
2. Getting DYNAMICO	2
3. Compiling DYNAMICO	2
Compiling with XIOS output	2
4. Running dynamico	3
Running dynamico on parallel computers	3
Use nudging with DYNAMICO	3

1. Introduction

The DYNAMICO project develops a new dynamical core for LMD-Z, the atmospheric general circulation model (GCM) part of IPSL-CM Earth System Model. The primary goal of DYNAMICO is to re-formulate in LMD-Z the horizontal advection and dynamics on a icosahedral grid, while preserving or improving their qualities with respect to accuracy, conservation laws and wave dispersion. In turn, a new grid refinement strategy is required. A broader goal is to revisit all fundamental features of the dynamical core, especially the shallow-atmosphere/traditional approximation, the vertical coordinate and the coupling with physics. Efficient implementation on present and future supercomputing architectures is also a key issue addressed by DYNAMICO.

2. Getting DYNAMICO

DYNAMICO is licensed under the [CeCILL](#) open source license. The latest version is accessible through svn :

```
svn co http://forge.ipsl.fr/dynamico/svn/codes/icosagcm/trunk
```

Registered IPSL forge users belonging to the DYNAMICO group can do :

```
svn co svn+ssh://mylogin@forge.ipsl.fr/ipsl/forge/projets/dynamico/svn/codes/icosagcm/trunk DYNAMICO
```

'mylogin' should be your forge login. svn will create the DYNAMICO directory and download the source code there. Source can also be browsed at <https://forge.ipsl.fr/dynamico/browser/codes/icosagcm/trunk>

3. Compiling DYNAMICO

DYNAMICO is written in Fortran 90 with some legacy code in Fortran 77. The build process is based on [FCM](#). DYNAMICO requires the NetCDF library, including the F90 modules. MPI is required for parallel execution but DYNAMICO can compile and run without MPI.

DYNAMICO depends on the NetCDF and BLAS libraries.

The compiling process is automated but some information is needed to guide it. This information is contained in text files in <https://forge.ipsl.fr/dynamico/browser/codes/icosagcm/trunk/arch> . Sample files corresponding to a few machines (Jean-Zay at IDRIS, Irène at TGCC) are present. Assuming you compile on Irène :

```
cd DYNAMICO
./make_icosg -arch X64_IRENE -parallel mpi -prod -job 8
```

will compile DYNAMICO. The make_icosg script accepts keyword-value pairs which drive its behaviour. The most important option is arch=ARCH (here ARCH=X64_IRENE). It directs make_icosg to use the information contained in :

- arch/arch-ARCH.env
- arch/arch-ARCH.fcm
- arch/arch-ARCH.path

The *.env file is a shell script that is executed by make_icosg. It sets up the environment for use by *.path. *.path defines paths to libraries and modules needed for compilation. *.fcm defines the commands used to compile, link, etc. as well as options to be passed to the compiler/linker .

The option "-job 8" is similar to "make -j 8" and compiles in parallel for speed. After a successful build the main executable is found in the bin/ directory.

Compiling with XIOS output

DYNAMICO can direct its output through XIOS, a parallel I/O library and server. See <https://forge.ipsl.fr/ioserver>. To enable XIOS output :

- get and compile XIOS in a separate directory
- set the variables XIOS_INCDIR , XIOS_LIBDIR and XIOS_LIB to appropriate values in your arch.path
- use the option "-with_xios" in your "make_icosg" command

Why use XIOS :

- without XIOS, each output field is written to a separate NetCDF file. Post-processing is required to group several fields together.

- with XIOS, several fields can be written to a few output files. This behavior is controlled by the input file `xios.xml` (required). See <https://forge.ipsl.fr/ioserver> for the syntax of this XML file.
- without XIOS, data to be written is communicated to the main MPI process, which writes to the NetCDF files while other MPI processes stay idle. This is not expected to scale to a large number of MPI processes.
- XIOS provides asynchronous, parallel I/O in order to scale to large MPI process counts.

4. Running dynamico

DYNAMICO can produce a rather large amount of output. It is therefore recommended to prepare a separate directory for each numerical experiment on a filesystem of adequate capacity. In this directory, copy the executable `icosa_gcm.exe`. You will find it in `DYNAMICO/bin/`, where `DYNAMICO` is the main directory containing the source code.

You will also need configuration files that define resolution, initial condition, etc. Sample files can be found in subdirectories of https://forge.ipsl.fr/dynamico/browser/codes/icosa_gcm/trunk/param_sets. Copy the `*.def` files from the desired sub-directory. There is typically `run.def` and `earth_const.def`. `run.def` is the main configuration file and includes `earth_const.def`

Currently DYNAMICO generates its own grid. It can run idealized test cases which define their own initial conditions. In that case no other input files are needed beyond `*.def` and, if using XIOS, `*.xml` files controlling XIOS behavior. It may also restart from a previous run, reading from a restart file.

Running dynamico on parallel computers

DYNAMICO can run in parallel by dividing the icosahedral mesh into tiles. There are at least 10 tiles corresponding to the 20 faces of the icosahedron joined in pairs to form rhombi. These 10 rhombi are further subdivided in `nsplit_i` x `nsplit_j` tiles. `nsplit_i` and `nsplit_j` are defined in `run.def`. `nsplit_X` needs not divide `nbp` exactly.

To run DYNAMICO on a parallel machine, you must first compile it with OpenMP and/or MPI. Then use `mpirun` or the equivalent command to run it and `OMP_NUM_THREADS` environment variable to fix the number of openMP threads. There must be less MPIxOpenMP processes than the `10 x nsplit_i x nsplit_j` tiles. There can be more tiles than processes, in which case each process will take care of several tiles.

Use nudging with DYNAMICO

Nudging is available in `IPSLCM7_work` configuration. To activate it you need to modify the parameter "ok_guide" in `dynamico.card` (not in `lmdz.card` - this one is only use in the case of the regular grid)

```
ok_guide=y
```

LibIGCM will **automatically** modify the parameter file `run_dynamico.def` to put

```
guided_type=nudging
```

User need to know which data will be used for the nudging. Only U & V, or / and T, or / and PS ? By default we proposed only nudging on U & V with ERA5 files on 0.25degree.

If you want to use another configuration you need to ask the person in charge of the configuration.

The next step is to indicate in `dynamico.card` which files will be used for the nudging. As explain before this example is for nudging on U & V.

```
[BoundaryFiles]
List=( $CCSTOREDIR/../../subipsl/subipsl/ERA5/NETCDF/GLOBAL_025/4xdaily/AN_PL/${year}/u.${year}${month}.aphe5.GLOBAL_025.nc
      ($CCSTOREDIR/../../subipsl/subipsl/ERA5/NETCDF/GLOBAL_025/4xdaily/AN_PL/${year}/v.${year}${month}.aphe5.GLOBAL_025.nc
ListNonDel=()
```

If your login is not included in the group `subipsl` (check with command `groups`), ask to Anne Cozic to add it.

Dynamico is using XIOS to read input files. For this we use an xml file with all informations needed by the model: `PARAM/nudging_dynamico.xml`. If it's not already by default in `dynamico.card` you need to add it to the parameters files section

```
[ParametersFiles]
List= \
```

```
(...)  
(${SUBMIT_DIR}/PARAM/nudging_dynamico.xml, .)
```

Currently (July 6, 2023) you need to modify the routine modeles/DYNAMICO/src/dissip/nudging.f90 and recompile. You need to comment these lines

```
IF (is_omp_master) CALL xios_set_file_attr("nudging_in",enabled=.TRUE.)  
IF (is_omp_master) CALL xios_set_file_attr("nudging_in0",enabled=.TRUE.)
```

Now you can run your simulation