OOPS Jb Changes

Mike Fisher

ECMWF

February 1, 2016

TYPE_JB_STRUCT

All the configuration data for Jb has been collected into a structure:

```
TYPE TYPE_JB_STRUCT
  TYPE (TYPE_JB_CONFIG)                          :: CONFIG
  TYPE (TYPE_JB_CONFIG2)                         :: CONFIG2
  TYPE (TYPE_JB_DATA)                            :: JB_DATA
  TYPE (TYPE_FCEMN_STRUCT)                       :: FCEMN
  TYPE (TYPE_VCORS_STRUCT), POINTER              :: VCORS(:)
  TYPE (TYPE_SPJB_VARS_INFO_STRUCT),POINTER      :: SPJB_VARS_INFO(:)
  TYPE (TYPE_WAVELETJB_CONFIG)                   :: WJBCONF
  TYPE (TYPE_WAVELETJB_DATA)                     :: WJBDATA
  TYPE (TYPE_WAVELETJB_VCOR_STRUCT),POINTER      :: WAVELET_VCORS(:,:) =>NULL()
  TYPE (TYPE_WAVELETJB_GRID_STRUCT),POINTER      :: GRID_DEFINITION(:) =>NULL()
  TYPE (TYPE_LAMWAVELETJB_CONFIG)                :: LAMWAVELET_CONFIG
  TYPE (TYPE_LAMWAVELETJB_INFO)                  :: LAMWAVELET_INFO
  TYPE (TYPE_LAMWAVELET_VCOR_STRUCT)             :: LAMWAVELET_VCOR
  TYPE (TYPE_LAMWAVELET_GRID_STRUCT)             :: LAMWAVELET_GRID
  TYPE (TYPE_JBCHVAR)                            :: JBCHVAR
END TYPE TYPE_JB_STRUCT
```

TYPE_JB_STRUCT

In OOPS, there may be more than one instance of Jb:

- Each inner-loop resolution will have its own Jb.
- The Jb code will be re-used for the model-error covariance matrix in weak-constraint 4dVar.

Some parts of the IFS code use a global structure configuration structure:

```
MODULE YOMJG
...
TYPE(TYPE_JB_STRUCT), POINTER :: JB_STRUCT => NULL()
...
END MODULE YOMJG
```

This global variable will disappear in a future cycle. Please do not use it in any new code!

Instead, pass configuration data by argument!

Background

The OOPS Jb code provides a covariance matrix to the C++ layer.

- Calculation of the background departure is done in the C++ layer and is not part of the covariance operator.

However, the background is required as a linearization state for the balance operators.

In IFS, the background handling was mixed up with the trajectory handling. A lot of the IO code and data structures were shared.

For OOPS, the background is passed to Jb by argument as a `FIELDS` type variable.

The global variables SP7A3 and SP7A2 have been removed.

Trajectory

The humidity (and ozone) change-of-variable requires a trajectory state for linearization.

Currently, this state uses the old global trajectory-handling code (`GET_TRAJ_GRID` etc.)

This will change soon to pass the trajectory into Jb by argument as a `FIELDS` type variable.

Example Code

```
SUBROUTINE CVAR3IN(YDGEOMETRY,YDGMV,YDGMV5,YDLAP,YDLEP,YDCV,YDSPEC,PGP3,PGP2, &
                 &   YD_BG,YD_TRAJEC,YD_JB_STRUCT)
...
TYPE(FIELDS),              INTENT(IN)    :: YD_BG
TYPE(TRAJ_TYPE),           INTENT(IN)    :: YD_TRAJEC
TYPE(TYPE_JB_STRUCT),      INTENT(IN)    :: YD_JB_STRUCT
...
CALL SQRTBIN(YDGEOMETRY,YDLAP,YDLEP,YDCV,YDSPEC,PGP3,PGP2,YD_JB_STRUCT)
...
    IF (YD_JB_STRUCT%CONFIG%LJB_NONLINEAR_BALANCE) THEN
      CALL BALNONLINTL(YDGEOMETRY,YDLAP,YDSPEC%VOR,YD_BG%YRSPEC%SP3D(:,:,IPTVOR),ZZP)
...
END SUBROUTINE CVAR3IN
```