DIGITAL FILTERING INITIALISATION IN THE CYCLE 44 OF ARPEGE/IFS.

YESSAD K. (METEO-FRANCE/CNRM/GMAP/ALGO)

March 6, 2017

Abstract:

This documentation describes the digital filtering initialisation (DFI) and its implementation in ARPEGE/IFS. Some theoretical aspects are recalled. The way of implementing the DFI in ARPEGE/IFS is then described, and an organigramme is provided. A namelist is provided.

Résumé:

Cette documentation décrit l'initialisation par filtres digitaux (DFI) et le code associé dans ARPEGE/IFS. On rappelle quelques notions théoriques. On décrit la façon dont les DFI ont été codés dans ARPEGE/IFS, avec présentation d'un organigramme. On fournit un exemple de namelist.

Contents

1 Introduction and purpose.

In the atmosphere there are "slow" evolving waves (Rossby waves) and "fast" evolving waves (for example gravity waves). The aim of the initialization is to control the amount of gravity waves by some filtering in the analysis. There are mainly two classes of algorithms to do that:

- The normal mode initialization (NMI), which is a spatial (vertical and horizontal) filter.
- The digital filter initialization (DFI), which is a temporal filter.

The DFI initialization can be implemented on all types of geometry, and is easier to use on stretched spherical geometry than the NMI initialization. The NMI scheme is no longer used and has been removed from the code in 2012.

We now focus on the digital filter initialization in this documentation. This documentation is an update of the documentation (IDDFI94) which was valid for cycle 11r2, but written in a more synthetic way.

2 Algorithm of digital filter initialization.

2.1 Basics about the algorithm.

The DFI initialization is a temporal filtering and requires at least a forward forecast and generally also at least a backward forecast in order to display a model state in a time span centred on t_0. If M_{span} is the number of integration timesteps of the time span, high frequency oscillations of X are removed by replacing X by a "good" temporal average of X:

$$\overline{X}(t_0) = \sum_{k=-M_{\text{span}}}^{k=M_{\text{span}}} h_k \tilde{X}(t_k) + C$$

where $t_k = t_0 + k\Delta t$. This temporal average is computed in spectral space.

- X is generally any state variable of the model (the DFI applies on all spectral prognostic variables).
- $t_k = t_0 + k\Delta t$, Δt is the timestep of the initialisation; $M_{\text{span}}\Delta t$ is the time span.
- h_k are the filtering coefficients, and there are several possible ways to compute them which will be detailed below.
- C is a correction term which can be zero or non-zero according to the way of applying the filtering.
- $\tilde{X}$ is a version of X which can be equal to X or slightly different from X according to the way of applying the filtering.

The DFI initialization can be not scale-dependent: in this case coefficients h_k do not depend on the wavenumbers. It can be scale-dependent (implemented in limited area models and for Dolph-Chebyshev and ideal low-pass filters only): in this case h_k depends on the wavenumbers (dependency on $\sqrt{\frac{n(n+1)}{a^2}}$ if spherical geometry, $\sqrt{\frac{m^2}{L_{\text{x}}^2} + \frac{n^2}{L_{\text{y}}^2}}$ if LAM model, where (m, n) are the zonal and meridian wavenumbers in a LAM model, $(L_{\text{x}}, L_{\text{y}})$ are the zonal and meridian domain lengths in a LAM model, n is the total wavenumber if spherical geometry).

2.2 Computation of temporal coefficients.

There are several ways to compute the coefficients h_k, which are controlled by some variables like **NTPDFI**.

2.2.1 Dolph-Chebyshev filter.

This is the last algorithm which has been coded, this algorithm is now used operationally. It is described for example in chapter 2 of (Lynch et. al, 1999). Expression of h_k is:

$$h_k = \frac{1}{2M_{\text{span}} + 1}\left[1 + 2r \sum_{j=1}^{j=M_{\text{span}}} \left(T_{2M_{\text{span}}}\left(x_0 \cos\left(\frac{\pi j}{2M_{\text{span}} + 1}\right)\right) \cos\left(\frac{2\pi jk}{2M_{\text{span}} + 1}\right)\right)\right]$$

where:

- $T_{2M_{\text{span}}}$ is the Chebyshev polynomial of order $2M_{\text{span}}$. Expression of the Chebyshev polynomials is:
  - $T_n(x) = \cos(n \arccos(x))$ if $\mid x \mid \leq 1$.
  - $T_n(x) = \cosh(n\ \text{arccosh}(x))$ if $\mid x \mid > 1$.
- r and x_0 are two parameters defined by:

  $$x_0 = \frac{1}{\cos((2\pi\Delta t)/(\tau_{\text{s}}))}$$

  where τ_{s} is a cut-off period, and:

  $$r = \frac{1}{\cosh(2M_{\text{span}}\text{arccosh}(x_0))}$$

When not combined with a simple filter (see below) this filter is never scale-dependent. It can also be combined with a simple filter (see below): scale-dependent or not scale-dependent versions.

2.2.2 Non recursive "ideal low-pass" filter (also called "simple" filter).

Expression of $h_{(k,m,n)}$ $(k \neq 0)$ is:

$$h_{(k,m,n)} = \frac{1}{H_{(m,n)}} f_k \frac{\sin(C_{(m,n)}\pi k/M_{\text{span}})}{\pi k}$$

Expression of h_0 is:

$$h_{(0,m,n)} = \frac{1}{H_{(m,n)}} C_{(m,n)} \frac{1}{M_{\text{span}}}$$

$H_{(m,n)}$ is a normalisation factor:

$$H_{(m,n)} = C_{(m,n)} \frac{1}{M_{\text{span}}} + \sum_{k=-M_{\text{span}}}^{k=-1} \left[f_k \frac{\sin(C_{(m,n)}\pi k/M_{\text{span}})}{\pi k} \right] + \sum_{k=1}^{k=M_{\text{span}}} \left[f_k \frac{\sin(C_{(m,n)}\pi k/M_{\text{span}})}{\pi k} \right]$$

Quantity H does a normalisation for h:

$$\sum_{k=-M_{\text{span}}}^{k=M_{\text{span}}} h_{(k,m,n)} = 1$$

Expression of f_k is:

- With the Lanczos window:

$$f_k = \frac{\sin(\pi k/(M_{\text{span}}+1))}{(\pi k/(M_{\text{span}}+1))}$$
$$f_0 = 1$$

- Without the Lanczos window:

$$f_k = 1$$

Expression of $C_{(m,n)}$ is, for limited area models:

$$C_{(m,n)} = \frac{M_{\text{span}}}{\pi} \min\left(\frac{\pi}{M_{\text{span}}} + 2R_{\text{dfis}}\pi\Delta t\sqrt{\frac{m^2}{L_{\text{x}}^2} + \frac{n^2}{L_{\text{y}}^2}}, \pi \right)$$

If $R_{\text{dfis}} = 0$, the filter is not scale-dependent and $C_{(m,n)} = 1$.
For spherical geometry, only $C_{(m,n)} = 1$ is implemented.

2.3 The different ways of filtering.

Once defined the coefficients h_k, calculation of $\overline{X}(t_0)$ requires at least a forward forecast and generally also at least a backward forecast in order to display a model state in a time span centred on t_0. But there are several possible algorithms and in some cases there are more than one backward forecast or more than one forward forecast. We can combine adiabatic forecasts (this is always the case for the backward forecasts) and diabatic forecasts. These different possibilities are controlled by the namelist variable **NEDFI**. In the adiabatic forecasts, the horizontal diffusion can be activated or not activated according to the value of variable **LADIFH**.

The main options used for operational applications are options 6 and 7:

* Option 6 (forward diabatic):

- Forward diabatic forecast with horizontal diffusion from $X(t_0)$, giving $X_+(t_0 + k\Delta t)$ for $0 \leq k \leq 2M_{\text{span}}$.
- $C = 0$.
- Non recursive filter: $\tilde{X}(t_0 + k\Delta t) = X_+(t_0 + (k + M_{\text{span}})\Delta t)$. The obtained filtered value $\overline{X}$ is centred on the instant $t_0 + M_{\text{span}}\Delta t$.
- There is also a recursive option in order to provide a filtered value valid for an instant between t_0 and $t_0 + M_{\text{span}}\Delta t$. In this case $\tilde{X} = X$ and we use h_k for k between 0 and $2M_{\text{span}}$ (instead of k between $-M_{\text{span}}$ and M_{span}).

This option is well adapted to the cases where the initialisation is applied to a file which is in the middle of the integration: screening, update of the trajectory, J_{c} constraint in a minimization.

$*$ Option 7 (diabatic without correction, non recursive):

- Backward adiabatic forecast from $X(t_0)$, giving $X(t_0+k\Delta t)$ for $-M_{\text{span}} \leq k \leq -1$, and a filtered value $X_{\text{ad,f}}(t_0 - 0.5M_{\text{span}}\Delta t)$ at the instant $t_0 - 0.5M_{\text{span}}\Delta t$.
- Forward diabatic forecast with horizontal diffusion from $X_{\text{ad,f}}(t_0 - 0.5M_{\text{span}}\Delta t)$, giving $X_+(t_0+k\Delta t)$ for $-0.5M_{\text{span}} \leq k \leq 0.5M_{\text{span}}$, and a filtered value $\overline{X}(t_0)$.
- $\tilde{X} = X_+$; $C = 0$.

The code contains additional options numbered from 1 to 5:

$*$ Option 1 (adiabatic, non recursive):

- Backward adiabatic forecast from $X(t_0)$, giving $X(t_0+k\Delta t)$ for $-M_{\text{span}} \leq k \leq -1$.
- Forward adiabatic forecast from $X(t_0)$, giving $X(t_0+k\Delta t)$ for $1 \leq k \leq M_{\text{span}}$.
- $\tilde{X} = X$; $C = 0$.

$*$ Option 2 (diabatic without correction, non recursive): This is a simplified and less optimised variant of option 7.

- Backward adiabatic forecast from $X(t_0)$, giving $X(t_0+k\Delta t)$ for $-M_{\text{span}} \leq k \leq -1$.
- Forward diabatic forecast with horizontal diffusion from $X(t_0 - M_{\text{span}}\Delta t)$, giving $X_+(t_0+k\Delta t)$ for $-M_{\text{span}} \leq k \leq M_{\text{span}}$. Let us remark that $X_+(t_0) \neq X(t_0)$.
- $\tilde{X} = X_+$; $C = 0$.

$*$ Option 3 (diabatic with correction, non recursive): The same as option 2, but $C = X(t_0) - X_+(t_0)$.

$*$ Option 4 (adiabatic + diabatic without correction, non recursive): This is a combination of the options 1 and 3.

- First apply the option 1 to compute a first filtered value, denoted by $X_{\text{f}}(t_0)$.
- Apply then the option 3 from $X_{\text{f}}(t_0)$; that provides: $X_+(t_0+k\Delta t)$ for $-M_{\text{span}} \leq k \leq M_{\text{span}}$. Let us remark that $X_+(t_0) \neq X(t_0)$ and $X_+(t_0) \neq X_{\text{f}}(t_0)$.
- $\tilde{X} = X_+$; $C = X_{\text{f}}(t_0) - X_+(t_0)$.

$*$ Option 5 (modified diabatic, non recursive): This is a complex combination of option 1 and option 2.

- First apply the option 1 from $X(t_0)$ to compute a first filtered value, denoted by $X_{\text{ad,f}}(t_0)$.
- Apply then the option 2 from $X(t_0)$ to compute a second filtered value, denoted by $X_{\text{f}}^{'}(t_0)$.
- During the two precedent operations, we successively do a backward adiabatic forecast and a forward diabatic forecast, which provides $X_+^{'}(t_0)$.
- Apply the option 1 from $X_+^{'}(t_0)$ to compute a third filtered value, denoted by $X_{\text{f}}^{''}(t_0)$.
- The final filtered field writes:

$$\overline{X}(t_0) = X_{\text{f}}^{'}(t_0) + X_{\text{ad,f}}(t_0) - X_{\text{f}}^{''}(t_0)$$

2.4 Digital filter initialization constraint in a variational assimilation.

A digital initialisation filtering constraint can be introduced in the 4DVAR assimilation via a term J_{c} (in the function J to be minimized) and J_{c} is proportional to $(\delta\mathbf{X} - \overline{(\delta\mathbf{X})})^T(\delta\mathbf{X} - \overline{(\delta\mathbf{X})})$ where $\overline{(\delta\mathbf{X})}$ is a temporal average of $\delta\mathbf{X}$ done with the same coefficients as in the digital filtering initialisation ($\mathbf{X}$ is the vector containing the state variables X). During the minimization, J_{c} diminishes, so $\mathbf{X}$ gets closer to its filtered version $\overline{(\delta\mathbf{X})}$. Remark that the option 6 presented above is particularly adapted to compute J_{c} in a 4DVAR assimilation, there is no need to do a backward integration in this case.

3 Organigramme.

3.1 Setup.

```
SU0YOMA -> SUINI
SU0YOMB ->
 * SUDFI
 * SUFW -> DOLFIL and SMPFIL
```

3.2 Direct code under CNT3.

```
CNT3 -> DFI ->
 * DFI2 ->
   0/ Initialize local variables and arrays:
   - ZEROACU
   - COPSP and COPGFL
   1/ Save constants to local ones before overwriting:
   2/ Launching different DFI sessions:
   - DFI3 -> (see organigramme below)
   - COPSP and COPGFL
   - CORSP and CORGFL
   - ZEROACU
   3/ Reset original constants and surface fields:
   - REAST
   - SUGRIDF -> (organigramme not detailed)
   - UPDCAL
   4/ Compute final spectral fields:
   - DIFSP
   - SUSPEC -> (organigramme not detailed)
   - COPSP and COPGFL
   - CORSP and CORGFL
 * DFI2MOD ->
   0/ Initialize local variables and arrays:
   - ZEROACU
   - COPSP
   1/ Save constants to local ones before overwriting:
   2 to 7/ Launching different DFI sessions:
   - DFI3 -> (see organigramme below)
   - COPSP
   - CORSP
   - ZEROACU
   8/ Correction:
   - CORSP
   - DIFSP
   9/ Reset original constants and surface fields:
   - REAST
   - SUGRIDF -> (organigramme not detailed)
```

3.3 Direct code under DFI3.

```
DFI3 ->
 1/ Set up for different DFI sessions
    - REAST
 2/ Temporal loop:
    2.1  Filter accumulation if necessary
    - DIGFIL
    - DIGP
    2.2  Coupling (if LAM model)
    - ELSRW -> (organigramme not detailed)
    2.3  Current value of the time step length
    - UPDTIM -> (organigramme not detailed)
    2.4  Reset semi-implicit solver (SI) and horizontal diffusion (HDIR)
    - SUNHSI -> (SI, organigramme not detailed)
    - SUHEG -> (SI, organigramme not detailed)
    - SUHDU -> (HDIR, organigramme not detailed)
    2.5  Time step
    - CHKOBTIM -> (Screening, organigramme not detailed)
    - STEPO -> (Time step management, organigramme not detailed)
    2.6  Printings
    - USER_CLOCK
    - DATE_AND_TIME
 3/ Reset YOMCT3:
    - OBSV -> (Screening, organigramme not detailed)
    - UPDOBS -> (Screening, organigramme not detailed)
 4/ Reset GFLGP to middle window values:
    - DIGP
```

3.4 Action of each routine.

∗ Setup routines:

- **SUINI**: set-up of control variables for initialisation.
- **SUDFI**: set-up of control variables and constants for DFI.
- **SUFW**: main routine for calculation of weights h_k.
- **DOLFIL**: calculation of weights h_k for the Dolph-Chebyshev filter.
- **SMPFIL**: calculation of weights h_k for the non recursive “ideal low-pass” filter.

∗ Direct code under CNT3:

- **CNT3**: controls integration job at level 3.
- **DFI**: controls integration for digital filter initialization at highest level.
- **DFI2**: controls integration for digital filter initialization at level 2.
- **DFI2MOD**: controls modified integration for digital filter initialization at level 2.
- **COPSP**: copy spectral arrays PSPB.. in PSPA...
- **COPGFL**: copy GFL arrays in array GFL.
- **CORSP**: copy the spectral difference “PSPB.. - PSPA..” into PSPA...
- **CORGFL**: copy the GFL difference in array GFL.
- **DIFSP**: copy the spectral difference “PSPB.. - PSPC..” into PSPA...
- **REAST**: adjusts astronomical variables (for example the date) before each DFI session.
- **SUSPEC**: initialize the spectral fields of the model (generally by reading a file).
- **SUGRIDF**: initialize the grid-point fields of the model (generally by reading a file).
- **UPDCAL**: update calendar.
- **ZEROACU**: sets back accumulating (spectral) arrays to zero.

∗ Direct code under DFI3:

- **DFI3**: controls integration for digital filter initialization at lowest level (plays the same role as **CNT4** for a non-DFI direct integration).
- **REAST**: see above.
- **DIGFIL**: computation of digitally filtered initial fields within time loop: step-by-step summing up accumulated DFI vectors.
- **DIGP**: memory transfers between GFL and a buffer to save information.
- **ELSRW**: control of time events for large scale fields (LAM model only).
- **UPDTIM**: update time of the model.
- **SUNHSI**: setup for the semi-implicit scheme in the non-hydrostatic model.
- **SUHEG**: setup for the semi-implicit scheme in the hydrostatic model: specific features for variable-mesh.
- **SUHDU**: setup for the horizontal diffusion scheme: specific features for variable-mesh.
- **CHKOBTIM**: controls if an observation time slot is available (3D/4D VAR).
- **STEPO**: controls a timestep of the model integration.
- **OBSV**: vertical part of the observation operator.
- **UPDOBS**: update observations.

Remarks:

- For more details about the semi-implicit scheme, see documentation (IDSI).
- For more details about the horizontal diffusion scheme, see documentation (IDDH).
- For more details about the 3DVAR/4DVAR scheme and the observation interpolator, see for example documentation (IDVAR).
- For more details about **STEPO**, see for example documentations (IDEUL) for Eulerian advection and (IDSL) for semi-Lagrangian advection. Note that for the backward integrations, there is no physics.

4 List of useful variables.

Only a useful subset of variables are described for the following list of modules and namelists.

4.1 YOMDFI and NAMDFI.

```
HWOPF  : Filtering weights "h(k)".
HWOPFA : Filtering weights "h(k)" (adiabatic).
HWF0   : Filtering weights for the mean of the fields.
HWFA0  : Filtering weights for the mean of the fields (adiabatic).
NTPDFI : Type of filter:
          * Non-recursive filters:
            1: Ideal low-pass.
            2: Ideal low-pass with Lanczos window.
            4 or 5: Dolph-Chebyshev.
         Default value is 4.
NEDFI  : Filtering scheme:
         0: Computing digital filter weights.
         1: Adiabatic digital filter initialisation.
         2: Diabatic digital filter initialisation without correction.
         3: Diabatic digital filter initialisation with correction.
         4: Adiabatic digital filter initialisation followed by diabatic one.
         5: Modified diabatic digital filter initialisation.
         6: Forward diabatic digital filter (finalisation).
         7: Backward+forward digital filter initialisation.
         Default value is 7 for direct integrations, 6 for use in a minimization via term "Jc".
NSTDFI : Number of DFI timesteps in one direction for foreward integrations
         (half-number of timesteps if NEDFI=6 or 7).
NSTDFIA: cf. NSTDFI for backward integrations.
NOFFSETDFI : offset before starting Jc-DFI diagnostics.
TAUS   : Period of stop-band edge (in seconds) for Dolph-Chebyshev filters ("tau_s").
TAUC   : Cut-off period (in seconds).
RTDFI  : Timestep for DFI (in seconds) for foreward integrations.
         Default value is equal to TSTEP.
RTDFIA : Timestep for DFI (in seconds) for backward integrations.
         Default value is equal to TSTEP.
RDFIS  : velocity below which there is no filtering in SSDFI (variable R_dfis).
LADIFH : .T./.F.: horizontal diffusion/no horizontal diffusion in adiabatic integrations.
         Default value is .T.
```

Some of them are in the namelist **NAMDFI**:

4.2 YOMINI and NAMINI.

```
LDFI   : .F.: No initialisation.
         .T.: Digital filter initialisation.
LBIAS  : .T.: Computing initialisation increment.
LINCR  : .T.: Incremental initialisation.
LINITER: .T. during initialisation, .F. elsewhere.
LSCRINI: .T. if screening prepared in digital filter finalisation.
```

The following variables are in the namelist **NAMINI**: **LDFI**, **LBIAS**, **LINCR**.

4.3 YOMJCDFI.

```
RACCSPA3,RACCSPA2,RACCSPA1: accumulation arrays of spectral variables used in
                            the DFI (resp. for 3D, 2D, 1D variables).
QNORM1,QNORM2,QNORM3      : definition of the norms used for Jc.
LDIVONLY                  : .T.: only the divergence is penalized.
RSUMJCDFI                 : arrays containing the contribution of a given
                            processor to the Jc-DFI cost function (summed in EVCOST).
NSUBDFI                   : Used in Jc-DFI for long windows in conf 131: cf. N_COUPLED_WINDOWS
                            in this case (number of coupled (sub)windows), 1 otherwise.
DFICHECK                  : ??? (comment not provided in the code)
                            Used in Jc-DFI for long windows.
```

The following variable is in the namelist **NAMDFI**: **LDIVONLY**.

5 An example of namelist for ARPEGE.

We can start from the ARPEGE namelist provided in documentation (IDSL). To switch on DFI initialisation, some information must be added in NAMDFI and NAMINI, for example:

```
&NAMDFI
  NEDFI=7,
  NSTDFI=18,
  TAUS=10800.,
/

&NAMINI
  LBIAS=.TRUE.,
  LINCR=.FALSE.,
  LDFI=.TRUE.,
/
```

Some explanation about the DFI initialisation used:

- **LDFI**=.TRUE. in **NAMINI**: activation of a DFI initialisation.
- **NEDFI**=7 in **NAMDFI**: backward+forward digital filter initialisation.
- **TAUS**=10800. in **NAMDFI**: the period of stop-band edge is 3 hours.
- **NSTDFI**=18 in **NAMDFI**: the timespan is **NSTDFI**$*$**RTDFI**=10800 s (3 hours).

The consequence is that the forward and backward integrations have a range of 3 hours.

6 References and internal notes.

6.1 Publications.

- Lynch, P., D. Giard and V. Ivanovici, 1997: Improving the efficiency of a digital filtering scheme for diabatic initialization. *Mon. Wea. Rev.*, **125**, 1976-1982.
- Lynch, P., R. McGrath and A. McDonald, 1999: Digital filter initialization for HIRLAM. *HIRLAM Technical Report Nr***42**, 22pp.
- Termonia, P., 2008: Scale-selective digital-filtering initialization. *Mon. Wea. Rev.*, **136**, 5246-5255.

6.2 Some internal notes and other ARPEGE notes.

- (TDECDAS) 2016: IFS technical documentation (CY41R2). Part II: data assimilation. Available at “https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation”.
- (TDECTEC) 2016: IFS technical documentation (CY41R2). Part VI: technical and computational procedures. Available at “https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation”.
- (IDBAS) Yessad, K., 2017: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 44 of ARPEGE/IFS (internal note).
- (IDDFI94) Giard, D., 1994: L’initialisation par filtre digital dans ARPEGE. Internal documentation in French valid for cycle 11r2, 22pp.
- (IDDFI01) Giard, D., 2001: Digital filtering and data assimilation. Internal documentation, 22pp, available on the internet server “http://www.umr-cnrm.fr/alatnet/gourdon/cours/DG/Plan.html”.
- (IDDH) Yessad, K., 2017: Horizontal diffusion in the cycle 44 of ARPEGE/IFS (internal note).
- (IDDM) Yessad, K., 2017: Distributed memory features in the cycle 44 of ARPEGE/IFS (internal note).
- (IDEUL) Yessad, K., 2017: Integration of the model equations, and Eulerian dynamics, in the cycle 44 of ARPEGE/IFS (internal note).
- (IDMINI) Yessad, K., 2017: Minimizations in the cycle 44 of ARPEGE/IFS (internal note).
- (IDSI) Yessad, K., 2017: Semi-implicit spectral computations in the cycle 44 of ARPEGE/IFS (internal note).
- (IDSL) Yessad, K., 2017: Semi-Lagrangian computations in the cycle 44 of ARPEGE/IFS (internal note).
- (IDVAR) Fischer, C., and L. Berre, 2007: The variational computations inside ARPEGE/ALADIN: cycle CY32. Internal note, 77pp. Available on the intranet server “http://www.umr-cnrm.fr/gmapdoc/”.