How to use OpenIFS in nudged configuration

In this article we explain how to prepare and configure OpenIFS 43r3v1 for a nudged simulation. Therein the model needs to read meteorological parameters at the grid scale from pre-computed external forcing files. These forcing files have to be created prior to the nudged OpenIFS model experiment and this process is also described here.

> (i) Please note that nudging in OpenIFS is an experimental research tool and therefore may change between model versions.
>
> The description on this page refers to nudging in gridpoint space which also allows regional nudging. This differs from nudging in spectral space which constrains the model in the longer wave numbers on a global scale.
>
> For further assistance on nudging and configuring OpenIFS please post your question in the OpenIFS User Forums or alternatively email openifs-support@ecmwf.int

1. Newtonian Relaxation

OpenIFS uses initial and boundary conditions to calculate its own model dynamics, i.e. meteorological variables that are resolved on the grid scale. It is however possible to constrain the model dynamics with external data. Newtonian relaxation, sometimes referred to as "nudging", is a simple form of data assimilation which allows the user to constrain or "force" the model's meteorological fields with reanalysis data. This is sometimes referred to as running the model in "offline" mode. In nudged configuration the model's dynamics is continually nudged towards the meteorological reanalysis independent of the run length of the experiment.

This method relaxes the model state towards gridded re-analysis data (or towards output from another atmospheric model, or gridded observational data) by adding a non-physical relaxation term to the model equations (Jeuken et al., 1996). In the equation below X represents a prognostic model variable and F_{model} the model forcing which determines the evolution of X. The relaxation term $G\ (X_{obs} - X)$ includes the relaxation coefficient G (in s^{-1}) which determines the "tightness" of the nudging.

$$ \frac{\partial X}{\partial t} = F_{model}(X) + G(X_{obs} - X) $$

This method can be useful, for instance, in sensitivity studies which aim to isolate the model physics or chemistry while preventing feedbacks to the model dynamics. Another example for its use is to align a climate model simulation closer to historic meteorology for comparison with measurements.

2. Planning the Experiment

For the preparation of a nudged model experiment the following points should be considered:

2.1. Nudged variables

The current model version permits nudging of the following 9 prognostic variables that are stored in two external forcing files (one gridpoint file and one spectral file) for each nudging step:

- Vorticity,
- divergence,
- temperature,
- surface pressure,
- specific humidity,
- ice water contents,
- liquid water contents,
- cloud fraction,
- and stratospheric ozone.

The decision on which of these variables should be nudged will depend on the scientific objectives of the experiment. Often it is not necessary to nudge all the variables, however as a minimum we recommend to constrain vorticity, divergence and temperature.

In this context disk space usage will become a consideration as an increasing number of nudged variables will result in larger forcing files. At T255L60, for instance, forcing files that contain all 9 variables require 70 MB for each time step (47 MB for the grid point file and 23 MB for the spectral file). This amounts to approximately 8.5 GB for one month with 6-hourly nudging.

2.2. Relaxation coefficients

As a standard all relaxation coefficients are initially set to 0.5 which results in a relatively "tight" nudging to the external data. These values should be adjusted for each experiment according to its objectives. Frequently the best results are obtained when different relaxation coefficients are used, specific to each nudged variable. The external analysis data is updated every six hours and the model linearly interpolates in time between these data points. Too tight nudging can result in unrealistic behaviour in the freely calculated model variables.

2.3. Grid resolution

One of the strengths of OpenIFS is its ability to operate with a wide range of grid resolutions. The model is however not capable of spatial interpolation of the gridded forcing file data. Therefore the forcing files need to be prepared at the horizontal and vertical resolution of the model experiment. ERA-Interim re-analyses have a horizontal resolution of T255 (79 km globally) with 60 levels and ERA-5 re-analyses are at T639 (31 km globally) with 137 levels. The IFS experiment described below is able to generate forcing files at either of these two resolutions. In order to nudge the model when using other resolutions the forcing files need to be first interpolated to the model grid.

> ⓘ **Note:** At present we only generate forcing files at the horizontal resolutions T255L60 and T639L137.

2.4. Data storage

The spectral and gridpoint forcing files need to be accessible to the model at runtime, i.e they must not be read from an offline tape archive. Depending on the length of the model experiment (e.g. for a seasonal forecasts or in climate model configuration) it is conceivable that several months or even years of 6-hourly forcing files need to be prepared. This will have implications for the available disk space. At T255 with 60 levels one year of forcing files can require up to 100 GB disk space (depending on the number of variables the files contain). Once the forcing files have been created their long-term storage may need to be considered.

3. Preparation of Forcing Files

This section describes a possible way to generate forcing files for nudged experiments. It is only necessary to carry out these steps if no nudging files are available and need to be created.

In its default nudging configuration the model requires a spectral and a gridpoint forcing file every six hours, at 0000, 0600, 1200, and 1800 UTC.

The default naming convention for these files are `rlxmlshYYYYMMDDHHHH` for the spectral files, and `rlxmlggYYYYMMDDHHHH` for the gridpoint files. YYYY, MM, DD, HHHH represent the year, month, day and hour of the forcing data. The model will expect to find these files in the experiment directory unless the path variables `CLPSHRLX` and `CLPGGRLX` in the namelist are set.

> ⓘ The name strings of the forcing files are generated by the model in `src/ifs/utility/updrlxref.F90`

The following two sections describe a method to generate forcing files using an IFS experiment on the ECMWF HPCF and examples for two forcing files are available for download below.

3.1. Using IFS – Scripts Branch

This section describes how forcing files can be generated from ERA-Interim reanalyses (at T255L60) using an IFS CY43R3 experiment. For this to succeed you will need to be able to submit an IFS suite using prepIFS. You will further need to make a local copy of a scripts branch using IFS Git commands.

A local copy of scripts branch `damk_CY43R3_getrelax` is required. This branch copies the forcing files from the `/fws` working directory to `$SCRATCH/rlxml/`. At present the directory `rlxml` must exist in the user's scratch space with global writing permissions.

Tasks - Prepare scripts branch

1. Load ifs git tools: `module load ifs-git-tools`
2. Add repository of user *damk*: `git ifsremote damk`
3. Make a copy of the scripts branch: `git ifsbranch -B bb-damk/damk_CY43R3_getrelax -b mygetrelax`
4. In script file `getrelax` (line 183) assign your username to variable ME
5. Commit the changes in git and push the branch to your repository (git add, git commit, git push)
6. On the ECMWF HPCF (cca or ccb) create the destination directory for the forcing files: `mkdir -p -m 777 $SCRATCH/rlxml`

3.2. IFS Experiment

IFS experiment **h7z9** is set up to run in nudged configuration and the *inigroup* task family (shown in the XCdp workflow GUI) will retrieve ERA-Interim re-analyses from the MARS archive and generate the required forcing files. You will need to make a copy of this suite, modify it as described in this section, and submit it to the ECMWF HPCF.

ⓘ **Note:** There is no need to run the *model* task family. The suite can be aborted after *inigroup* has completed.

In prepIFS select the start date and run length in order to produce forcing files. This is done by adjusting variables INIBEGINDATE, INIENDDATE and INITIME such that all show the same start date (usually the first day of the month), e.g. 2013070100.

FCLENGTH should be set to 744 for one month or 1488 for two months of forcing files to be created.

Tasks - Prepare IFS experiment

1. Start prepIFS
2. In menu "File" click on "Add user" and include user *damk*
3. Copy experiment h7z9 from user *damk*
4. Set INIBEGINDATE and INIENDDATE to 2013070100 (or your chosen start date)
5. Set INIBY to zero
6. The INITIME list of individual initial dates should have just one entry, 2013070100
7. Set FCLENGTH to 744 to create forcing files for one calendar month
8. Replace the scripting branch with your own copy created in the previous section
9. Check and submit the suite to the HPCF

For every 6-hour period the IFS suite will create one gridpoint and one spectral forcing file from ERA-Interim analyses at T255L60 reduced linear Gaussian horizontal grid. It is essential that the forcing files have the same horizontal grid and vertical levels as the OpenIFS experiment.

3.3. Forcing File Examples

Examples for a spectral and a gridpoint forcing file are available for download from http://download.ecmwf.int/test-data/openifs/nudging/examples/ (no login required for access).

```
rlxmlgg201308251200    12-Sep-2019 10:00    48312000    [ md5: cc7428b4016637aee63277d14b6149e9 ]
rlxmlsh201308251200    12-Sep-2019 10:00    24109200    [ md5: 3722948745891d1de0ae5e5b8c1a7e33 ]
```
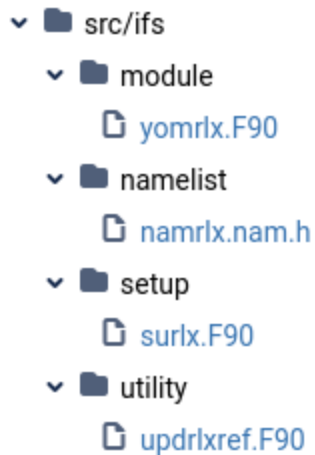
These two forcing files for 25-08-2013 12:00 were created using the IFS experiment described above. They are in GRIB format and can be examined using the command line tools available from the ecCodes library.

```
$ grib_ls rlxmlgg201308251200 | head
rlxmlgg201308251200
edition     centre      typeOfLevel level       dataDate    stepRange   dataType    shortName
packingType gridType
1           ecmf        hybrid      1           20130825    0           an          q
grid_simple reduced_gg
1           ecmf        hybrid      1           20130825    0           an          ciwc
grid_simple reduced_gg
1           ecmf        hybrid      1           20130825    0           an          clwc
grid_simple reduced_gg
1           ecmf        hybrid      1           20130825    0           an          cc
grid_simple reduced_gg
1           ecmf        hybrid      1           20130825    0           an          o3
grid_simple reduced_gg
1           ecmf        hybrid      2           20130825    0           an          q
grid_simple reduced_gg
1           ecmf        hybrid      2           20130825    0           an          ciwc
grid_simple reduced_gg
1           ecmf        hybrid      2           20130825    0           an          clwc
grid_simple reduced_gg

$ grib_ls rlxmlsh201308251200 | head
rlxmlsh201308251200
edition     centre      typeOfLevel level       dataDate    stepRange   dataType    shortName
packingType gridType
1           ecmf        hybrid      1           20130825    0           an          lnsp
spectral_complex  sh
1           ecmf        hybrid      1           20130825    0           an          vo
spectral_complex  sh
1           ecmf        hybrid      1           20130825    0           an          d
spectral_complex  sh
1           ecmf        hybrid      1           20130825    0           an          t
spectral_complex  sh
1           ecmf        hybrid      2           20130825    0           an          vo
spectral_complex  sh
1           ecmf        hybrid      2           20130825    0           an          d
spectral_complex  sh
1           ecmf        hybrid      2           20130825    0           an          t
spectral_complex  sh
1           ecmf        hybrid      3           20130825    0           an          vo
spectral_complex  sh
```

4. Relevant Source Code Files

In OpenIFS 43r3v1 the source code required for nudging is included by default. This section points out some of the relevant files within the source code, however in order to simply enable nudging this part is not essential to understand as the only changes that need to be applied are in the namelist file. The following part How to Enable Nudging describes the required namelist changes.

- ✓ 📁 src/ifs
  - ✓ 📁 module
    - 🗋 yomrlx.F90
  - ✓ 📁 namelist
    - 🗋 namrlx.nam.h
  - ✓ 📁 setup
    - 🗋 surlx.F90
  - ✓ 📁 utility
    - 🗋 updrlxref.F90

In order to control the model's nudging configuration the following files of the source code are relevant:

- `namrlx.nam.h` – the namelist block which appears in the fort.4 file. These variables allow to customise the model's nudging behaviour without recompiling the model source code.
- `yomrlx.F90` – in this module the variables in the namelist block are defined.
- `surlx.F90` – this is the setup routine for the namelist variables, which includes their initialisation with default values and consistency checks.
- `updrlxref.F90` – this subroutine updates the reference fields for the relaxation.

The nudging code in OpenIFS 43r3v1 differs from that in IFS CY43R3 insofar that the variables CLPSHRLX and CLPGGRLX have been added to the namelist. These two variables allow to specify the location of the directory with the forcing files on the file system. If these are not set then the model will expect to find the forcing files in the experiment directory.

The relaxation to the analyses involves code in further routines (e.g. `yomsrlx.F90`, `relaxgp.F90`, `stepo.F90`, `stepo_oops.F90`, `su0yomb.F90`, and `updtim.F90`) however this part does not need to be changed if the existing nudging capability of the model is to be used. The code in `relaxgp.F90` performs the actual relaxation in grid point space.

4.1. How to Enable Nudging

To enable nudging the namelist `NAMRLX` will need to be included in the namelist file `fort.4`. No further modifications should be necessary. The values below show one possible configuration reflecting standard settings. As described previously, the question of which variables should be relaxed to the external analysis, the model grid domain where relaxation is applied, and the tightness of the nudging, controlled by the relaxation coefficient, are very much dependent on the objective of the model experiment.

Default settings in the NAMRLX namelist

```
&NAMRLX
  LRLXG=true,    ! Master switch to enable nudging

  LRLXVO=true,   ! vorticity
  LRLXDI=true,   ! divergence
  LRLXTE=true,   ! temperature
  LRLXQ=true,    ! specific humidity
  LRLXQL=true,   ! liquid water
  LRLXQI=true,   ! ice water
  LRLXQC=true,   ! cloud fraction
  LRLXLP=true,   ! log surf pressure
  LRLXO3=true,   ! ozone

  NFRLXG=2,      ! number of analysis time steps in memory
  NFRLXU=6,      ! interval between two references fields (in hours) - this is equal to NFRHIS in nudging

  NRLXLMIN=1,    ! top model level for nudging
  NRLXLMAX=60,   ! bottom model level for nudging
  NRLXLMINU=-1,  ! uppermost model level for velocity variables (-1 sets to NRLXLMIN)
  NRLXLMAXU=-1,  ! lowermost model level for velocity variables (-1 sets to NRLXLMAX)

  XRLXVO=0.5,    ! relaxation coefficients
  XRLXDI=0.5,
  XRLXTE=0.5,
  XRLXQ=0.5,
  XRLXO3=0.5,
  XRLXLP=0.5,

  ALATRLX1=90,   ! latitude domain for nudging
  ALATRLX2=-90,
  ALONRLX1=0,    ! longitude domain for nudging
  ALONRLX2=360,

  AXRLX=-0.5,    ! smoothing parameter for lon, lat and in the vertical
  AYRLX=-0.5,
  AZRLX=1.0,

  CLPSHRLX='../rlxml/sh/',  ! relative path to spectral forcing files
  CLPGGRLX='../rlxml/gg/',  ! relative path to gridpoint forcing files
/
```

ⓘ **Note:** When using OpenIFS versions prior to CY43R3 the available namelist parameters can differ from the ones shown above. For instance, nudging of ozone is not available in CY40R1.

5. References

- Jeuken, A., et al. (1996) On the potential of assimilating meteorological analyses in a global climate model for the purposes of model validation, *J . Geophys. Res.*, **101**, 16,939–16,950. doi:10.1029/96JD01218.