Test implementation of new semi-Lagrangian interpolators in ARPEGE/ALADIN cycle 32t1alr01

Results of RC LACE research stay in Prague 16.7.–18.8.2007
under scientific supervision of Filip Váňa

1 Introduction

This document briefly describes test implementation of new semi-Lagrangian interpolators in ARPEGE/ALADIN cycle 32t1alr01. It is meant mainly as technical guide giving quick overview of general code design. It does not address scientific issues which will be covered in separate report.

Main objective of presented work was to modify SLHD scheme (Semi-Lagrangian Horizontal Diffusion) by introducing 1-parametric class of second order accurate interpolators [1] which are more selectively damping than currently used blend of cubic Lagrange polynomial and linear interpolator with smoother. Increased accuracy of the new interpolators should improve conservative properties of semi-Lagrangian scheme, but for the moment it is not clear whether their diffusivity will be sufficient for SLHD scheme. Therefore, extensive 3D testing should follow.

Secondary task was to improve efficiency of semi-Lagrangian interpolations, if possible. This issue becomes important with introduction of new 3D prognostic fields which are subject to advection and possibly SLHD.

2 New semi-Lagrangian interpolators

High order 3D and 2D interpolators in model (using 32 and 12-point stencils) are built from simpler 1D interpolators, which work on 4-point stencil [4]. In the old scheme high order 4-point interpolator is cubic Lagrange polynomial, in the new scheme it is replaced by 1-parametric class of second order accurate interpolators with tunable diffusivity. This class can be constructed as weighted combination of any two such interpolators. Most pragmatic choice seems to be combination of cubic Lagrange interpolator F_{lag} with quadratic interpolator F_{quad}:

$$F = F_{\text{lag}} + \kappa(F_{\text{quad}} - F_{\text{lag}}) \qquad \kappa_{\min} \leq \kappa \leq \kappa_{\max} \tag{1}$$

Here κ is weight controlling diffusivity of interpolator F and it is not necessarily restricted to interval $[0, 1]$. In SLHD scheme it is derived from horizontal deformation of the flow [2]. Diffusion strength of interpolator F increases with κ, while order of equivalent diffusion[1] remains 4.

[1]Concept of equivalent diffusion is well defined only for longer waves. For short waves, damping properties of interpolator can differ significantly from power law.

Reasons leading to this choice of basic interpolators are following: Cubic Lagrange interpolator F_{lag} is already implemented in the model and its computation is fairly optimized. From remaining second order accurate interpolators quadratic interpolator F_{quad} is the cheapest one, since all other interpolators are cubic. Using notations from appendix A of [1] it can be expressed as:

$$F_{\text{quad}}(\xi, p, q, y_0, y_1, y_2, y_3) = y_1(1-\xi) + y_2\xi + a\xi(1-\xi) \tag{2}$$

$$a = \frac{-y_0 + y_1(1+p-q) + y_2(1-p+q) - y_3}{p(1+p) + q(1+q)}$$

$$\xi \equiv \frac{x - x_1}{x_2 - x_1} \qquad p \equiv \frac{x_1 - x_0}{x_2 - x_1} \qquad q \equiv \frac{x_3 - x_2}{x_2 - x_1}$$

Comparing with equation (9) of [1] it can be seen that formula (2) is equivalent to following expressions for parameters (a_1, a_2):

$$a_1(p,q) = -\frac{1}{p(1+p) + q(1+q)} \qquad a_2(p,q) = \frac{1}{p(1+p) + q(1+q)} \tag{3}$$

In case of regular nodes ($p = q = 1$) they reduce to $(a_1, a_2) = (-\frac{1}{4}, \frac{1}{4})$.

3 Code implementation

Described code implementation is only preliminary. Its main goal was to enable extensive 3D testing, which should demonstrate capabilities of the new interpolators. Test results may imply additional modifications of the code.

Most essential change was to precompute SLHD interpolation weights in `LASCAW/ELASCAW`. It means that these subroutines now compute two sets of weights: normal (corresponding to $\kappa = \kappa_{\min}$) and diffusive (using actual value of κ). Setting $\kappa_{\min} \neq 0$ redefines basic high order interpolator for all advected fields, i.e. not only those which are subject to SLHD.

In order to improve code efficiency, computation of weights for high order interpolations in longitude were moved from `LAITRI/LAIDDI` to `LASCAW/ELASCAW`. In the old scheme they were computed repeatedly for each advected field, now they are computed only once. In the new scheme subroutines `LAITRI/LAIDDI` are no longer restricted to cubic Lagrange polynomial and the type of interpolation is fully controlled by input weights.

3.1 Simplified calling trees

Given trees are relevant for 3D prognostic fields which are subject to SLHD. In case of quasi-monotonic treatment, corresponding `QM` or `QMH` counterpart of interpolation subroutine is called. Parts under `LARCINHA` and `LARCINHB` are active only for non-hydrostatic case with advection of vertical velocity w (option `LGWADV`), when half level quantity needs to be advected.

In the old scheme weight κ diagnosed in subroutine `LATTE_KAPPA` is passed up to the level of `GP_MODEL` and then through `LAPINEB` branch of `CALL_SL` down to `LAITSLD`.

Here it is finally used to combine results of high order interpolations (performed by LAITRI) and linear interpolations with additional smoother (performed by LAITLI_HD). Combintation is done in LADIFF.

Simplified calling tree of the old scheme:

```
GP_MODEL
  CPG
  | CPG_DYN
  |   LACDYN
  |     LATTE_KAPPA -> horizontal deformation, weight kappa
  |
  CALL_SL
    LAPINEA -> weights for interpolations into origin points
    | LARMES/ELARMES -> origin points of SL trajectories
    | | LARCINA
    | |   LASCAW/ELASCAW -> weights for trajectory research
    | |
    | LARCINA
    | | LASCAW/ELASCAW -> full level interpolation weights
    | |
    | LARCINHA
    |   LASCAW/ELASCAW -> half level interpolation weights
    |
    LAPINEB -> interpolated 3D and 2D fields
      LARCINB -> interpolated 3D full level and 2D fields
      | LAITRE_GMV
      | | LAITSLD/LAITSLDQM/LAITSLDQMH
      | |   LAITRI/LAITQM/LAITQMH -> 3D cubic Lagrange interpolations
      | |   LAITLI_HD -> 3D linear interpolations with smoother
      | |   LADIFF -> blending the two with weight kappa
      | |
      | LAITRE_GFL
      | | LAITSLD/LAITSLDQM/LAITSLDQMH
      | |   ...
      | |
      | LAIDDI/LAIDQM -> 2D cubic Lagrange interpolations
      |
      LARCINHB -> interpolated 3D half level fields
        LAITRE_GMV
          LAITSLD/LAITSLDQM/LAITSLDQMH
            ...
```

Main difference in the new scheme is that from GP_MODEL weight κ goes through LAPINEA branch of CALL_SL down to LASCAW/ELASCAW. Here it is used to compute diffusive interpolation weights, which are transferred together with normal weights into LABINEB branch of CALL_SL, to be finally used in LAITRI. For the time being no 2D fields are subject to SLHD, i.e. subroutine LAIDDI always gets normal weights on input.

Simplified calling tree of the new scheme:

```
GP_MODEL
  CPG
  | CPG_DYN
  |   LACDYN
  |     LATTE_KAPPA -> horizontal deformation, weight kappa
  |
  CALL_SL
    LAPINEA -> weights for interpolations into origin points
    | LARMES/ELARMES -> origin points of SL trajectories
    | | LARCINA
    | |   LASCAW/ELASCAW -> weights for trajectory research
    | |
    | LARCINA
    | | LASCAW/ELASCAW -> two sets of full level interpolation
    | |                   weights (normal and diffusive)
    | LARCINHA
    |   LASCAW/ELASCAW -> two sets of half level interpolation
    |                     weights (normal and diffusive)
    LAPINEB -> interpolated 3D and 2D fields
      LARCINB -> interpolated 3D full level and 2D fields
      | LAITRE_GMV
      | | LAITRI/LAITQM/LAITQMH -> 3D interpolations using
      | |                          diffusive weights
      | LAITRE_GFL
      | | LAITRI/LAITQM/LAITQMH -> 3D interpolations using
      | |                          diffusive weights
      | LAIDDI/LAIDQM           -> 2D interpolations using
      |                            normal weights
      LARCINHB -> interpolated 3D half level fields
        LAITRE_GMV
          LAITRI/LAITQM/LAITQMH -> 3D interpolations using
                                   diffusive weights
```

3.2 Removed subroutines

`arp/adiab:`

LADIFF
LAIDSP/LAIDSPQM
LAITLI_HD
LAITSLD/LAITSLDQM/LAITSLDQMH
LAITSLDSP/LAITSLDSPQM/LAITSLDSPQMH
LAITSP/LAITSPQM/LAITSPQMH

In order to enable use of old SLHD scheme within new code structure, functionality of smoother from LAITLI_HD still has to be transferred to LASCAW/ELASCAW.

Beside necessary design changes related to introduction of new semi-Lagrangian interpolators it was decided to remove spline interpolators [3] which were originally introduced to improve conservative properties of SLHD scheme (option `LRSPLINE_[X]` for GMV field `[X]`, resp. attribute `LRSPLINE` for GFL fields). In practice they caused unexpected deterioration of temperature and geopotential bias at some parts of troposphere, probably due to their slight amplification of long waves. That is why they were switched off and did not enter operations.

3.3 New subroutines

None for the moment.

3.4 Modified subroutines

In the old scheme weights for cubic Lagrange interpolations in latitude and in vertical are passed via arrays `PCLA` and `PVINTW`. In the new scheme additional array `PCLO` for passing interpolation weights in longitude was introduced. It is dimensioned `PCLO(KPROMA,KFLEV,3,2)` since in longitudinal direction three weights on two latitude rows are needed. In order to have consistent notations, dimensioning of arrays `PCLA` and `PVINTW` was shifted accordingly in modified subroutines:

old		new
`PCLA   (KPROMA,KFLEV,2:4)`	→	`PCLA   (KPROMA,KFLEV,3)`
`PWINTW(KPROMA,KFLEV,2:4)`	→	`PVINTW(KPROMA,KFLEV,3)`

Additional set of diffusive interpolation weights for SLHD scheme is passed via new arrays `PCLOSLD`, `PCLASLD` and `PVINTWSLD`. Array `PVINTDS` containing vertical interpolation distances for splines was removed.

`ald/adiab:`

`ELARMES` – removed arguments `KULOUT`, `PVINTDS`. Added argument `PCLO`. Modified calls to `LARCINA` with dummy variables in place of SLHD related quantities (diffusive weights are useless in trajectory research).

`ELARMES5` – modified call to `LARCINA` with dummy variables in place of SLHD related quantities.

`ELASCAW` – removed arguments `KDGUN`, `KFLDX`, `KBUFP` and `PVINTDS`. Added arguments `PVSLD`, `PKAPPA`, `PCLASLD`, `PCLO`, `PCLOSLD` and `PVINTWSLD`. Auxilary functions `FLAG1,2,3` (cubic Lagrange weights) and `FQUAD1,2,3` (quadratic weights) introduced for case with regular nodes:

$$
\begin{aligned}
\mathrm{FLAG1}(\xi) &= \tfrac{1}{2}(\xi+1)(\xi-1)(\xi-2) & \mathrm{FQUAD1}(\xi) &= (1-\xi)(1+\tfrac{1}{4}\xi)\\
\mathrm{FLAG2}(\xi) &= -\tfrac{1}{2}(\xi+1)\xi(\xi-2) & \mathrm{FQUAD2}(\xi) &= \xi(\tfrac{5}{4}-\tfrac{1}{4}\xi)\\
\mathrm{FLAG3}(\xi) &= \tfrac{1}{6}(\xi+1)\xi(\xi-1) & \mathrm{FQUAD3}(\xi) &= \tfrac{1}{4}\xi(\xi-1)
\end{aligned}
$$

Computation of diffusive weights is activated only for `LSLHD` with one of `KWIS` configurations 103, 104, 105 and 203. In case of iterative scheme (`NSITER>0`) it is performed only for the last iteration (NCURRENT_ITER==NSITER), for previous iterations cubic Lagrange weights are returned. Impact of this simplification on convergence was not evaluated yet.

In ALADIN irregular nodes can occur only in vertical, where quadratic interpolator is evaluated using auxiliary array PVSLD. Its elements 1, 2 and 3 (first index) contain values $-(1+p-q)$, $-(1-p+q)$ and $-1/[p(1+p)+q(1+q)]$ which depend only on definition of eta coordinate and can be precomputed in setup. Computation of quadratic weights than proceeds as follows (symbol := denotes assignment):

$$
\begin{aligned}
w_2 &:= \xi \\
w_1 &:= 1 - w_2 \\
w_3 &:= \mathrm{PVSLD}_3 \cdot w_1 \cdot w_2 \\
w_1 &:= w_1 + \mathrm{PVSLD}_1 \cdot w_3 \\
w_2 &:= w_2 + \mathrm{PVSLD}_2 \cdot w_3
\end{aligned}
$$

In this way all three weights are evaluated using 3 additions and 4 multiplications. Weight w_0 is never computed explicitly, since it sums up to 1 with the remaining weights.

`ald/setup:`

`SUEGEO2` – fix for SLHD in vertical plane 2D model.

`arp/adiab:`

`CALL_SL` – added local arrays `ZCLO`, `ZCLOH`, `ZCLOSLD`, `ZCLOHSLD`, `ZCLASLD`, `ZCLAHSLD`, `ZVINTWSLD` and `ZVINTWHSLD` needed for transfer of extra weigths from `LAPINEA` to `LAPINEB`. Removed local arrays `ZVINTDS` and `ZVINTDSH` which were used to transfer vertical distances for spline interpolations. Modified calls to `LAPINEA` and `LAPINEB`.

`CALL_SL_AD` – added local array `ZCLO5` needed for transfer of interpolation weights in longitude from `LAPINEA5` to `LAPINEBAD`. Modified calls to `LAPINEA5` and `LAPINEBAD`.

`CALL_SL_TL` – added local array `ZCLO5` which should be filled by `LAPINEA5`. Modified call to `LAPINEA5`.

`CPG2LAG` – removed useless local array `IBUFP1`. Modified call to `LADINE`.

`LADINE` – removed argument `KBUFP`. Modified calls to `LARMES2` and `LAINOR2`.

`LADINEAD` – modified call to `LAINOR2`.

`LAIDDI/LAIDQM` – removed argument `PDLAT`. Added argument `PCLO`. Interpolations fully driven by input weights.

`LAINOR2` – removed argument `KBUFP`. Modified call to `LARCIN2`.

LAITRE_GFL – removed arguments LDSLHD, PVINTDS and PKAPPA. Added arguments PCLO, PCLOSLD, PCLASLD and PVINTWSLD. Interpolation subroutines LAITRI/LAITQM/LAITQMH are called with normal or diffusive weights, depending on configuration string CDINT.

LAITRE_GMV – removed arguments LDSPLINE, PDVER, PVINTDS and PKAPPA. Added arguments PCLO, PCLOSLD, PCLASLD and PVINTWSLD. Interpolation subroutines LAITRI/LAITQM/LAITQMH are called with normal or diffusive weights, depending on argument LDSLHD.

LAITRI/LAITQM/LAITQMH – removed argument PDVER. Added argument PCLO. Interpolations fully driven by input weights.

LAPINEA – removed arguments PVINTDS and PVINTDSH. Added arguments PKAPPA, PCLO, PCLOSLD, PCLASLD, PVINTWSLD, PCLOH, PCLOHSLD, PCLAHSLD and PVINTWHSLD. Modified calls to LARMES/ELARMES, LARCINA and LARCINHA. Arrays RSLD1, RSLD2, RSLD3 from module YOMLEG and VSLD/VSLDH from YOMGEM are passed to LARCINA/LARCINHA as input arguments.

LAPINEA5 – added argument PCLO5. Modified call to LARCINA with dummy variables in place of SLHD related quantities.

LAPINEB – removed arguments PVINTDS, PVINTDSH and PKAPPA. Added arguments PCLO, PCLOSLD, PCLASLD, PVINTWSLD, PCLOH, PCLOHSLD, PCLAHSLD and PVINTWHSLD. Modified calls to LARCINB and LARCINHB.

LAPINEBAD – added argument PCLO5. Modified call to LARCINB5.

LARCIN2 – removed argument KBUFP. Added local array ZCLO needed for transfer of interpolation weights in longitude from LASCAW/ELASCAW to LAIDDI/LAIDQM. Modified calls to LASCAW/ELASCAW with dummy variables in place of SLHD related quantities.

LARCIN2AD – modified call to LASCAW with dummy variables in place of SLHD related quantities.

LARCINA – removed arguments KDGUN and PVINTDS. Added arguments PSLD1, PSLD2, PSLD3, PVSLD, PKAPPA, PCLO, PCLOSLD, PCLASLD and PVINTWSLD. Modified calls to LASCAW/ELASCAW.

LARCINB – removed arguments LDSPLINEW, LDSPLINET, LDSPLINEPD, LDSPLINEVD, LDSPLINEP, PVINTDS and PKAPPA. Added arguments PCLO, PCLOSLD, PCLASLD and PVINTWSLD. Modified calls to LAITRE_GMV, LAITRE_GFL and LAIDDI/LAIDQM. Removed calls to LAIDSP/LAIDSPQM.

LARCINB5 – added argument PCLO. Modified calls to LAITRI.

LARCINHA – removed arguments KFLDX, KDGUN and PVINTDSH. Added arguments PSLD1, PSLD2, PSLD3, PVSLDH, PKAPPA, PCLOH, PCLOHSLD, PCLAHSLD and PVINTWHSLD. Modified calls to LASCAW/ELASCAW.

LARCINHB – removed arguments LDSPLINEVD and PVINTDSH. Added arguments PCLOH, PCLOHSLD, PCLAHSLD and PVINTWHSLD. Modified call to LAITRE_GMV.

LARMES2 – removed arguments KBUFP and PDT. Modified call to LARCIN2.

LARMES25 – modified call to LARCIN2.

LARMES – removed argument PVINTDS. Added argument PCLO. Modified calls to LARCINA with dummy variables in place of SLHD related quantities (diffusive weights are useless in trajectory research).

LARMES5 – modified call to LARCINA with dummy variables in place of SLHD related quantities.

LASCAW – removed arguments KFLDX, KBUFP and PVINTDS. Added arguments PSLD1, PSLD2, PSLD3, PVSLD, PKAPPA, PCLASLD, PCLO, PCLOSLD and PVINTWSLD.

Modifications are analogical to those of ELASCAW, with two exceptions. First, irregular nodes are possible not only in vertical but also in latitude. That is why quadratic interpolator in latitude is evaluated using auxiliary arrays PSLD1, PSLD2 and PSLD3 containing values $-(1+p-q)$, $-(1-p+q)$ and $-1/[p(1+p)+q(1+q)]$. Since these values depend only on used latitude division, they can be precomputed in setup. Second, interpolation nodes on different latitude rows are not necessarily aligned (relative interpolation distance PDLO can be different for each row). For this reason interpolation weights PCLO/PCLOSLD which are needed on the two inner rows have to be computed independently.

LATTE_KAPPA – output array PKAPPA rescaled to vary from SLHDKMIN to SLHDKMAX (in the old scheme it varied from 0 to SLHDKMAX):

$$\kappa = \kappa_{\min} + (\kappa_{\max} - \kappa_{\min}) \frac{\Delta t\, F(d)}{1 + \Delta t\, F(d)}$$

Here Δt is model timestep, d is horizontal deformation of the flow and function $F(d)$ is described in [2] (text around equations (3)–(5) therein).

arp/setup:

SUALLO – added allocation of arrays VSLD and VSLDH (only in LSLHD case, second array is required only for LGWADV option).

SUCSLINT – removed argument LDRSPLINE. Modified setting of the string CDSLINT determining type of GFL semi-Lagrangian interpolation:

old		new
LAITSLD	→	LAITRI(SLD)
LAITSLDQM	→	LAITQM(SLD)
LAITSLDQMH	→	LAITQMH(SLD)
LAITSLDSP		removed
LAITSLDSPQM		removed
LAITSLDSPQMH		removed
LAITSP		removed
LAITSPQM		removed
LAITSPQMH		removed

SUDIM1 – removed setting of attribute LRSPLINE.

SUDYN – removed setting of variables LRSPLINE_[X] and SLHDKMAX.

SUDYNA – added default setting of variables SLHDKMIN and SLHDKMAX which correspond to tresholds κ_{min} and κ_{max}. Enabled SLHD on vertical velocity w in LGWADV case (so far tested only in vertical plane 2D model).

Tests of user supplied values SLHDKMIN and SLHDKMAX are not yet activated, since it is not clear what is the safe range for weight κ. So far it was verified that κ between -2 and 12 provides reasonable results, but this interval may still be expanded or collapsed slightly.

Setting nonzero SLHDKMIN activates LSLHD switch even if SLHD scheme is off for all advected fields. This is necessary because with $\kappa_{\mathrm{min}} \neq 0$ basic high order interpolator is no longer cubic Lagrange polynomial F_{lag}, but the combination $F_{\mathrm{lag}} + \kappa_{\mathrm{min}}(F_{\mathrm{quad}} - F_{\mathrm{lag}})$. In order to compute its weights, both cubic Lagrange and quadratic weights are needed. Active LSLHD switch ensures allocations and setup of auxiliary arrays as well as computation of extra quadratic weights in LASCAW/ELASCAW.

SUSC2B – added allocation and initialization of arrays RSLD1, RSLD2 and RSLD3 (only in LSLHD case). They contain auxiliary quantities $-(1 + p - q)$, $-(1 - p + q)$ and $-1/[p(1 + p) + q(1 + q)]$ needed for quadratic interpolations in latitude.

SUVERT – added initialization of arrays VSLD and VSLDH (only in LSLHD case, second array is filled only for LGWADV option). They contain full and half level quantities $-(1+p-q)$, $-(1-p+q)$ and $-1/[p(1+p)+q(1+q)]$ needed for quadratic interpolations in vertical.

arp/pp_obs:

OBSHOR – removed argument KIBUFP1, modified call to SLINT.

SLINT – removed argument KIBUFP1. Modified calls to LASCAW/ELASCAW with dummy variables in place of SLHD related quantities (diffusive weights are useless for interpolations into observation points).

Before computation of high order interpolation weights variable LSLHD from module YOMDYNA is set to false. This deactivates extra computations in LASCAW/ELASCAW, which then return cubic Lagrange weights. Afterwards LSLHD is reset to its original value.

SLINTAD – modified call to LASCAW/ELASCAW with dummy variables in place of SLHD related quantities.

arp/utility:

DEALLO – added deallocation of arrays VSLD and VSLDH.

3.5 Modified modules

arp/module:

TYPE_GFLS – removed attribute LRSPLINE from type TYPE_GFL_NAML.

YOMDYN – removed variables LRSPLINE_[X]. Variable SLHDKMAX moved to YOMDYNA.

YOMDYNA – added variables SLHDKMIN and SLHDKMAX.

YOMGEM – added allocatable arrays VSLD(:,:) and VSLDH(:,:).

YOMLEG – added allocatable arrays RSLD1(:), RSLD2(:) and RSLD3(:).

3.6 Modified namelists

arp/namelist:

NAMDYN – removed variables LRSPLINE_[X]. Variable SLHDKMAX moved to NAMDYNA. Variables ALPHINT and GAMMAX controlling SLHD smoother were kept, but they have currently no effect since subroutine LAITLI_HD was removed and its functionality was not yet transferred to LASCAW/ELASCAW.

NAMDYNA – added variables SLHDKMIN and SLHDKMAX. They correspond to tresholds $\kappa_{\min}$ and $\kappa_{\max}$. Nonzero SLHDKMIN redefines basic high-order interpolator for all advected fields.

4 Technical validation

Neutral impact of the new code design on existing semi-Lagrangian scheme was tested in adiabatic mode. With SLHD scheme off, deviations from reference solution were following:

model	domain, resolution	number of timesteps	difference in spectral norms
ALADIN	LACE, 9.0 km	10	last digit
ARPEGE	T63, unstretched	48	last 2–3 digits

Dataflow of the new diffusive weights was checked by activating SLHD scheme with SLHDKMIN and SLHDKMAX being zero. In this case set of diffusive weights was passed to interpolation subroutines, but they were effectively computed for cubic Lagrange interpolator. Impact on spectral norms was of the same order as in previous test. Spectral diffusion was turned off completely, in order to prevent activation of reduced and supporting spectral diffusions for fields which were subject to SLHD.

Implementation of the new interpolators was checked in vertical plane 2D model using André Robert's bubble test #1 from [1]. For chosen second order accurate interpolators reference results were obtained with dirty version of the code, specifying directly parameters (a_1, a_2). Then the test was repeated with the new code, setting `SLHDKMIN=SLHDKMAX` to corresponding value. In this way reference results were nicely reproduced, but it must be remembered that vertical plane 2D model does not involve interpolations in longitude.

Impact of the new SLHD scheme on kinetic energy spectrum was evaluated using 6 hour integrations with ALARO-0 physics minus 3MT. SLHD was active for 7 fields: temperature, u and v wind, specific humidity, cloud water, cloud ice and turbulent kinetic energy. In order to better see damping properties of various semi-Lagrangian interpolators, spectral diffusion was switched off completely in these experiments. Figure 1 shows that the new SLHD scheme with reasonable values of $\kappa_{\min}$ and $\kappa_{\max}$ gives comparable amount of damping for short waves as the old one. However, due to its higher scale selectivity action on intermediate waves is much weaker. On one hand this is needed in order to improve conservative properties, on the other hand it is not clear whether it will be sufficient to control development of artificial cyclones.

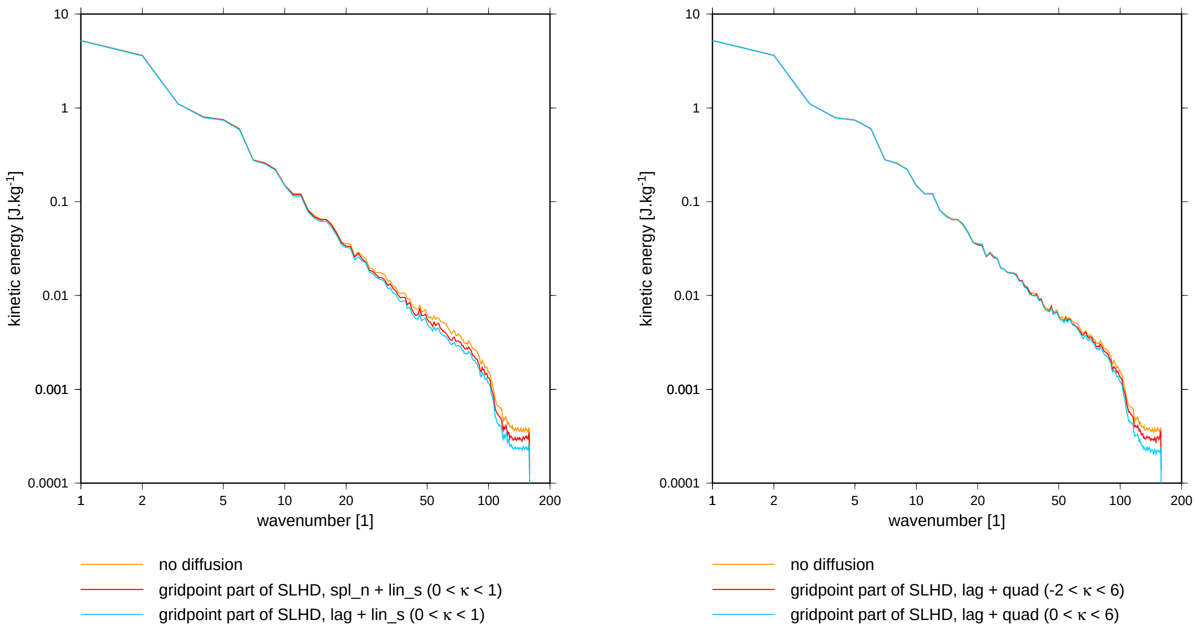

Figure 1: Impact of gridpoint part of SLHD on kinetic energy spectrum. Outcome of 6 hour integration with ALARO-0 physics minus 3MT and deactivated spectral diffusion: left – old SLHD scheme, right – new SLHD scheme. Legend notations: spl_n – natural cubic spline, lag – cubic Lagrange polynomial, lin_s – linear interpolator with smoother, quad – quadratic interpolator.

Improved conservative properties of the new SLHD scheme are demonstrated on figure 2. It shows differences in mean sea level pressure field after 36 hour integrations using again ALARO-0 minus 3MT, but this time always including spectral part of SLHD.

It can be seen that while activating of the old SLHD scheme causes mostly positive bias (dominating red color on the left panel), response of the new SLHD scheme is weaker and better balanced (right panel). Impact on other fields still have to be examined.

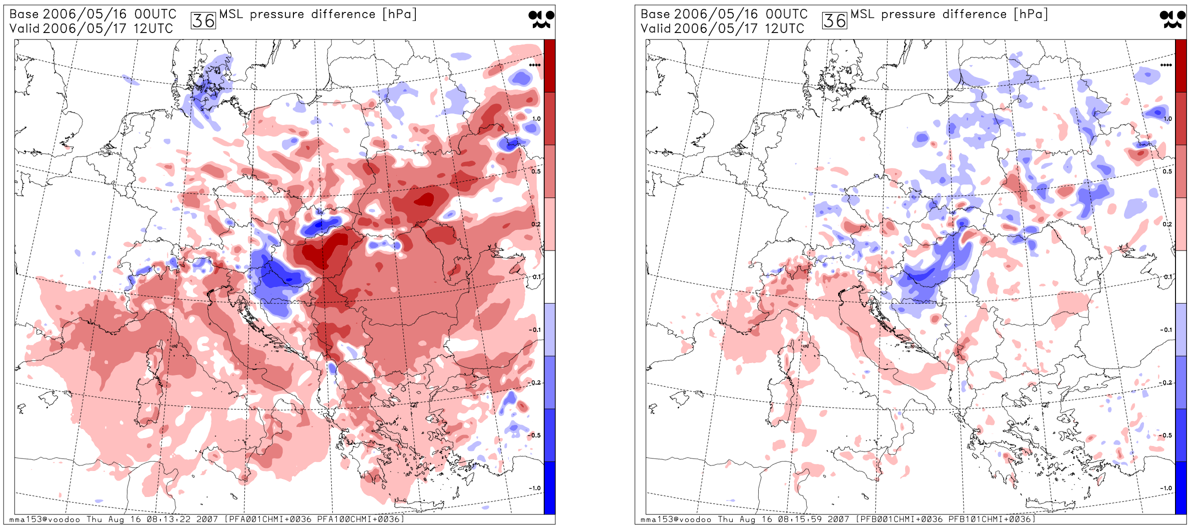

Figure 2: Impact of SLHD scheme on mean sea level pressure. Outcome of 36 hour integration with ALARO-0 physics minus 3MT: left – old scheme using cubic Lagrange polynomial combined with linear interpolator and smoother, right – new scheme with $0 \leq \kappa \leq 6$. Reference integration used cubic Lagrange polynomial ($\kappa = 0$) and spectral diffusion identical with that of SLHD scheme. Contour levels used for pressure difference are ±0.1, ±0.2, ±0.5 and ±1.0 hPa.

Efficiency of the new code was tested on NEC SX-6 platform. It was expected that common computation of all interpolation weights in `LASCAW/ELASCAW` will bring savings in `LAITRI`. However, vectorization of the new `LAITRI` is less efficient, leading to slight increase of its CPU time even if number of operations was reduced. Net effect with 9 advected fields (temperature, u and v wind, specific humidity, cloud water, cloud ice, rain, snow and turbulent kinetic energy) and SLHD scheme off is 0.36 % increase of total CPU time. New code becomes more efficient only with SLHD scheme on, for example with 7 diffused fields (rain and snow undiffused) total CPU time decreases by 5.56 %:

`LSLHD`	t_{old} [s]	t_{new} [s]
`.F.`	1696.322	1702.436 (+0.36 %)
`.T.`	1834.748	1732.816 (−5.56 %)

These results were obtained in benchmark mode using ALARO-0 physics minus 3MT. Length of integrations was 90 timesteps and there was no digital filter initialization.

5 Summary and future steps

New class of second order accurate semi-Lagrangian interpolators was implemented in SLHD scheme. Test code was developed on top of cycle 32t1alr01. Basic technical validation in both ALADIN and ARPEGE was performed, proving neutral impact of the new code structure on semi-Lagrangian computations. Brief look at kinetic energy spectra confirmed higher scale selectivity of the new interpolators compared to the old SLHD scheme. Impact on mean sea level pressure field is weaker and better balanced than for the old scheme, promising improved conservative properties.

Before the new SLHD scheme can be accepted, following items have to be addressed:

- For the moment it is not clear whether higher selectivity of the new SLHD interpolators will provide sufficient damping for intermediate scales. Efficiency in suppressing development of artificial cyclones must be compared against the old scheme. If not sufficient, addition of Laplacian smoother (as proposed by P. Bénard) will have to be considered.
- New scheme has to be run in parallel suite for certain period in order to see its impact on tropospheric scores.
- Resolution independent tunings developed for old SLHD scheme have to be revisited.
- If the new SLHD scheme is going to replace the old one, it will have to coexist with it in model code for some transition period. To enable this, functionality of smoother from `LAITLI_HD` has to be implemented in `LASCAW/ELASCAW`.[2]
- Vectorization of the new `LAITRI` should be improved. Efficiency of the new code on scalar machines was not evaluated yet.
- TL/AD (tangent linear and adjoint) code existing for case `.NOT.LSLHD` was not tested. Once the new SLHD scheme gets its final shape, TL/AD code will have to be written also for `LSLHD` case.

Appendix

A Where to find what

New code is under CVSTUC branches on `kappa`:

```
Ald_mma157_AL32t1alr01_int
Arp_mma157_CY32t1alr01_int
```

Loading scripts are on `kappa`:

```
~mma157/cycle_32t1alr01/load/
  load_00_sx6
  load_00_sx6.ftrace
  load_01_sx6
  load_01_sx6.ftrace
```

[2]The old SLHD scheme cannot be reproduced exactly within the new code, see appendix B.

Used masters are on `archiv`:

```
~mma157/bin/
  master_al32t1alr01_00_sx6
  master_al32t1alr01_00_sx6.ftrace
  master_al32t1alr01_01_sx6
  master_al32t1alr01_01_sx6.ftrace
```

Version `00` is reference, version `01` contains new SLHD interpolators. Executables with suffix `.ftrace` were compiled and loaded with profiling options.

Integration scripts, output listings, input and output FA files can be found on `sx68`. Some of directories are symbolic links, with data being physically located under `/work/` directory:

`~mma157/m3d/exp_slhd/`	root directory
`ecto/`	namelists for ectoplasme
`fpos/`	postprocessing files – ALADIN
`fpos_arp/`	postprocessing files – ARPEGE
`icmsh/`	historical files – ALADIN
`icmsh_arp/`	historical files – ARPEGE
`lbc/`	coupling files for ALADIN
`script/`	integration scripts and output listings
`README`	description of experiments

In order to release disk space, some FA files had to be moved to `archiv` where they are kept in the same directory tree under user `mma157`.

All scripts were written in perl. In order to use them, one must add path

```
/home/mma/mma157/lib/perl
```

into enviromental variable `PERLLIB`. It tells perl where to look for local modules. NQS jobs based on perl scripts must be submitted using command `qsub -x`, so that user environment is copied for the job.

B Follow-up remarks

Some important facts were realized only after the work described in this documentation was finished. Currently these are:

1. Let I_F be 32-point 3D interpolator based on 4-point 1D interpolator F. Then, having 1D interpolators A (accurate) and D (diffusive), there are two ways how to combine them in order to get 3D interpolator with controlled diffusivity:

$$(1-\kappa)I_A + \kappa I_D \neq I_{(1-\kappa)A+\kappa D} \tag{4}$$

As indicated, these two ways are not equivalent. Expression on left hand side is used in the old SLHD scheme, where two 3D interpolations are needed to advect diffused field. Expression on right hand side is used in the new scheme, where only one 3D interpolation is needed. Unfortunately, inequality (4) means that within the new code structure it will not be possible to reproduce old SLDH scheme exactly. Still, there is a chance that the two implementations will give similar results for κ close enough to 0 or 1, since the difference is proportional to $\kappa(1-\kappa)$ and it should be (loosely speaking) third order in $(A-D)$.

2. In the new code, basic high order interpolator (cubic Lagrange polynomial) can be redefined by setting nonzero `SLHDKMIN`. In such case `LSLHD` is set to true even if no fields are subject to SLHD. Undesired side effect is then useless call of subroutine `LATTE_KAPPA`. Therefore, it will be better not to modify original `LSLHD` setting, but to introduce additional logical key which will drive allocation and filling of necessary extra arrays in cases when quadratic interpolator is needed.

References

[1] Mašek, J., 2006: Study of semi-Lagrangian interpolators in idealized framework. *RC LACE stay report*

[2] Váňa, F., 2005: Semi-Lagrangian horizontal diffusion in ALADIN/ARPEGE. *RC LACE/CHMI/ONPP internal document*

[3] Váňa, F., 2005: Spline interpolation in semi-Lagrangian advection scheme of ALADIN/ARPEGE/IFS. *RC LACE/CHMI/ONPP internal document*

[4] Yessad, K., 2007: Semi-Lagrangian computations in the cycle 32 of ARPEGE/IFS. *Météo-France/CNRM/GMAP/ALGO internal document*