Synthesis document about a consistent set of interfacing proposals

J.-F. Geleyn, B. Catry, B. Hansen-Sass, P. Termonia and M. Tudor
(incorporating suggestions by Y. Bouteloup, G. Hello, M. Hamrud and M. Janousek)

Version 2 (11/03/05)

1 Foreword

Given the current evolution in NWP to run large software ensembles at different resolutions with physics developed for different resolutions, a reflection on a framework (interface) allowing easy exchange of physical packages (or part of it) and possible easy comparison of those packages seems appropriate. The joint ALADIN-2/HIRLAM Training Course and Working Group meeting on Physics-Dynamics Interface (TCWGPDI thereafter) held in Prague (22-26/11/04) tasked the ALADIN-2 coordinator and an ad-hoc team he selected with the writing of the present technical document. Its aim is to synthesise in a consistent ensemble various ideas and proposals that came out of the working group parts of the TCWGPDI or that were proposed before and mentioned in the presentations at TCWGPDI. Furthermore it outlines the consequences on existing code parts of the touched projects, should these proposals be implemented.

2 Introduction

The ideas developed in this document are attempting to meet the needs of a kind of wish list. Trying to logically transform those into a set of identified goals leads to four points.

- (**1**) If one wants to really have the possibility of exchanges/comparisons of part/whole of different physical packages inside the wide ensemble that will encompass AROME, ALARO, ARPEGE (in its various declinations), the HIRLAM planned evolution and (may be) IFS, a minimum set of strict rules is necessary.
- (**2**) In order to keep the door open to future interesting evolutions, these rules should be thought for a maximum of generality with respect to existing high-level choices and a minimum of impact on existing low-level codes in the said ensemble (or at least the parts of it that will join the rationalisation effort). High-level means in principle above the interface (i.e. set-up, dynamics and data-flow) while low-level means in principle below the interface (i.e. physics).
- (**3**) Given the already very limited adjustment space that the previous double constraint leaves on any meaningful choice, what follows is strictly targeted to the sole current common characteristics of the dynamics inside the above-mentioned ensemble (p-type vertical coordinate with the use of hydrostatic pressure for a non-hydrostatic extension, semi-implicit (SI) and semi-Lagrangian time-stepping, spectral). It also applies for the time being only to purely 1-D parameterisation computations that may be invoked for a geographically arbitrary set of points defined at high-level (NPROMA-type strategy). In other words, **one is not aiming here at universal 'plug-compatibility'**.
- (**4**) The only high-level general options that will be considered at the design stage (other might be added later on if they fit in the already chosen framework -and not the reverse-) are the following:
  - (**a**) physics called before or after dynamics (anyhow always before the Helmholtz solver in one given SI step);

- (**b**) conservation (δ_m=0) or variation (δ_m=1) of the atmospheric mass with water fluxes at the surface;
- (**c**) parallel physics or sequential one (the 'sequence' may include dynamical tendencies in the case of physics after dynamics; conversely one is strictly limiting the physical tendencies' influence on the dynamics to the consequences of the choice δ_m=1);
- (**d**) projection of the diabatic heat source only on T (anyhow the obligatory choice in the hydrostatic case) or on the couple [T,p].

These options are not meant to be compulsory ones in all their declinations. Simply internal model choices should avoid mixed choices, especially in the second and fourth case. This set of options does not encompass the position of application of physical increments along the semi-Lagrangian trajectory. This is not because this position should be unique, but because it will be assumed to always follow the rule of consistency between time and space choices along the said trajectory. Hence the position is only dictated by the time, or in practice by the explicitness or implicitness of the used scheme. Similarly, the position inside a Predictor/Corrector (P/C) time scheme should in principle be deductible from other choices, if one wants to avoid expensive double cost solutions. The consequences of these choices on the handling and high-level control of the non-prognostic information circulating from one time-step to the next or from predictor-step to corrector-step has not yet been analysed but it is thought that this does not interfere with all the following. A further version of this document will obviously deal with this topic, in an unchanged spirit.

Compromises to the above wish list will have to be accepted. We propose the following five central choices, presently presented as coding rules that should make possible a future realisation of the wish list.

- Rule A: The control of the various options (the above ones and those particular to any physics but interacting with (**a**), (**b**), (**c**) or (**d**) in their application) shall be done either at a low-level in the code and with a very specific meaning for the logical switches or at high-level with the aim to control the set-up sequence and/or the calling of blocks of physics routines or the I/Os related to their calls. This rule (**A**) allows starting the implementation 'small step by small step' (basically from the existing) while leaving the door open to future consistent evolutions of all kinds. Low-level routines are meant as those seen by the appropriate level of interfacing (hence buffer routines may be implemented to comply with the rule without touching much the computational part, even if more integration would be desirable) and the prognostic quantities are those having a time evolution equation in budget form (or depend only linearly on them). For more details see paragraph 3.1.

- Rule B: The link between individual tendencies and the projection of a summed forcing on the dynamical terms **will obey a single set of governing diabatic equations** which should be derived both from basic physical rules for going from the Navier-Stokes scales to the one of parameterisation choices and from common rules for simplifying hypotheses, independently of the dynamical core they should be applied at the time of discretisation. It will be supposed that whenever a low-level routine does not automatically assumes in its algorithms the full application of these equations, the above-mentioned projection and the link with DDH-type diagnostics (and other diagnostics if needed) will be corrected (and this correction documented) at an intermediate interfacing level, chosen on a case to case basis (either within the tendency-fluxes conversions -preferred AROME solution- or at the output level of parameterisation routines -preferred HIRLAM solution-). The correction may first be purely formal but it is then recommended to either make it unnecessary or to make it evolve towards a 'physical' version. This rule (**B**) is meant to ensure consistency while allowing the above-mentioned minimum start. In

the specific case of AROME/ALARO this implies to fully take into account the transport of enthalpy by precipitations (which is not currently the case). For more details see paragraph 3.2.

- Rules C: Diagnostics development (aiming at comparisons of different physical packages). We propose to asses this wish in a DDH-type framework. A negotiated two-level (mandatory and optional) classification of the atmospheric terms to be treated in the DDH-type framework will serve as guideline for determining which variables of the low-level routines are necessary in the input/output flow related to them (the mandatory level is by definition constraining). It will of course be sought that the classification and the governing equations are internally consistent. All this does not directly apply to the surface, which is anyhow supposed to be externalised in other modes, be it with an explicit or implicit algorithm. This however assumes that the externalised surface blocks return to the atmospheric part all the necessary information for the computation of the boundary condition closing the set of governing diabatic equations for all mandatory terms. The rules (**C**) of this paragraph try to solve (on the basis of the existing IFS/ARPEGE well stabilised practice) the dilemma of diagnostics being either meaningless or so model-dependent that they can be neither standardised nor inter-compared. In the specific case of AROME/ALARO depending completely or partly on Meso-NH physics, the fact to use a DDH-type framework implies to be able to provide four time integrals of moisture rate of changes (corresponding to condensation, freezing, evaporation and sublimation) as well as two more similar integrals related to auto-conversion. These terms are not existing currently. For more details see paragraph 3.3.

- Rule D: In order to simplify to the utmost the application of the three preceding rules, low-level routines could communicate on the basis of three 'status' for each relevant 'prognostic' or 'diagnostically-equivalent' variable: **reference**, **initial** and **final**. The first two statuses are input ones (and their potential distinction corresponds to the use of past information in the budget computations in the case of a sequential option) and the third status is an output one. A logical switch could also be controlling whether or not the 'reference-type' input was used in the routine. The reference status is in principle appropriate only for prognostic variables but its formal inclusion for the 'diagnostically equivalent' type of variables may prepare some yet unforeseen situation. This rule (**D**) would aim at going out of the flux vs. tendencies dilemma while allowing flexibility in the upper structure at quasi-unchanged interfaces for the low-level routines. For instance, the degree of implicitness of the internal computations would not need to be known by the interfacing system. The full use of rule D's potential might require additional work inside some low-level routines (so that they may be called transparently in all circumstances), but, when keeping at first a clone of the present situation, this can easily be by-passed through ignoring the 'reference input'. At that stage of the specification, it has not yet been analysed whether the 'authorised temporary use' of by-passing techniques around the strict application of rule B will influence the practical use of rule D. In any case, for more details about the latter in its present state, see paragraph 3.4.

- Rules E: The proposed application of the above architecture assumes that high-level control routines know about the type of 'mandatory' terms that every low-level routine will handle. This requires having a set-up system (to be completely defined only after the other choices have been fully validated) that attributes name of routines (and probably also positioning of their calls and parallel/sequential character of it, for cross-checking purposes first) contributing to a 'mandatory tendency' into a table indexed on this tendency. In principle, in a second stage, it should be possible to control, only from namelist parameters, the position and nature of the calls, provided the routines are designed to be called in an absolute sense, without implicit reference to the type of time-stepping they are fitted in, with the notable exception of the use or not of the 'reference status'. Of course this particular use can be neutralised at the level

of the call by passing identical reference values and initial values in the input stream. All the above does not mean that the whole physics should be called in one block or in one single mode (between sequential and parallel) but that a given routine should be called only once inside a given dynamical time-step (a P/C sequence is a single time-step in this definition). If multiple calls are absolutely necessary for scientific reasons, the introduction of an attribute system controlling the number and nature of calls will become necessary. This last set of rules (**E**) is meant to favour flexibility and generality which are two of the pillars for the fulfilment of our above-mentioned goal number one. Even if nothing is yet chosen for the concrete coding application of these rules, for more details about them, see paragraph 3.5.

The ensuing topical part of the document will be split between further explaining the 'rules' and looking at their consequences from the point of view of the 'users' (i.e. those people developing or maintaining the low-level routines). Nothing advanced will however be said about the work of people in charge of the architectural part of the work, should these ideas be concretised. It is indeed assumed that the few people entrusted with this work would get a free-hand for their detailed choices, provided they strictly fulfil all conditions related to the present document and its negotiated upgrades on the basis of acceptable users' constraints. **The readers which will be either already satisfied with the above explanations or only interested in practical consequences of the proposed solutions may well jump over the next '*further precisions*' section and go directly to '*consequences for low-level routines*'.**

3 Further precisions

3.1 Rule A

3.1.1 Existing code structures

The rules A until E should in the first place allow accommodating the known code structures.

For the models that are currently under interest, the existing time-step organisations resemble either the one of ARPEGE/ALADIN or that of the ECMWF model (Wedi). The AROME prototype adopted the structure of ALADIN/ARPEGE (with the exception that it is not parallel but sequential). Although the details differ, the time-step of the HIRLAM model follows the same structure as that of the ECMWF model. The physics computation of the HIRLAM model is indeed also sequential. The approach in this document is to propose rules for creating interfaces that accommodate these structures and leaves as much space open as possible for future developments.

For the sake of the discussion, let us denote the time-step computation as,

$$\frac{DF}{Dt} + L = R + \Phi\,, \tag{1}$$

for a field F and where L represents the linearised dynamics, R the non-linear residual and Φ the physics. In two-time level semi-implicit time-stepping this becomes:

$$\frac{F_A^+ - F_D^0}{\Delta t} + \frac{1}{2}\left(L_A^+ + L_D^0\right) = R + \Phi\,. \tag{2}$$

The two basic approaches to the time-step organisation are given as

1. in ARPEGE/ALADIN (which is also representative for the AROME prototype): the time-step is organised as described in table 1 [the notation follows the one of Eq. (2)].

	computation	result
1	inv. FFT, inv. Legendre transformation	$F(t)$
2	call physics (`APLPAR`)	Φ
3	update tendencies	$F_A^* = F(t) + \Delta t \Phi$
4	compute departure(, middle) point (**D**, **M**)	
5	interpolate to **D**(, **M**)	F_D^*
6	explicit part dynamics	F^{exp}
7	FFT, Legendre transformation	
8	Helmholtz, Horizontal diffusion	F_A^+

Table 1: Organisation of the time-step in ARPEGE/ALADIN

Note from this table that:

- the physics is computed before the dynamics,
- the physics is coupled at the departure point of the SL trajectory,
- the physics is computed in a parallel manner (but sequential in the AROME prototype)

2. The second time-step organisation is the one of ECMWF (and of the HIRLAM model): the code is organised as described in the table 2,

	computation	result
1	inv. FFT, inv. Legendre transformation	$F(t)$
2	lin. terms, non-lin. $R(t)$ and $R(t-\Delta t)$	L_A^0, R_A^-, R_A^0
3	compute departure point (**D**)	
4	interpolate to **D**	L^0, $R^0 = (2R - R^-)$
5	adiabatic explicit tendencies at arrival point (**A**)	$\tilde{D}$
6	interpolate diab. tendencies of rad., conv. and cl. at t to **D**	Φ^0
7	tendencies of parameterized processes	$\Phi^+(F(t), \tilde{D}, fractional)$
8	add tendencies of adiabatic and diabatic processes	$F_D^0 - \frac{1}{2}L^0 + \Delta t(R^{\frac{1}{2}} + \Phi^{\frac{1}{2}})$
9	FFT, Legendre transformation	
10	Helmholtz, Horizontal diffusion	F^+

Table 2: Organisation of the time-step in the ECMWF model. The notation $R^{\frac{1}{2}} \equiv \frac{1}{2}\left(R_D^0 + R_A^+\right)$ and $\Phi^{\frac{1}{2}} \equiv \frac{1}{2}\left(\Phi_D^0 + \Phi_A^+\right)$ is used.

where,

$$\tilde{D} = \frac{F_A^{exp} - F_A^0}{\Delta t}\,, \tag{3}$$

is the tendency of the explicit dynamics, but entirely taken in the arrival point of the semi-Lagrangian trajectory. Please note in table 2 that in point 6 the physics is kept from the previous time-step and then interpolated.

So in the case of the ECMWF:

- the physics is computed after the explicit part of the dynamics, but before the Helmholtz solver
- the physics is coupled at the departure point
- the physics is computed in a sequential manner

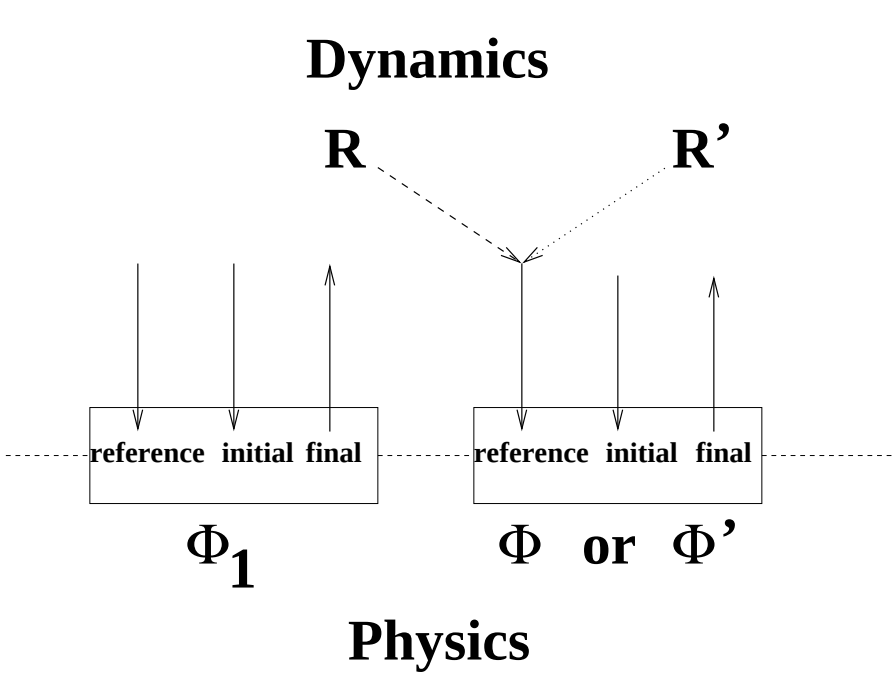


Figure 1: Hypothetical illustration of the use of the reference when replacing one sequential physics parameterization Φ by another Φ', where both use some output of physics computed earlier in the time-step; for instance suppose Φ needs the output of Φ_1, and Φ' needs some different output, say Φ_0 (not shown). This can be organised at high-level by passing a different reference $R'[\Phi_0]$ instead of $R[\Phi_1]$. In this way, the change to be made to the routines Φ and Φ' can be reduced to a strict minimum.

3.1.2 Further consistent evolutions

The choice between different time-step organisations should be justified by criteria such as stability and accuracy. In Staniforth *et al.*, the possibility of computing half of the physics before and half of the physics after the dynamics, in their so-called symmetrised split implicit scheme, has also been studied, and it was shown to be more accurate than non-symmetric schemes. However for this the physics has to be computed twice. Operationally this is not advisable. The ECMWF scheme attempts to get higher order accuracy by using the physics computed from the time-step before. However, this is the physics that was computed before the Helmholtz solver of the previous time-step, representing a compromise of the ideal. Specifically, the physics that was computed in point 6 in the table is averaged with the one from step 7, whereas to strictly apply what is in Staniforth *et al.*, $F(t)$ should be used.

The rules proposed in this document were invented to stay as general as possible and not to exclude any interesting future ideas to improve the organisation of the time-step in order to approach the "ideal" more closely. Moreover, we want to leave open the possibility to run the model in research mode with two calls of the physics per time-step for scientific tests, be it only as a reference for testing new, not yet proposed, time-step organisations. Another possibility would be to thoroughly test sequential physics in the ARPEGE/ALADIN setup, i.e. before the dynamics. The reason could for instance be that we want to import sequential HIRLAM physics as easily as meso-NH physics in ALARO where the organisation of the time-step is currently copied from ARPEGE/ALADIN. However, the type of the "sequentialness" might differ between the two solutions but the communication with the interface should be the same. Passing the specifically needed data through the reference status, whereas the physics parameterization decides how to use it, can solve this (rule D). The organisation of the time-step and possible consistency checks happen at high level and not at low level. This

is illustrated in Figure 1. Suppose we want to replace physics Φ by physics Φ'. The interface stays the same for both. If Φ' needs different input then a different reference is provided by the dynamics through the use of high-level switches.

In fact, each existing routine should communicate to the same interface, irrespective of its sequential or parallel nature. If one wishes a system where it is possible to swap one physics parameterization for another (for instance Meso-NH for HIRLAM or vice versa), both physics-parameterization codes should have the same type of input and output (i.e. accept the reference, initial, final state, and a logical tag in exactly the same way) to guarantee that they can be plugged in the same way in the interface. So, for such advanced experiments, some modification of the physics routines is unavoidable. The aim of rule A and rule D is to restrict any changes of the physics-parameterization code to a strict minimum. On the other hand, existing code should be slightly extended in the sense of rule D to make the two above-mentioned experiments possible, but only at a later stage than the first implementation of the interface.

So rule A is invented to implement the existing schemes and allow small-step-by-small-step changes to gradually eliminate theoretically non-ideal solutions (in the sense of solutions offered in simplified models such as in Staniforth *et al.*), with a minimum of modifications to the low-level physics routines.

3.2 Rule B

Rule B states that all low-level routines obey a single set of governing diabatic equations and that any inconsistency with this set of equations has to be corrected at the lower level. This single set of equations will include the thermodynamic equation in its most general mass-weighted form and a possibility to project heat on both temperature and pressure changes. The origin of the development was the AROME-equation paper by Stein *et al.* and the study of it by one of us (MT) for compatibility with the current ARPEGE/ALADIN set of equations.

We consider a micro-physical scheme with dry air (q_a), water vapor (q_v), liquid water (q_l), rain water (q_r), ice (q_i) and snow (q_s). Hail and graupel are not considered here as they are thermodynamically the same as snow and can be incorporated into it for our purposes. Liquid and ice water will remain in the particle and only rain and snow will precipitate. This precipitation is represented by the barycentric mass flux $P_l + P_i = \rho_r w_r + \rho_s w_s$, with w_r and w_s the respective barycentric vertical velocity of the rain water and of the snow. Next to this 'real' flux we also have the following pseudo fluxes: P_l' representing the integral of the transfer between vapor and liquid water due to condensation/evaporation; P_l'' representing the integral of the transfer between liquid and rainwater due to auto-conversion; P_l''' representing the integral of the transfer between rainwater and the water vapor due to evaporation of the falling liquid precipitation. P_i' representing the integral of the transfer between vapor and ice due to freezing/sublimation; P_i'' representing the integral of the transfer between ice and snow due to auto-conversion; P_i''' representing the integral of the transfer between snow and the water vapor due to sublimation of the falling solid precipitation. Apart from the auto-conversions all phase-changes are considered to go through the vapor phase.

At the surface we consider an evaporation flux E, a liquid precipitation flux R and a solid precipitation flux S (all positive downwards). We therefore have the following advective and diffusive fluxes (with $\delta_m = 0, 1$ a switch to preserve or vary the atmospheric total mass):

	$\delta_m = 0$		$\delta_m = 1$	
	advective	diffusive	advective	diffusive
q_a	0	$-g(E+R+S)$	$g(E+R+S)q_a$	$-g(E+R+S)q_a$
q_v	0	gE	$g(E+R+S)q_v$	$gE - g(E+R+S)q_v$
q_l	0	0	$g(E+R+S)q_l$	$-g(E+R+S)q_l$
q_r	0	gR	$g(E+R+S)q_r$	$gR - g(E+R+S)q_r$
q_i	0	0	$g(E+R+S)q_i$	$-g(E+R+S)q_i$
q_s	0	gS	$g(E+R+S)q_s$	$gS - g(E+R+S)q_s$

The surface pressure tendency equation becomes

$$\frac{\partial \pi_s}{\partial t} = -\int_0^1 \nabla.\left(\vec{v}\frac{\partial p}{\partial \eta}\right) d\eta - \delta_m g(E+R+S), \tag{4}$$

while the expressions for the vertical velocities become independent of the precipitation fluxes. The model coordinate-related one can be written as

$$\dot{\eta}\frac{\partial p}{\partial \eta} = B(\eta)\int_0^1 \nabla.\left(\vec{v}\frac{\partial p}{\partial \eta}\right) d\eta - \int_0^\eta \nabla.\left(\vec{v}\frac{\partial p}{\partial \eta}\right) d\eta + \delta_m g B(\eta)(E+R+S) \tag{5}$$

with $B(\eta)$ the classical function defining the vertical coordinate. This can be rewritten as

$$\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)^{\delta_m=1} = \left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)^{\delta_m=0} + \delta_m g B(\eta)(E+R+S). \tag{6}$$

The pressure-related vertical velocity becomes

$$\omega = \vec{v}.\nabla p - \int_0^\eta \nabla.\left(\vec{v}\frac{\partial p}{\partial \eta}\right) d\eta. \tag{7}$$

The conservation of the different mass-species can be written as:

$$\frac{dq_v}{dt} = -g\frac{\partial}{\partial p}(P_l' + P_i') + g\frac{\partial}{\partial p}(P_l''' + P_i''') + \delta_m g\frac{\partial}{\partial p}\left(q_v\frac{P_l+P_i}{1-q_r-q_s}\right) - g\frac{\partial J_{q_v}}{\partial p} \tag{8}$$

$$\frac{dq_l}{dt} = +g\frac{\partial P_l'}{\partial p} - g\frac{\partial P_l''}{\partial p} + \delta_m g\frac{\partial}{\partial p}\left(q_l\frac{P_l+P_i}{1-q_r-q_s}\right) - g\frac{\partial J_{q_l}}{\partial p} \tag{9}$$

$$\frac{dq_r}{dt} = +g\frac{\partial P_l''}{\partial p} - g\frac{\partial P_l'''}{\partial p} - g\frac{\partial P_l}{\partial p} \tag{10}$$

$$\frac{dq_i}{dt} = +g\frac{\partial P_i'}{\partial p} - g\frac{\partial P_i''}{\partial p} + \delta_m g\frac{\partial}{\partial p}\left(q_i\frac{P_l+P_i}{1-q_r-q_s}\right) - g\frac{\partial J_{q_i}}{\partial p} \tag{11}$$

$$\frac{dq_s}{dt} = +g\frac{\partial P_i''}{\partial p} - g\frac{\partial P_i'''}{\partial p} - g\frac{\partial P_i}{\partial p} \tag{12}$$

$$\frac{dq_a}{dt} = (1-\delta_m)g\left(\frac{\partial P_l}{\partial p} + \frac{\partial P_i}{\partial p}\right) + \delta_m g\frac{\partial}{\partial p}\left(q_a\frac{P_l+P_i}{1-q_r-q_s}\right) - g\frac{\partial J_{q_a}}{\partial p} \tag{13}$$

where $J_{q_a}, J_{q_v}, J_{q_l}$ and J_{q_i} are the respective diffusive fluxes such that $J_{q_a} + J_{q_v} + J_{q_l} + J_{q_i} = 0$. Using the equations above and the relation

$$L_{l|i}(T) = L_{l|i}(T=0) + (c_{pv} - c_{l|i})T \tag{14}$$

we can write the *local* thermodynamic equation in the following flux conservative form:

$$\left(\frac{\partial}{\partial t}(c_p T)\right)_\Phi = -g\frac{\partial}{\partial p}[(c_l - c_{pd})P_l T + (c_i - c_{pd})P_i T - \delta_m(\hat{c} - c_{pd})(P_l + P_i)T]$$

$$+L_l(T=0)\left(\frac{\partial P_l'}{\partial p}-\frac{\partial P_l'''}{\partial p}\right)+L_i(T=0)\left(\frac{\partial P_i'}{\partial p}-\frac{\partial P_i'''}{\partial p}\right)-g\frac{\partial J_s}{\partial p}-g\frac{\partial J_{rad}}{\partial p}=-g\frac{\partial J_{total}}{\partial p} \tag{15}$$

with

$$\hat{c}=\frac{c_{pd}q_a+c_{pv}q_v+c_lq_l+c_iq_i}{1-q_r-q_s}, \tag{16}$$

J_s the diffusive enthalpy flux and J_{rad} the radiative flux. J_{total} is a short-hand notation for showing that the whole right hand side of equation (15) is only a flux divergence.

To make this a workable set of equations we need a link between the relative fluxes/velocities used here and the absolute ones which will result from certain parameterizations. If we put $P_l^*=w_r^*\rho_r$ and $P_i^*=w_s^*\rho_s$ respectively the rain and snow flux as seen by the micro-physical scheme (the 'absolute' fluxes, with w_r^* and w_s^* the 'absolute' falling velocities), then we can write the corresponding barycentric fluxes as

$$P_l = w_r\rho_r=(1-q_r)P_l^*-q_rP_i^* \tag{17}$$

$$P_i = w_s\rho_s=(1-q_s)P_i^*-q_sP_l^*, \tag{18}$$

so that

$$P_l+P_i=(1-q_r-q_s)(P_l+P_i)^*. \tag{19}$$

The *full* thermodynamic equation can be derived starting form the general entropy expression given by Marquet which has been tranformed to have an expression per unit of mass:

$$S'=q_aS=(q_ac_{pd}+q_tc_{pv})\ln(T)-q_aR_d\ln(p-e)-q_tR_v\ln(e)-\frac{L_l(T)}{T}(q_l+q_r)-\frac{L_i(T)}{T}(q_i+q_s) \tag{20}$$

with $q_t=q_v+q_l+q_r+q_i+q_s$ and S' the total entropy per unit of mass. Note that this expression does not include any precipitation processes but those will be added later. Time derivation and inclusion of the entropy changes associated with precipitation (following Bannon), we finaly find (with $\widetilde{Q}$ the diabatic heat source out of which the precipitation effects are filtered)

$$\widetilde{Q}=c_p\frac{dT}{dt}-\frac{1}{\rho}\frac{dp}{dt}-gL_l(T)\left(\frac{\partial P_l'}{\partial p}-\frac{\partial P_l'''}{\partial p}\right)-gL_i(T)\left(\frac{\partial P_i'}{\partial p}-\frac{\partial P_i'''}{\partial p}\right)$$

$$+\left[gc_lP_l+gc_iP_i-g\hat{c}(P_l+P_i)\right]\frac{\partial T}{\partial p} \tag{21}$$

which we can rewrite as

$$c_p\frac{dT}{dt}-RT\frac{d\ln(p)}{dt}=Q, \tag{22}$$

with Q the full diabatic heat source. If we define this heat source as

$$Q=-g\frac{\partial J_{total}}{\partial p}-T\frac{dc_p}{dt}. \tag{23}$$

we can rewrite (21) as

$$\frac{d(c_pT)}{dt}-RT\frac{d\ln(p)}{dt}=-g\frac{\partial J_{total}}{\partial p}. \tag{24}$$

All the above is only true when the addition/removal of heat is projected only on a temperature change and has no pressure change equivalent. This is of course no problem in the hydrostatic case but when going to non-hydrostatism, we would like to include also the more physical option where any heat source is projected on both temperature and pressure changes.

In the general case (phase-changes and precipitation) we have in the hydrostatic and anelastic cases the following set of equations (with $D_3 = -d\ln(\rho)/dt$ the three-dimensional divergence):

$$c_p \frac{dT}{dt} - RT\frac{d\ln(p)}{dt} = Q \tag{25}$$

$$c_v \frac{d\ln(p)}{dt} + c_p D_3 = 0, \tag{26}$$

whereas in the compressible case we have the following option

$$c_v \frac{dT}{dt} + RT\ D_3 = Q + T\frac{dR}{dt} \tag{27}$$

$$c_v \frac{d\ln(p)}{dt} + c_p D_3 = \frac{Q}{T} + \frac{c_p}{R}\frac{dR}{dt}. \tag{28}$$

Note that the first equation of both sets are equivalent. From the practical point of view this means replacing

$$\frac{d(c_p T)}{dt} - RT\frac{d\ln(p)}{dt} = -g\frac{\partial J_{total}}{\partial p} \tag{29}$$

$$c_v \frac{d\ln(p)}{dt} + c_p D_3 = 0 \tag{30}$$

by

$$\frac{d(c_v T)}{dt} + RT\ D_3 = -g\frac{\partial J_{total}}{\partial p} \tag{31}$$

$$\frac{d\ln(c_v p/R)}{dt} + \frac{c_p}{c_v} D_3 = -\frac{g}{c_v T}\frac{\partial J_{total}}{\partial p} \tag{32}$$

3.3 Rules C

Some routines will need to add so-called 'diagnostic-equivalent' variables to their local input/output stream. Apart from the already available tendencies of T, q_v, q_l, q_r, q_i and q_s, we also need 6 time integrals: 4 time integrals of the moisture rate of change for the possible phase changes in the proposed micro-physical scheme (so these are always related to water vapor, see rule B), i.e.

$$\int \dot{q}_{v\to l}\ dt, \int \dot{q}_{v\to i}\ dt, \int \dot{q}_{v\to r}\ dt, \int \dot{q}_{v\to s}\ dt. \tag{33}$$

These are the equivalent of (from left to right): condensation (P_l'), freezing (P_i'), evaporation (P_l''') and sublimation (P_i'''). Besides these 4 quantities (which are needed in the thermodynamic equation) we need, in order to keep our budgets closed, two more time integrals related to the auto-conversion between the condensates, i.e.

$$\int \dot{q}_{l\to r}\ dt, \int \dot{q}_{i\to s}\ dt. \tag{34}$$

Together with the expression of the tendencies we now have a closed set of 12 equations with 12 unknowns ($J_s, J_v, J_l, J_i, P_l', P_l'', P_l''', P_l, P_i', P_i'', P_i'''$ and P_i). This set of 12 equations is fortunately very easy to solve but this will not be demonstrated here because of the lengthy derivation.

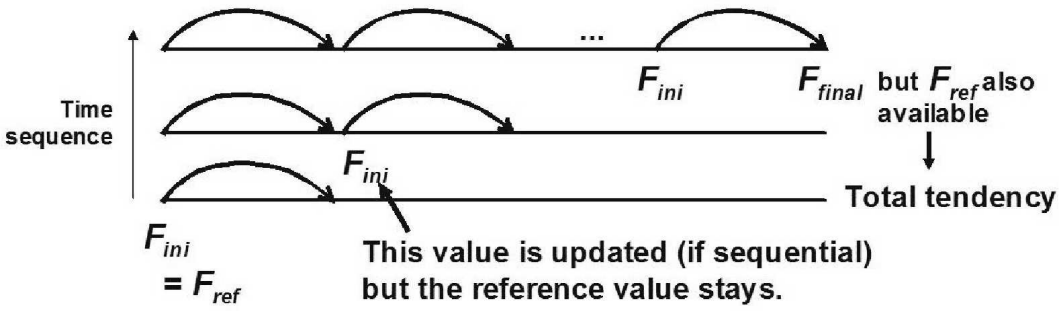
At the moment DDH provides cumulated (from 0 to current time) tendencies and fluxes, from dynamics and physics, for all prognostic variables that were used in ALADIN so far. This is stored in a DDH file and later processed with DDH tools (summing, accumulating and differencing of fields). Additional fields will need to be added to this system in parallel with the addition of four diagnostic-equivalent updates (two already exist in the current framework).

The part where tendencies are calculated from divergence of physical fluxes will also have to be updated. To minimise all these modifications at the level of the DDH machinery, low-level routines or intermediate 'communication' routines should of course prepare the necessary input, in the spirit of what is currently done in IFS for the ECMWF physics.

3.4 Rule D

In order to become independent of a single and fixed way to organize the time-step (as it is currently the case in ALADIN: the order and construction of the routines are such that physics is applied before dynamics and in a parallel way), the calling of the parameterisations should be organized in such a way that the lower level routines are unaware of the followed time-stepping strategy. This means the inclusion of a number of options (see the various possibilities according to rule E) so that in theory every possible combination is possible without changing the low level routines.

Moreover, we must keep in mind that it is more logical to sum fluxes than tendencies because tendencies depend on the way the time discretisation is performed, while fluxes have a 'physical' absolute meaning. This is indeed by following this 'Green-Ostrogradsky' line of thoughts that rule B was designed, starting from the general universal set of non-discretised equations proposed for AROME. For non-linear computations like those associated to the resulting thermodynamic equation or to the evolution of the turbulent kinetic energy, the sum of the tendencies of the different processes is thus not equivalent to the total tendency including all processes after the fluxes' summation. This problem is easy to solve in the case of 'parallel calls' to all low-level routines, but it becomes more complex when one uses the 'sequential calls' strategy for the same routines. To overcome the difficulty, we then need to find a way that always allows to go back to the start of any 'sequence' of parameterisation-related calls. In the simple case of routines exchanging only end and start points (see the graphical representation) we propose to communicate between low-level routines on the basis of three values (statuses) for each prognostic or diagnostically equivalent variable: a **reference** input value which is the value at the beginning of the relevant sequence inside the time-step (to be defined at the control level) and is untouched during the said sequence; an **initial** input value specific to the computed process and which is updated after each of them in the sequential case but not in the parallel case; and a **final** output value which determines the tendency of the process itself but also helps computing consistent fluxes throughout the sequence inside the time-step.



If it was only for the above purpose, it would obviously not be necessary to pass the reference status to the low-level routines. But there are cases in the sequential strategy when the scientific choice is to use as input for a given process not only the state of the atmosphere incorporating the impacts of the preceding steps but also part or all of the associated tendencies (current example: the turbulent vertical diffusion in the IFS model that puts dynamical tendencies on the right hand side of its linear

system inversion process). Here the knowledge inside the routine itself of the reference values allows the flexibility to transparently use the computations in both sequential (see above) and parallel modes (with identity between reference and initial statuses in the latter case). Of course, the return of a tag telling whether or not this possibility was used is necessary in order to perform a consistent conversion to the conservative flux-form at the control level. More complex strategies are probably possible with this system, perhaps requiring the use of other (optional) arguments, but they have not been explored yet.

3.5 Rules E

Since the ensemble considered here should deal with all kind of grid sizes, one knows from experience and theory that the model setup should allow for a comprehensive set of parameterizations including convection at one end and sophisticated microphysics at the other end.

The inclusion of microphysics means that more prognostic variables may be used. The possibility to combine parameterisations coming from different origins is also interesting. However, allowing for such a lot of flexibility in a more complex environment surely implies a careful design, with for instance a control mechanism that automatically checks whether a given selection of parameterisations and options makes sense or not. The following components connected to a flexible way of running and experimenting on physics are now proposed:

3.5.1 Definition of "processes"

In order to avoid unrealistic experimentation it seems very natural to define, perhaps in a fairly rigid way, the different processes which the model may describe. The overall idea is that for a given model run a number of parameterizations are chosen, each characterised e.g. by a given integer number. The different parameterizations are not allowed to "overlap" in the sense that the same process would be described twice.

For example, some convection scheme as presently coded may often include not only vertical subgrid transports but also the precipitation release with associated microphysics. Hence it cannot be combined with another microphysics scheme treating precipitation release in general because then precipitation processes are treated twice. If it is possible after some minor rewriting to avoid the actual precipitation release from the convection scheme it should be possible to combine it with a more advanced precipitation release scheme. This example seems to apply in the HIRLAM case for the present reference scheme where precipitation release and associated microphysics are treated in a separate routine that could be switched off and probably be replaced by a call to a sophisticated precipitation release with little rewriting.

One may imagine an ASCII file defining the different processes, e.g. "dynamics", "turbulence", "shallow convection", "deep convection", "grid scale condensation", "cloud cover parameterisation", "precipitation release including phase changes ", "solar radiation", "thermal radiation", "surface processes". An agreement to the full set of definitions would have to be made. Furthermore the so-defined processes must of course be characterised by the number and kind of DDH-type terms (see rule C) they treat.

3.5.2 Definition of "schemes"

A "scheme" as used today may often consist of several subroutines that, as a whole, describe one or more physical processes. In a revised system the idea is that a scheme is characterized by a unique assigned number (integer) followed by information e.g. integers associated with the processes that are

described by the parameterization. It may be one or more processes as defined in the process definition file. Some schemes as coded today have some switches in their namelists allowing perhaps for different number of processes described. In that case each combination of possible processes feasible with the scheme should probably (for convenience) be defined as a scheme. Taking as an example again a condensation scheme that can be run with or without precipitation release, it defines possibly two separate schemes. The one without precipitation release can be combined with a more advanced precipitation release, the other not.

In order to link the scheme-process correspondence with the above mentioned DDH-type classification of the processes it should also be specified (e.g. also by integers) which prognostic variables are updated by the scheme. In application of rule A, the relevant time-step information should also quantify the scheme (position of call and parallel vs. sequential type).

In summary, it is proposed that a file is established with a unique definition of any scheme in terms of identifying numbers (e.g. by integers).

3.5.3 Definition of current run and of consistency control mechanism

While the above can be implemented at the beginning as a mean for an a posteriori control (and only later as a mean for fully defining the physics' configuration from the set-up stage) a special namelist must anyhow be created containing the schemes of the individual processes chosen for the given model. In order for the user to actually choose a meaningful combination of schemes an offline checking program could be made verifying that some combinations are sensible. In case that the same process is treated more than once a serious warning would be printed out. The information to the checking program would come from the 'scheme definition file' and the 'process definition file'. In case the model can be run with a reduced set of prognostic variables (to save memory) it will also be possible to warn against impossible schemes requiring prognostic variables that will not be available. In order to make the suggested system work it will require some rewriting of communication parts and namelists. If this latter work is manageable the ideas above may be a way forward to achieve flexibility when progressively replacing the 'warning' function by a 'control' one, as already mentioned.

4 Consequences for low-level routines (mainly but not only with the Meso-NH situation in mind)

4.1 Rule A

It is difficult at that stage to be more precise about the definition and use of appropriate low-level switches (like for instance for the possibility to take surface slopes into account in the physics calculations). But there shall be no single explicit link in low-level routines to the high-level switches. Hence choices (**a**), (**c**) and (**d**) of our fourth goal (see above) should be removed from the individual parameterisation routines, if currently present in them (one hopes it is not the case, of course). The exception made for choice (**b**) is linked to the fact that it interferes with low-level dynamical routines and should therefore better be treated homogeneously with respect to the data flow structure, i.e. as a low-level switch of multiple occurrences (like currently in ARPEGE/ALADIN). The default will eventually be (δ_m=1), as the more logical solution for a fully mass weighted set of equations. Hence routines already implicitly in this configuration don't need to be modified.

4.2 Rule B

Someone should verify, on the paper, the consistency of the internal algorithms of each routine with the set of equations. This is not mandatory (the system will still work, especially owing to the provision

of the 'correction mechanism' (cf. supra) to be used as a temporary measure) but good scientific practice.

4.3 Rules C

Some routines may need to add so-called 'diagnostically equivalent' variables to their local (input/)output stream (typically the time integral of a moisture rate of change due to either of the water phase- or type modification).

4.4 Rule D

Internal computations of the routines using a 'middle' time level will have to consider it as a 'non-prognostic' input, or better to get rid of its algorithmic use (using the 'reference status' to keep this particularity would mean a cheating of the system that could lead to very serious bugs at a later stage and should therefore be strictly avoided).

4.5 Rules E

If zero-D-type routines (typically thermodynamic-adjustment ones) would require to be called several times in a time-step, the above-mentioned attribute mechanism ought to be activated, at least inside the scope of application of the proposed interface. The future application of the rules in a progressively more 'control-type' mode would require marginal evolutions of the low level routines' I/Os in order to help structuring the overlying arborescence of calls that will contain the control facilities.

5 Outlook

The present document does not have the ambition to be perfect. First of all its perimeter was the result of a set of arbitrary choices, guided by the double aim to have rather high ultimate ambitions, in order to be able to federate further efforts, but to also care for a quasi-continuity with the existing situation in the early phase of a progressive potential implementation plan, in order to avoid unacceptable constraints. There are surely some omissions, or small errors or ambiguities inside the document, which readers with a different angle of view will discover and help correcting. Should it lead to any concrete action, unexpected problems will unavoidably show up. They should be treated pragmatically in the respect of the general economy of the proposal. The authors do however believe that this is the best document they could produce within a very short time in order to fulfil the task assigned to them by the TCWGPDI, while producing something consistent in their own eyes. Since we want in principle to avoid the never-ending cycle of not enough precise specifications leading to requests for complements of information rather than impact studies, we elected to be a bit didactic in the Section of *'Further precisions'* and to be quite 'to the point' in the Section of *'Consequences for low-level routines'*.