**Spline interpolation in semi-Lagrangian advection scheme of ALADIN/ARPEGE/IFS**

Filip Váňa
CHMI/ONPP LACE 4th February 2005

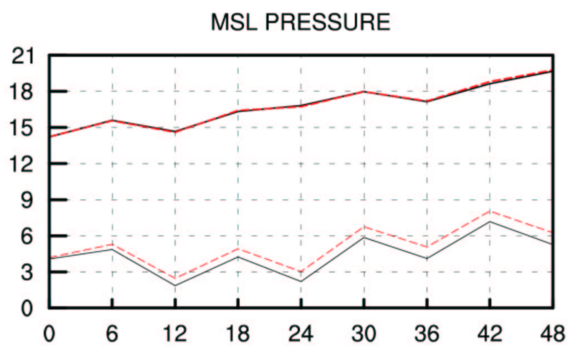
file : ReportSpline.tex

# 1 Introduction

Most operational NWP models are currently using for their advection computation the semi-Lagrangian scheme. This scheme among some other advantages allows fairly longer integrational timestep with respect to the Eulerian advection while preserving the computational model grid contrary to the pure Lagrangian solution. The price to achieve those nice features is typically to perform an interpolation of advected field at every integration timestep. The accuracy of such interpolation is then a compromise given by need to keep it sufficiently precise and not too expensive with respect to the model performance. Typically for most of the prognostic field this compromise is reached by interpolation based on cubic polynomials (Staniforth and Côté, 1991).

The ALADIN/ARPEGE/IFS code uses for the purpose of the semi-Lagrangian "accurate" interpolation sort of 2D and 3D interpolatores based on Lagrange cubic polynomial in destinations close to the target point and linear interpolation for the outer sphere (Yessad, 2004). This interpolation works with sufficient efficiency and computational cost (Ritchie et al.,1995). However its performance is limited by the performance of the Lagrange cubic interpolatores, which tends to be sometimes too stiff when applied to rapidly changing quantity (field with dominating small scale features). This known limitation is already on current operational scales for some fields too restrictive. Hence for example vertical interpolation for prognostic ozone can be optionally performed by spline (IFS) or by Hermite interpolatores (ARPEGE/Climat). However the use of those higher order interpolations is restricted just to the vertical direction and prognostic ozone.

Since the semi-Lagrangian horizontal diffusion (SLHD) becomes a model feature, it has implicitly raised a need for more precise semi-Lagrangian interpolation. When SLHD is activated the original semi-Lagrangian interpolator is selectively corrupted by additional diffusive interpolation. Consequently the conservative ability of the advection scheme is deteriorated. It is known feature that since the inability to conserve total mass the semi-Lagrangian models produces generally slight positive bias of surface pressure (Gravel and Staniforth, 1994) This tendency is typically small enough that especially for the purpose of NWP it can be ignored. The presence of SLHD further enhances the positive MSLP bias tendency caused by the semi-Lagrangian scheme.




This feature is illustrated by the figure displaying MSLP bias (lower lines) and rmse (upper lines) signals computed from 19 days parallel test of ALADIN/LACE with SLHD (red colour) and without (black colour). The signal is not really dramatic and as proven it is even not further cumulated with an continuous assimilation cycle using the results of previous runs as the background fields for the new analyses. Once again the conclusion can be that we can live with slightly worse model performance in the term of mass conservation having the benefit of relatively cheap non-linear horizontal damping.

The aim of a NWP research should be always the interest to improve a model performance keeping all good features of a code rather then to replace one advantage by another (even when it seems to be a good deal). This would imply a need to search for a possibility to reduce the above mentioned side weakness of SLHD. Studying the structure of the semi-Lagrangian interpolator with activated SLHD it has been concluded that just something like between 0 and 15 % of the whole interpolated amount is obtained by the diffusive interpolator. This is not a big contribution which means that there is not really much freedom to further reduce the portion of the diffusive interpolator while keeping the same diffusive properties of the SLHD. Logically the most promising way seems to improve the performance of the accurate interpolator. When the more precise interpolator will be contaminated by the diffusive interpolation, the total performance ideally should be around the performance of the current Lagrange cubic interpolator which is generally considered as sufficient for the NWP purpose.

Of course a new high order interpolation should not be much more expensive with respect to the current one. Otherwise the scheme will not be competitive with the original one. Other constraint for the new potential interpolator specific to the ALADIN/ARPEGE/IFS model is the ability of the interpolatores to be evaluated locally allowing computers to use the profit of a parallel computation.

# 2 Spline interpolation in ALADIN/ARPEGE/IFS

Keeping previous restrictions new interpolator was designed for semi-Lagrangian interpolation in the ALADIN/ ARPEGE/IFS model. It has been designed in the exactly same way like the current high order interpolator (the order of computation, the interpolation grid) with the only difference that the Lagrange cubic interpolations are replaced by cubic interpolatores with smooth first derivative and continuous second derivative. This definition fits the definition of splines. The new interpolator then can be considered as spline on four points.

## 2.1 A bit of theory

The general spline interpolation formula can be written as (Press et al.,1986):

$$y = Ay_i + By_{i+1} + Cy_i^{''} + Dy_{i+1}^{''} \quad , \tag{1}$$

where $A$ and $B$ are the weights for linear interpolation ($B = 1 - A$) and

$$C \equiv \frac{1}{6}(A^3 - A)(x_{i+1} - x_i)^2 \quad ,$$

$$D \equiv \frac{1}{6}(B^3 - B)(x_{i+1} - x_i)^2 \quad .$$

Here the $x_i$ are grid points coordinates with corresponding known values of an interpolated amount $y_i$.

The unknown values for second derivatives $y_i^{''}$ are obtained by using the condition for continuity of first derivatives. For a given $N$ points it gives set of $N - 2$ equations:

$$\frac{x_i - x_{i-1}}{6}y_{i-1}^{''} + \frac{x_{i+1} - x_{i-1}}{3}y_i^{''} + \frac{x_{i+1} - x_i}{6}y_{i+1}^{''} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \quad . \tag{2}$$

To complete this system for $N$ variables the values for $y_1^{''}$ and $y_N^{''}$ has to be defined. The simplest solution used also for our purpose is to define so-called natural spline by setting

$$y_1^{''} = y_N^{''} = 0 \quad .$$

## 2.2 ALADIN/ARPEGE/IFS implementation

As already mentioned for the semi-Lagrangian interpolation in ALADIN/ARPEGE/IFS the "local" approach of spline is used so the $N$ will be always equal to $4$. In such case with the natural spline boundary condition the equation (2) can be reduced to system of two equation for the two unknowns $y_2^{''}$ and $y_3^{''}$:

$$\begin{aligned} \frac{x_3 - x_1}{3}y_2^{''} + \frac{x_3 - x_2}{6}y_3^{''} &= \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1} \\ \frac{x_3 - x_2}{6}y_2^{''} + \frac{x_4 - x_2}{3}y_3^{''} &= \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \end{aligned} \tag{3}$$

This set is always diagonally dominant when $x_2 - x_1 \neq 0$ and $x_4 - x_3 \neq 0$ which is always the case with the model grid. This means that a solution always exists as:

$$y_2^{''} = \frac{C_1 B_2 - C_2 B_1}{A_1 B_2 - A_2 B_1} \quad ,$$

$$y_3^{''} = \frac{A_1 C_2 - C_1 A_2}{A_1 B_2 - A_2 B_1} \quad ,$$

where:

$$A_1 = \frac{x_3 - x_1}{3} \qquad B_1 = \frac{x_3 - x_2}{6} \qquad C_1 = \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1} \quad ,$$

$$A_2 = \frac{x_3 - x_2}{6} \qquad B_2 = \frac{x_4 - x_2}{3} \qquad C_2 = \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \quad .$$

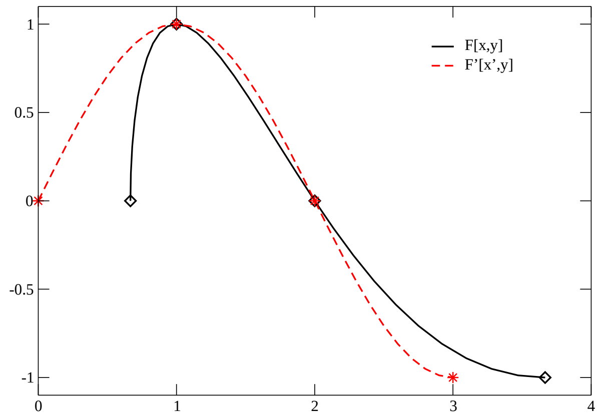



Figure 1: *The true function F to be interpolated on stretched model grid fitting the model gridpoints (black full line with the points marked as diamonds) and the equivalent function F' transformed to regular grid which is interpolated instead along vertical during spline interpolation (red dashed curve with the points marked as stars). The target area to be interpolated is located between ipoints 1 and 2 on x-axis.*

The horizontal interpolation can be fairly simplified by interpolating virtual function F'[$x_i^{'}$,$y_i$] instead of F[$x_i$,$y_i$]. Here the $x_i$ stands for general model computational grid while $x_i^{'}$ represents virtual regular grid ($x_{i+1}^{'} - x_i^{'} \equiv 1$). In case of ALADIN grid $x_i^{'} \equiv x_i$, hence F $\equiv$ F'. The two functions F and F' are illustrated by figure 1. This trick is applied just to horizontal mesh since here the computational grid distribution is controlled by some rules (gauss grid, stretching) ensuring that the derivatives of an interpolated amount on the virtual grid would still somehow correspond with the computational grid. Vertical grid is determined by namelist without any apriori restriction, so the interpolation is performed on the real grid along this direction. Fortunately the vertical interpolation is performed just once at the end of the 3D interpolation, so it is not causing a dramatic increase of the model computational cost.

The equations (3) will have then for horizontal interpolation solution:

$$y_2^{''} = \frac{2}{5}\left(4y_1 - 9y_2 + 6y_3 - y_4\right)$$

$$y_3^{''} = \frac{2}{5}\left(-y_1 + 6y_2 - 9y_3 + 4y_4\right)$$

In this case also the computation of the coefficients $C$ and $D$ in (1) can be simplified to just:

$$C = \frac{1}{6}(A^3 - A)$$

$$D = \frac{1}{6}(B^3 - B)$$

Thanks to this simplification the computation of this interpolator is just around 2.8% more expensive than the less exact Lagrange cubic interpolation.[1] There is still some space this performance further optimise, but it is questionable how much it would improve the final performance (i.e. it can happen that a lot of code-work will improve this performance by just negligible factor).

### 2.3 User's guide

To switch the current Lagrange high order interpolation to the one with splines in the ALADIN/ARPEGE/IFS model is quite simple. A set of **NAMDYN** namelist switches called LRSPLINE_[X] for separate variable (kind of variables) is defined to activate (when set to `.TRUE.`) the spline interpolation:

| | |
|---|---|
| `LRSPLINE_W` | for horizontal flow components |
| `LRSPLINE_T` | for temperature |
| `LRSPLINE_SPD` | for (NH) pressure departure |
| `LRSPLINE_SVD` | for (NH) vertical divergence |
| `LRSPLINE_P` | for continuity equation |
| `LRSPLINE_Q` | for moisture |
| `LRSPLINE_O3` | for ozone |
| `LRSPLINE_V` | for other GFL fields |

*(Note that for ozone this spline interpolation has higher priority than quasi-monotone vertical spline interpolation (IFS) which both have higher priorities than vertical Hermite interpolatores (ARPEGE/Climat.)*
All the other features of the semi-Lagrangian interpolatores like quasi-monotonicity (keys LQM[X]), horizontal quasi-monotonicity (keys LQMH[X]) or SLHD (key LSLHD) are preserved independently to the actual value of LRSPLINE_[X]. The defaults values for cycle CY29T1 are `.FALSE.`, only in case of SLHD are LRSPLINE_W, LRSPLINE_T, LRSPLINE_Q and in case of NH dynamics LRSPLINE_SPD with LRSPLINE_SVD automatically set to `.TRUE.`.

# 3 Performance of the splines

So what should one expect from the splines used instead of the Lagrange cubic interpolatores despite some increase of CPU time consumption? Surely it is an improvement of the advection scheme interpolation precision reducing a model random damping. This effect is reflected by the figure 2 showing the response of interpolatores to the kinetic energy spectra during the academic frontogenetic idealised adiabatic 3D experiment with the model ALADIN (Váňa, 2003). It is evident, that when spline is used instead of Lagrange interpolatores, the inherent diffusion of the semi-Lagrangian scheme is reduced especially for the small scale information.

The figure 3 shows the mean quadratic error of several interpolatores with respect to the waves of model spectrum (with quadratic truncation). As it can be seen the Lagrange interpolation is outperforming the others for the long waves. Once the interpolated quantity becomes more rapidly changing, the spline interpolator fulfilling the additional conditions for derivatives starts to interpolate with smaller error. This makes the spline interpolator especially profitable when some rapidly changing fields (of small scale character) will be advected

Anyway some extensive validation to prove the response of the spline interpolation in term of increase of the computational precision of the model still has to be done. Up to now no parallel test focused to this new model feature has been launched. Hence currently we can just speculate about an possible improvement of the model scores.

Other reason to use the splines in the ALADIN/ARPEGE/IFS model was linked to the SLHD diffusion. To prove clear profit from the existence of spline interpolator for this case is relatively easy. As shown on the figure 4 the SLHD creates a systematic positive bias of the surface pressure. When the accurate interpolator of

[1]This result was obtained with operational ARPEGE TL359L41c2.4 on Fujitsu VPP5000 and spline interpolation used for u,v,T,q but not $\pi_s$

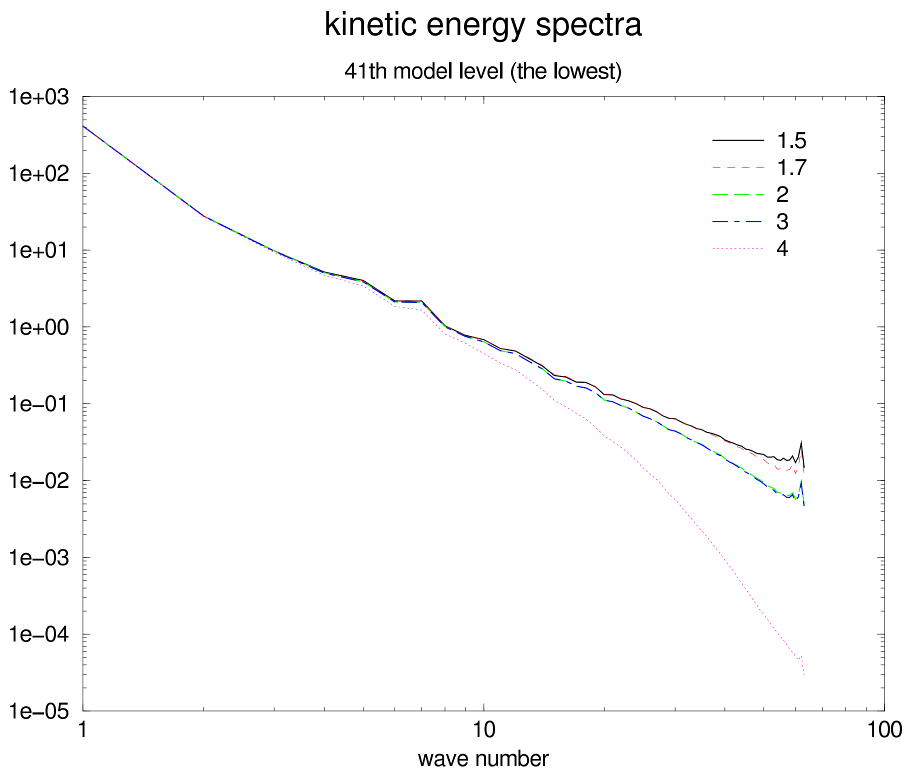



Figure 2: *The kinetic energy spectra from the idealised adiabatic frontal development simulated with the model ALADIN as a result of used interpolatores for semi-Lagrangian scheme. Black full line (1.5) represents the spline interpolation used for all interpolation (N[X]LAG=2), dashed red line (1.7) represents the same with the small difference that the tendency part of the interpolated amount is interpolated by linear interpolation (N[X]LAG=3). Long dashed green line (2) is representing the spectra obtained when Lagrange cubic interpolator is used exclusively for whole s-L amount (N[X]LAG=2), dot-dashed blue line (3) represents the Lagrange interpolator used for fields while the tendency are interpolated by linear interpolation (N[X]LAG=3). Finally dotted violet line (4) represents the result after just linear interpolation.*

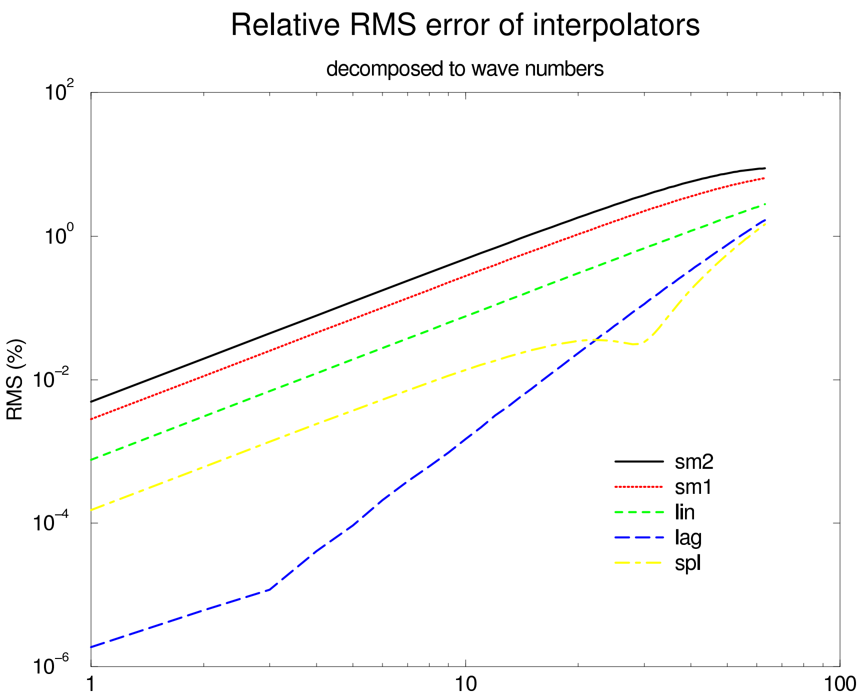



Figure 3: *Mean quadratic error (as the percentage of the wave amplitude) of different interpolatores obtained as the result of $10^6$ 1D interpolation of separate waves from a model spectrum with quadratic truncation. The curves represents following interpolatores:* `sm2` *- average of 4 adjacent points,* `sm1` *- average of 2 adjacent points,* `lin` *- linear interpolation,* `lag` *- cubic Lagrange interpolation and* `spl` *- natural cubic spline on 4 points.*

SLHD is represented by the spline interpolator, the positive bias tendency is significantly reduced (the portion of areas with warm - yellow and red - colours is reduced). This reflect the ability of the more precise spline interpolator to reduce the bias caused by semi-Lagrangian interpolation.

# 4 Conclusion

Since CY29T1 the spline interpolation is available as an alternative to the other interpolations for semi-Lagrangian scheme advection. This interpolation tends to be more precise than the default Lagrange interpolator especially for the fields with the dominating small scale character. To achieve such increase of accuracy, one has to expect increase of the model CPU consumption by around of 3 % of the total performance. Contrary to the other alternatives to the default interpolator this new interpolator can be used with all model fields being advected by semi-Lagrangian scheme.

Another advantage of the more precise spline interpolator is its ability to reduce systematic MSL pressure bias. Since the SLHD produces the opposite effect it is especially useful to combine SLHD with this new kind of interpolator.

Moreover even when it is preferably constructed for the regular mesh, the spline interpolation is introduced in very general way. It means that its usage is not restricted by SLHD or by other interpolation switches. It can be combined with any other constraint for semi-Lagrangian scheme (quasi-monotonicity, N[X]LAG,...)

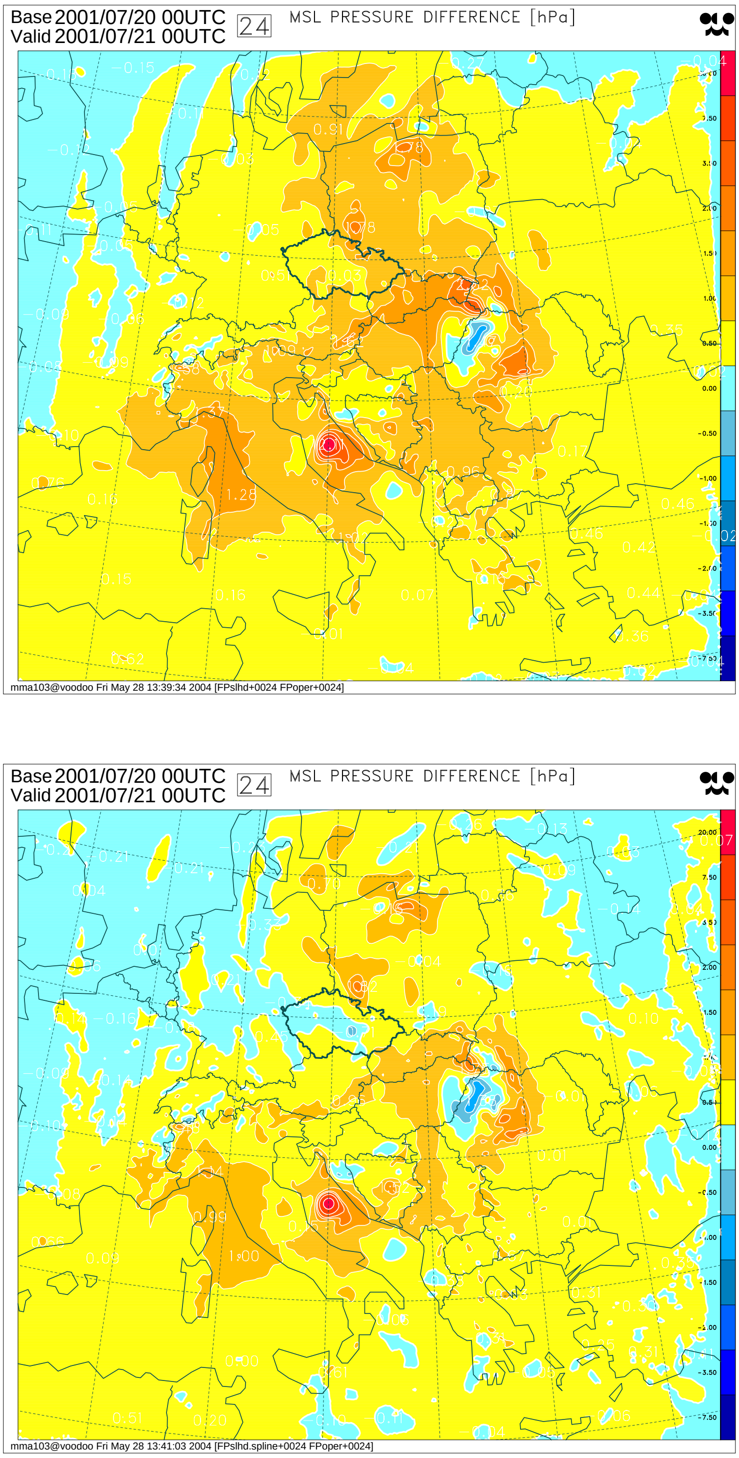



Figure 4: *The MSL pressure difference of 24 hours forecast of ALADIN/LACE with SLHD compared to the spectral diffusion as reference (***upper figure***) and with SLHD + spline interpolation compared to the same reference (spectral diffusion with default Lagrange interpolatores) (***bottom figure***).*

# 5 References


Gravel S. and Staniforth A. - A Mass-Conserving Semi-Lagrangian Scheme for the Shallow-Water Equations.
*Monthly Weather Review* **Vol. 122**, No. 1, pp. 243-248.

Press W.H.,Teukolsky S.A., Vetterling W.T. and Flannery B.P. - Numerical Recipes in Fortran 77 - The Art of Scientific Computing. Second edition.
*Cambridge University Press*, 1986, 1992.

Ritchie H., Temperton C., Simmons A., Hortal M., Davies T., Dent D. and Hamrud M. - Implementation of the semi-Lagrangian method in a high-resolution version of the ECMWF forecast model.
*Mon. Wea. Rev.***Vol. 123**, February 1995, pp. 489–514.

Staniforth A. and Côté, J - Semi-Lagrangian integration schemes for atmospheric models - - a review.
*Mon. Wea. Rev.* **Vol. 119**, September 1991, pp. 2206–2223.

Váňa F. - Semi-Lagrangian advection scheme with controlled damping - an alternative way to nonlinear horizontal diffusion in a numerical weather prediction model.
*PhD thesis*, Charles University Prague, 2003.
*In Czech with extended English and French abstracts*

Yessad K. - Semi-Lagrangian computations in the cycle 29 of ARPEGE/IFS.
*Meteo-France/CNRM/GMAP/ALGO internal memo*, 2004.