A conservative reconstruction scheme for the interpolation of extensive quantities in the Lagrangian particle dispersion model FLEXPART

Sabine Hittmeir[1,2], Anne Philipp[2], and Petra Seibert[3]

[1]Faculty of Mathematics, University of Vienna, Vienna, Austria
[2]Department of Meteorology and Geophysics, University of Vienna, Vienna, Austria
[3]Institute of Meteorology, University of Natural Resources and Life Sciences, Vienna, Austria

Correspondence to: Sabine Hittmeir (sabine.hittmeir@univie.ac.at)

Abstract. Lagrangian particle dispersion models require interpolation of all meteorological input variables to the position in space and time of computational particles. The widely-used model FLEXPART uses linear interpolation for this purpose, implying that the discrete input fields contain point values. As this is not the case for precipitation (and other fluxes) which represent cell averages or integrals, a preprocessing scheme is applied which ensures the conservation of the integral quantity with the linear interpolation in FLEXPART, at least for the temporal dimension. However, this mass conservation is not ensured per grid cell, and the scheme thus has undesirable properties such as temporal smoothing of the precipitation rates. Therefore, a new reconstruction algorithm was developed, in two variants. It introduces additional supporting grid points in each time interval and is to be used with a piecewise-linear interpolation to reconstruct the precipitation time series in FLEXPART. It fulfils the desired requirements by preserving the integral precipitation in each time interval, guaranteeing continuity at interval boundaries, and maintaining non-negativity. The function values of the reconstruction algorithm at the sub-grid and boundary grid points constitute the degrees of freedom which can be prescribed in various ways. With the requirements mentioned it was possible to derive a suitable piecewise-linear reconstruction. To improve the monotonicity behaviour, two versions of a filter were also developed that form a part of the final algorithm. Currently, the algorithm is meant primarily for the temporal dimension. It was shown to significantly improve the reconstruction of hourly precipitation time series from three-hourly input data. Preliminary considerations for the extension to additional dimensions are also included as well as suggestions for a range of possible applications beyond the case of precipitation in a Lagrangian particle model.

1 Motivation

In numerical models, extensive variables (those being proportional to the volume or area that they represent, e. g. as mass and energy) are usually discretised as grid-cell integral values so that conservation properties can be fulfilled. A typical example is the precipitation flux in a meteorological forecasting model. Usually, one is interested in the precipitation at the surface, and thus the quantity of interest is a two-dimensional horizontal integral (coordinates x and y) over each grid cell. Furthermore, it is accumulated over time t during the model run, and written out at distinct intervals together with other variables, such as

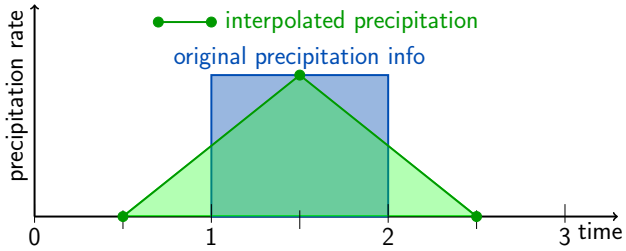

Figure 1. Illustration of the basic problem using an isolated precipitation event lasting one time interval represented by the thick blue line. The amount of precipitation is given by the blue-shaded area. Simple discretisation would use the green circles as discrete grid-point representation and interpolate linearly in between, indicated by the green line and the green-shaded area. Note that supporting points for the interpolation are shifted by a half time interval compared to the times when other meteorological fields are available.

wind, temperature, etc. After the trivial postprocessing step of de-accumulation, each value thus represents an integral in x, y, t space – the total amount of precipitation which fell on a discrete grid cell during a discrete time interval.

In Lagrangian particle dispersion models (LPDMs) (Lin et al., 2013), it is necessary to interpolate the field variables obtained from a meteorological model to particle positions in (three-dimensional) space and in time. The simplest option, to assign the value corresponding to the grid cell in which the particle resides (often called "nearest-neighbour" method), is not sufficiently accurate. For example, in the case of precipitation, this would lead to an unrealistic, checkerboard-like deposition field. A simple approach for improvement would be to ascribe the gridded precipitation value to the spatio-temporal centre of the space-time cell and then perform linear interpolation between these points, as illustrated in Fig. 1. While this works for the case of intensive field quantities (velocity, temperature, etc.) where gridded data truly represent point values, it is not a satisfactory solution for extensive quantities, as it does not conserve the total amount in each time interval, as will be shown below. The problem became obvious to us with respect to the precipitation field in the LPDM FLEXPART (Stohl et al., 2005). Therefore, it will be discussed using this example, even though the problem is of a general nature and the solution proposed has a wide range of possible applications.

1.1 Introduction of the problem at the example of precipitation in FLEXPART

FLEXPART is a Lagrangian particle dispersion model (LPDM), which is typically applied to study air pollution, but is also used for other problems requiring the quantification of atmospheric transport, such as the global water cycle or the exchange between the stratosphere and the troposphere, see Stohl et al. (2005). Before a FLEXPART simulation can be done, a separate preprocessor is used to extract the meteorological input data from the Meteorological Archival and Retrieval System (MARS) of the European Centre for Medium Range Weather Forecasts (ECMWF) and prepare them for use in the model.[1] (The model is also able to ingest meteorological fields from the United States National Center for Environmental Prediction. However, it was originally designed for ECMWF fields and we limit our discussion here to this case.) Currently, a relatively simple method

[1]This software is available from the FLEXPART community website in different versions as described in https://www.flexpart.eu/wiki/FpInputMetEcmwf.

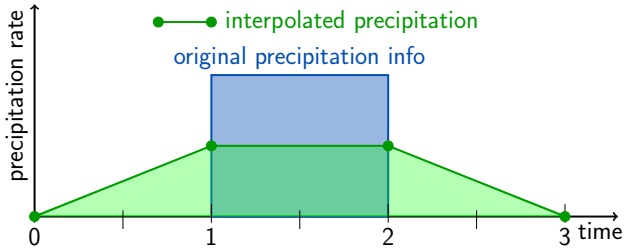

Figure 2. Example of so-called 'disaggregation' of precipitation data for the use in FLEXPART as currently implemented, with the case of an isolated precipitation period. Note that the supporting points for the interpolation now coincide with the times when other meteorological fields are available. Colours are used as in Fig. 1.

processes the precipitation fields in a way that is consistent with the scheme applied in FLEXPART for all variables, linear interpolation between times where input fields are available. Under these conditions, it is not possible to conserve the original amount of precipitation in each grid cell. The best option, which was realised in the preprocessing, is conservation within the interval under consideration plus the two adjacent ones. Unfortunately, this leads to undesired temporal smoothing of the precipitation time series – maxima are damped and minima are raised. It is even possible to produce non-zero precipitation in dry intervals bordering a precipitation period as shown in Fig. 2. This is obviously undesirable as it will affect wet scavenging, a very efficient removal process for many atmospheric trace species. Wet deposition may be produced in grid cells where none should occur, or too little in others. Different versions of the FLEXPART data extraction software refer to this process as 'disaggregation' or 'de-accumulation'.

Horizontally, the precipitation values are averages for a grid cell around the grid point to which they are ascribed, and FLEXPART uses bilinear interpolation to obtain precipitation rates at particle positions. This causes the same problem of spreading out the information to the neighbouring grid cells and implied smoothing.[2] However, the supporting points in space are not shifted between precipitation and other variables as it is the case for the temporal dimension.

The goal of this work is to develop a reconstruction algorithm for the one-dimensional temporal setting which

- strictly conserves the amount of precipitation within each single time interval,
- preserves the non-negativity,
- is continuous at the interval borders,
- and ideally also should reflect a natural precipitation curve. This latter condition can be understood in the sense that the reconstruction graph should possess good monotonicity properties.

[2]In reality, the problem is even more complex. In ECMWF's MARS archive, variables such as precipitation are stored on a reduced Gaussian grid, and upon extraction to the latitude-longitude grid they are interpolated without paying attention to mass conservation. This needs to be addressed in the future on the level of the software used internally by the MARS system. Our discussion here is assuming that this would already have happened, and even if that is not the case, adding another step of non-mass-conserving interpolation makes things even worse.

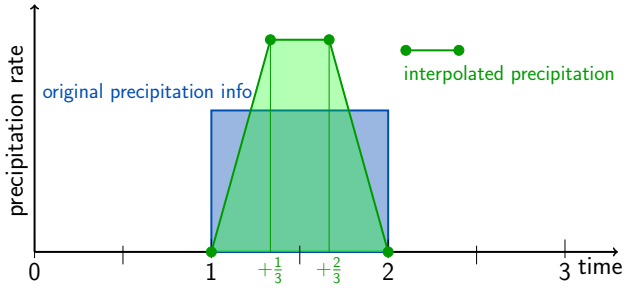

Figure 3. Precipitation rate linearly interpolated using a sub-grid with two additional points. Colours as in Fig. 1.

– Furthermore, it should be easy and efficient to implement within the existing framework of the FLEXPART code and its data extraction preprocessor.

These requirements on the reconstruction algorithm imply that time intervals with no precipitation remain unchanged, i. e. the reconstructed values vanish throughout this whole time interval, too. In the simplest scenario of an isolated precipitation event, where in the time interval before and after the data values are zero, the reconstruction algorithm therefore has to vanish at the boundaries of the interval, too. The additional conditions of continuity and conservation of the precipitation amount then require to introduce sub-grid points if we want to keep a linear interpolation (Fig. 3). The height thereby is determined by the condition of conservation of the integral of the function over the time interval. The motivation for a linear formulation arises from the last point of the above list of desirable properties.

It can be noted that in principle a single sub-grid point per time interval would be sufficient. This, however, would result in very high function values and steep slopes of the reconstructed curve, which appears to be less realistic and thus not desirable.

As we shall see in the next section, closing the algorithm for such isolated precipitation events is quite straightforward, since the only degree of freedom constituting the height of the reconstruction function is determined by the amount of precipitation in the interval. However, the situation becomes much more involved if longer periods of precipitation occur, i. e., several consecutive time intervals with positive data. Then, in general, each sub-grid function value constitutes one degree of freedom (Fig. 4).

Therefore, in order to close the algorithm, we have to fix all of these additionally arising degrees of freedom. As a first step we make a choice for the slope in the central subinterval, which relates the two inner sub-grid function values. Three possible approaches are discussed for this choice. Conservation provides a second condition. These two can be considered to determine the two inner sub-grid points. Then, the function values at the grid points in between time intervals of positive data are left to be prescribed, and as each point belongs to two intervals, this corresponds to the third degree of freedom. The steps leading to the final algorithm (of which there are some variants) are presented in Sect. 3. Ways for extending the one-dimensional algorithms to two-dimensional setting are briefly discussed as well. Note that we use the wording ‘reconstruction algorithm’ and ‘interpolation algorithm’ interchangeably.

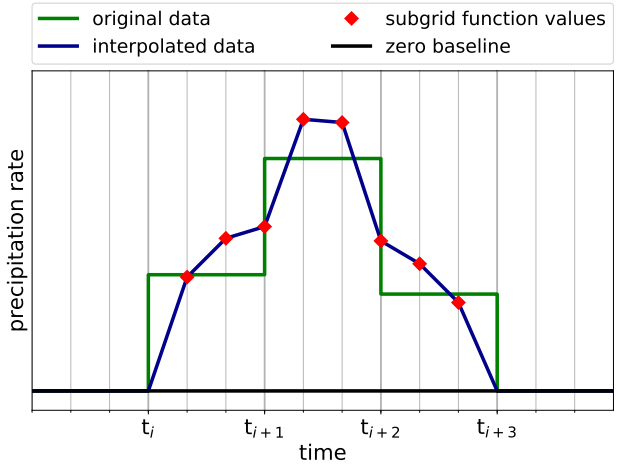

Figure 4. Illustration of a reconstruction for longer periods with positive data values, where each sub-grid function value constitutes one degree of freedom.

In the following Sect. 2, some existing literature on conservative reconstruction algorithms is briefly reviewed, with emphasis on applications used for semi-Lagrangian advection schemes in Eulerian models. To our knowledge, confirmed by contacts with people active in this field, a piecewise-linear, conservative reconstruction algorithm using a sub-grid has not yet been proposed.

Section 4 then presents an evaluation of the algorithms by verifying them with synthesised data and validating them with original data from the ECMWF where the available 1-hourly resolution serves as a reference data set and the 3-hourly resolution as input for the interpolation algorithms. The verification demonstrates that the demanded requirements are indeed fulfilled and the validation compares the accuracy of the new algorithms with the currently used one to show the improvements.

The conclusions (Sect. 5) summarise the findings, present an outlook for the next steps, and suggest a range of possible applications of the new reconstruction algorithm beyond the narrow case of precipitation input to a Lagrangian particle dispersion model.

2 Possible approaches and related literature

A widely-used form of interpolation is the well-known spline interpolation consisting of piecewise polynomials, which are typically chosen as cubic ones (e. g., Hämmerlin and Hoffmann, 1994; Hermann, 2011). The task of finding an appropriate piecewise polynomial interpolation, which is non-negative, continuous, mass conserving and monotonic is a challenging one. This fact has also been pointed out for example by White et al. (2009), where it is stated that building a reconstruction profile satisfying all these requirements is generally not possible. Typically, the reconstruction profiles are not continuous on the edges of the grid cells. Sufficient conditions for positive spline interpolation in form of inequalities on the interpolation's coefficients have been derived in Schmidt and Heß (1988).

The issue of mass-conservative interpolation emerge also in the context of semi-Lagrangian finite-volume advection schemes, which have become very popular. These schemes, with a two-dimensional application in mind, are known under the heading of "remapping". Eulerian grid cells are mapped to the respective areas covered in the previous time step, and then the mass in this area is calculated by a reconstruction function from the available grid-cell average values (Lauritzen et al., 2010). Apart from global interpolation functions such as Fourier methods, piecewise defined polynomials are the method of choice in this context. They can be piecewise constant, linear, parabolic (second-order) or cubic (third-order). The first two options are usually discarded as not being accurate enough. While this application shares the need for the reconstruction to be conservative and positive definite, and the aim of preserving monotonicity, with our problem there are some aspects where the characteristics of the problems are different. For the advection schemes, continuity at the cell interface is not a strict condition. However, they need to be able to reconstruct sharp peaks inside each volume, as otherwise through the repetitive application during the numerical integrations, strong artificial smoothing of sharp structures would result. Therefore, at least second-order and if possible higher-order methods are preferred. The drawback of higher-order reconstruction functions is their tendency to overshoot and produce wiggles which have to be removed or reduced by some sort of filter. However, as in the remapping process one always integrates over some domain, this is less of an issue than in our case, where for each computational particle we need an interpolated value at exactly one point.

An interesting example of such a semi-Lagrangian conservative remapping is given by Zerroukat et al. (2002). The coefficients of the one-dimensional cubic spline in each interval are determined using the conservation of mass in the underlying interval. The function values at the left border points are determined by an additional spline interpolation using the condition of mass conservation in the two preceding, the current and the following intervals. The function values at the right border points are determined in a similar fashion. This construction in particular provides continuity. A monotonic and positive definite filter has then been applied to this semi-Lagrangian scheme (Zerroukat et al., 2005a), which first detects regions of non-monotonic behaviour, and then locally reduces the order of the polynomial until monotonicity is regained. An improved version of this filter without reducing the order of the interpolating polynomial in most cases is provided in Zerroukat et al. (2005b), where a parabolic spline is used for interpolation. The basic algorithm in all these cases is one-dimensional, where the application to the two-dimensional case has been explicitly demonstrated only in Zerroukat et al. (2005a), as a combination of the so-called cascade splitting and the one-dimensional algorithm.

Considering the mentioned differences between the reconstruction problem arising in the context of semi-Lagrangian advection schemes and of the LPDM FLEXPART, and in addition that linear interpolation is used in FLEXPART for all other quantities and that evaluation of the interpolation function has to be done efficiently for up to millions of particles in each time step, we have chosen to construct a non-negative, continuous and conservative reconstruction algorithm based upon piecewise-linear interpolation. Contrary to standard piecewise-linear methods, we divide each grid interval into three subintervals, so that our method has some similarity with a piecewise parabolic approach while being simpler and presumably faster.

3 Derivation of the interpolation algorithm

3.1 Notation and basic requirements

In accordance with the considerations presented in Sect. 1, we consider our input data to be precipitation values defined over a period $[0,T]$, where T is the time at the end of the period. They are available as amounts (or equivalently, as average precipitation rates) during $N-1$ constant time intervals of duration Δt, bounded by equidistant times t_i where

$$t_i = i\,\Delta t\,, \quad i \in \{0,\ldots,N\}. \tag{1}$$

The time intervals for which the precipitation amounts are defined are denoted as

$$I_i = (t_i, t_{i+1}] = (i\Delta t, (i+1)\,\Delta t] \quad \text{for} \quad i \in \mathcal{I} := \{0,\ldots,N-1\}\,. \tag{2}$$

The precipitation rate is then represented as a step function $g : [0,T] \to \mathbb{R}$ with values

$$g(t) = g_i \quad \text{for} \quad t \in I_i\,,\ i \in \mathcal{I}. \tag{3}$$

As an abbreviation, we write $g(0) = g_0$. The total precipitation within one time interval I_i is then given by

$$\int_{I_i} g\,\mathrm{d}t = g_i\,\Delta t\,. \tag{4}$$

Our aim is to find a piecewise-linear function $f : [0,T] \to \mathbb{R}$ to serve as interpolation. We require it

1. to be continuous,
2. to preserve the non-negativity such that f satisfies

$$f \geq 0\,, \text{and} \tag{5}$$

3. to conserve the precipitation amount within each single time interval I_i, i. e.

$$\int_{I_i} f\,\mathrm{d}t = g_i \Delta t \tag{6}$$

In particular,

$$g_i = 0 \quad \Leftrightarrow \quad f(t) = 0 \quad \text{for} \quad t \in I_i. \tag{7}$$

These conditions are also listed in Table 2 as the three strict and main requirements of the algorithm. They necessitate the introduction of sub-grid points. A single sub-grid point was deemed insufficient for a realistic representation of precipitation time series. For simplicity, we choose an equidistant sub-grid setting with two additional points

$$t_i^{(1)} = t_i + \frac{1}{3}\,\Delta t \quad \text{and} \quad t_i^{(2)} = t_i + \frac{2}{3}\,\Delta t = t_{i+1} - \frac{1}{3}\,\Delta t \quad \text{for} \quad i \in \mathcal{I}\,. \tag{8}$$

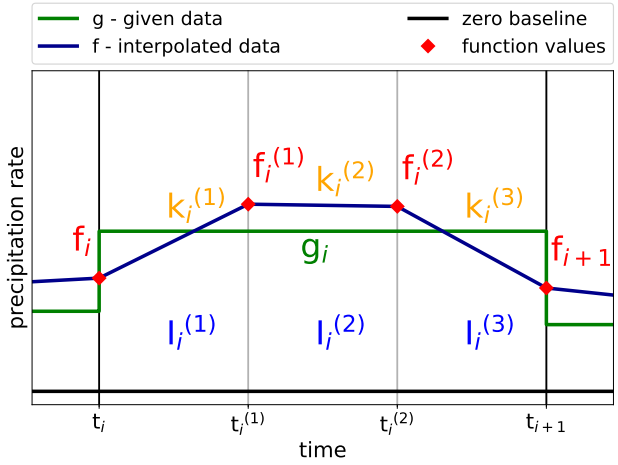

Figure 5. Schematic overview of the basic notation in a precipitation interval with the original precipitation rate g (green) as a step function and the interpolated data f (dark blue) as the piecewise-linear function. The original time interval with fixed grid length Δt is split equidistantly in three subintervals denoted by $I_i^{(1,2,3)}$, with the slopes in the subintervals as denoted by $k_i^{(1,2,3)}$. The sub-grid function values $f_i, f_i^{(1,2)}, f_{i+1}$ are marked by red diamonds.

The subintervals resulting from these sub-grid points are defined as $I_i^{(1)} = (t_i, t_i^{(1)}]$, $I_i^{(2)} = (t_i^{(1)}, t_i^{(2)}]$, $I_i^{(3)} = (t_i^{(2)}, t_{i+1}]$ and the slopes of the interpolation algorithm f in these subintervals are denoted accordingly by $k_i^{(1)}$, $k_i^{(2)}$, $k_i^{(3)}$. The sub-grid function values are abbreviated in the following as $f_i := f(t_i)$, $f_i^{(1)} := f(t_i^{(1)})$, $f_i^{(2)} := f(t_i^{(2)})$. Figure 5 shows a schematic overview of these definitions.

It is evident that the function f in I_i is uniquely determined by the function values f_i, $f_i^{(1)}$, $f_i^{(2)}$, f_{i+1}, with linear interpolation between them. Equivalently, the problem can be stated in terms of the slopes, such that the function f in I_i is determined uniquely by $f_i, k_i^{(1)}, k_i^{(2)}, k_i^{(3)}$. This equivalence is based upon the relations between slopes and function values:

$$f_i^{(1)} = f_i + \frac{1}{3} k_i^{(1)} \Delta t, \qquad f_i^{(2)} = f_i^{(1)} + \frac{1}{3} k_i^{(2)} \Delta t, \qquad f_{i+1} = f_i^{(2)} + \frac{1}{3} k_i^{(3)} \Delta t. \tag{9}$$

The key requirement for the interpolation algorithm f is to preserve precipitation amount within each single time interval I_i as specified in Eq. (6). Therefore,

$$P_i = \int_{I_i} f \, \mathrm{d}t = \int_{I_i} g \, \mathrm{d}t \quad \text{for} \quad \forall i \in \mathcal{I}, \tag{10}$$

or, equivalently, equal areas underneath function graphs of f and g. In the following, we thus refer to Eq. (10) also as the equal-area condition. In terms of the function values, Eq. (10) amounts (after division by Δt) to

$$g_i = \frac{1}{6}\left(f_i + f_{i+1} + 2(f_i^{(1)} + f_i^{(2)})\right) = \frac{1}{6} f_i + \frac{2}{6} f_i^{(1)} + \frac{2}{6} f_i^{(2)} + \frac{1}{6} f_{i+1}. \tag{11}$$

In order to fulfil Eq. (5) (non-negativity) we need a solution satisfying

$$f_i, f_i^{(1)}, f_i^{(2)} \geq 0 \quad \text{for} \quad \forall i \in \mathcal{I}. \tag{12}$$

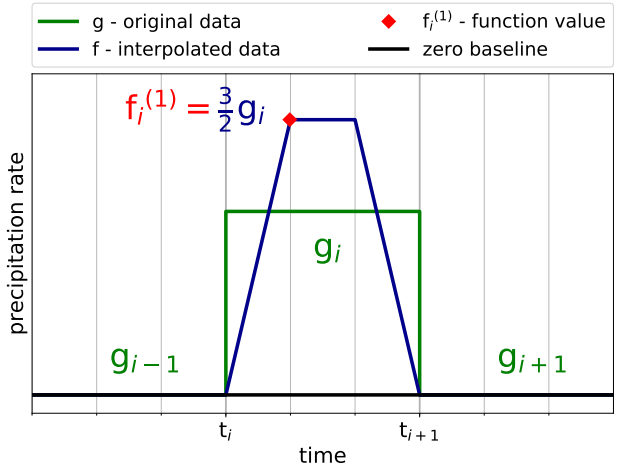

Figure 6. Isolated precipitation event (no precipitation in g_{i-1} and g_{i+1}). The only degree of freedom is given by the function value $f_i^{(1)} = \frac{3}{2} g_i$ and is marked by a red square.

As already mentioned above, a further consequence of the equal-area condition and the continuity condition is in particular

$$g_i = 0 \quad \Rightarrow \quad f_i = f_i^{(1)} = f_i^{(2)} = f_{i+1} = 0\,, \tag{13}$$

such that periods with zero precipitation rate remain unchanged.

3.2 Isolated precipitation in a single time interval

We first demonstrate the basic idea of the interpolation algorithm for the simplest case of an isolated precipitation event, i. e. we assume an interval $i \in \mathcal{I}$ with $g_i > 0$ and $g_{i-1} = g_{i+1} = 0$ (see Fig. 6). As Eq. (13) then holds in the surrounding intervals I_{i-1} and I_{i+1}, the continuity condition yields $f_i = f_{i+1} = 0$ at the boundary of the interval. Moreover, as we do not want to create artificial asymmetry in the problem, we let f be constant in the centred subinterval $I_i^{(2)}$, such that the only function value left to be determined is $f_i^{(1)} = f_i^{(2)}$, which constitutes thus the only degree of freedom in the problem. This height of the interpolation function is now obtained via the equal-area condition Eq. (11), which in this particular case amounts to

$$f_i^{(1)} = \frac{3}{2} g_i\,, \tag{14}$$

therewith closing the algorithm.

3.3 General case

Whereas the derivation of the algorithm for the isolated precipitation event is straightforward, the problem becomes considerably more involved if consecutive intervals with non-zero precipitation occur. Treating each interval as an isolated precipitation event as demonstrated in Fig. 6 by forcing the function values at the original grid points to vanish would be possible, but such

an algorithm is not acceptable, as the interpolation function f has to reflect the actual course of precipitation in a realistic way. All function values f_i inbetween periods with non-zero precipitation should be positive, too. Thus, they constitute additional degrees of freedom and need to be determined. The main challenge for a suitable interpolation algorithm lies in finding a proper way to deal with these additional degrees of freedom.

Therefore, we now consider the case of two consecutive intervals with non-zero data $g_i\, g_{i+1} > 0$ for some $i \in \mathcal{I}$. The first function value f_i is assumed to be given by the algorithm in the preceding interval I_{i-1}, or – for the first interval – a prescribed boundary value. Then, since $g_{i+1} > 0$, we neither have the condition of vanishing f_{i+1}, nor is the symmetry relation $f_i^{(1)} = f_i^{(2)}$ desirable generally. Thus, with given f_i, in general there are the three degrees of freedom $f_i^{(1)}$, $f_i^{(2)}$, f_{i+1}, associated with the interval I_i, as illustrated in Fig. 4. Since the equal-area condition corresponds to only one of them, two additional relations are required.

3.3.1 Boundary conditions

In the following, we assume the boundary values

$$f_0 := f(0) \quad \text{and} \quad f_N := f(T) \tag{15}$$

as being prescribed. If their values are not explicitly provided, a simple assumption is to consider the precipitation rate to be constant in time and thus $f(0) = g(0)$ and $f(T) = g(T)$, or, equivalently, $f_0 = g_0$ and $f_N = g_{N-1}$.

3.3.2 Prescribing the central slope

As a means to reflect the actual course of precipitation, a natural first step is to prescribe the central slope $k_i^{(2)}$. We choose it as the average of the slopes in the outer two subintervals, i. e.

$$k_i^{(2)} = \frac{k_i^{(1)} + k_i^{(3)}}{2}\,, \tag{16}$$

which has the desirable property that it allows for the particular case $k_i^{(1)} = k_i^{(2)} = k_i^{(3)}$. Moreover, inserting Eq. (9), we obtain the equivalent expression

$$k_i^{(2)} = \frac{f_{i+1} - f_i}{\Delta t}\,. \tag{17}$$

This result is quite intuitive in the sense that it corresponds to the mean slope of the interpolation function throughout the interval I_i. Letting $k_i^{(2)}$ being determined via Eq. (17), the function values $f_i^{(2)}$ are uniquely determined by $f_i^{(1)}$ through Eq. (9) as

$$f_i^{(2)} = f_i^{(1)} + \frac{1}{3} k_i^{(2)} \Delta t = f_i^{(1)} + \frac{1}{3}(f_{i+1} - f_i)\,, \tag{18}$$

and thus the degrees of freedom are reduced accordingly.

Other possible approaches for the central slope which have not been selected would be:

(i) Setting $k_i^{(2)} = 0$, which is the simplest choice for $k_i^{(2)}$. It was used for the isolated precipitation event. This means that f is constant in the central subintervals $I_i^{(2)}$, and thus $f_i^{(1)} = f_i^{(2)}$. This choice, however, does not reflect a natural precipitation curve.

(ii) A more advanced, data-driven approach would be to represent the tendency of the surrounding data values by the centred finite difference

$$k_i^{(2)} = \frac{g_{i+1} - g_{i-1}}{2\Delta t}. \tag{19}$$

The problem here is to fulfil the condition of non-negativity if g_i is small compared to one of its neighbouring values.

3.3.3 Using the equal-area condition

Now, the function values $f_i^{(1)}$ are determined in a way that the equal-area condition in Eq. (11) is satisfied, which, after inserting Eq. (18) yields

$$g_i = \frac{1}{18}(5f_{i+1} + f_i) + \frac{2}{3}f_i^{(1)}. \tag{20}$$

We thus obtain for the sub-grid function values

$$f_i^{(1)} = \frac{3}{2}g_i - \frac{1}{12}f_i - \frac{5}{12}f_{i+1}, \tag{21}$$

$$f_i^{(2)} = \frac{3}{2}g_i - \frac{5}{12}f_i - \frac{1}{12}f_{i+1}. \tag{22}$$

3.3.4 Closing the algorithm under the condition of non-negativity

Equations Eq. (21) and Eq. (22) show that the algorithm is closed once the function values at the grid points f_{i+1} are determined. Thus, the function values f_{i+1} for indices $i \in \mathcal{I}$ with $g_i\, g_{i+1} > 0$ still need to be determined. An obvious first choice would be to use the arithmetic mean value of the surrounding data values g_i and g_{i+1}. However, in order to fulfil Eq. (13), a case distinction is required to deal separately with $g_{i+1} > 0$ and $g_{i+1} = 0$ for a given $g_i > 0$. This would lead to a lack of continuity between the cases of precipitation equal to or only close to zero. For the latter case, the algorithm would even produce negative values. Therefore, the arithmetic mean is not a good choice. A better choice is the geometric mean

$$f_{i+1} = \sqrt{g_i\, g_{i+1}} \quad \text{for} \quad i \in \mathcal{I}, \tag{23}$$

which has the main advantage that the case distinction is not required. With the additional Eq. (23) for the function value f_{i+1}, having Eq. (21) and Eq. (22) for the sub-grid values $f_i^{(1)}$ and $f_i^{(2)}$, respectively, the algorithm is now closed. However, the problem of negative values still can arise in the case of small values in between larger ones. This is due to the fact that for $g_{i+1} \to 0$, the geometric mean $\sqrt{g_i\, g_{i+1}}$ converges to 0 in general too slowly.

A further possible approach would be to assign $f_{i+1} = \min\{g_i, g_{i+1}\}$ with $k_i^{(2)} = 0$, which fulfils the non-negativity, but gives a non-monotonic solution curve and thus is not producing a natural precipitation curve. A less restrictive approach

would be to use $f_{i+1} = \min\{f_i^{(2)}, f_{i+1}^{(1)}\}$. However, this would also lead to one slope being artificially set to zero, and thus be incompatible with a realistic course of the precipitation. Furthermore, this solution would be implicit, as the interval I_i depends on the solution in I_{i+1}. Since the relation is via a minimum function, one would have to distinguish between all possible cases of function values in the whole interval of precipitation, probably a too complex operation for longer periods.

We shall note that instead of prescribing the function values at the grid points directly, there are also other possible approaches. Two of them have been looked at and are discussed in the following paragraphs.

Instead of a function value, we might prescribe an additional slope. We tested a basic finite difference approach in terms of the involved data values as well as a symmetric version of it. As this does not preseve monotonicity, we also derived a global algorithm, where the slope of the right subinterval in I_i is equal to the slope in the left subinterval of the next time interval. Thus, the solution in the time interval I_i depends on the solution in the next time interval I_{i+1}, such that the algorithm cannot be solved for each time interval individually anymore, but has to be solved globally in form of a linear system. This algorithm was shown to create better monotonicity properties, but the implementation is much more complicated and solving the problem is clearly computationally much more expensive. All of these algorithms, however, have a common fault, namely the violation of non-negativity. This is caused by the fact that the algorithms with prescribed slopes all rely on a case distinction, whether one involved precipitation value is positive or not, and therefore are not continuous with respect to vanishing values. One should also note that algorithms based on prescribing additionally the slopes $k_i^{(3)}$ for $g_i\, g_{i+1} > 0$ are not invariant with respect to being solved forward or backward in time. This results from the asymmetry of the slopes, since when solving backwards in time, the roles of the slopes $k_i^{(3)}$ and $k_i^{(1)}$ from the original problem are interchanged.

Another approach would be to formulate the reconstruction problem as an optimisation problem. However, for large data sets this turned out to be much more expensive than the ad-hoc methods described before (using the MATLAB Optimisation Toolbox). As the final aim is to solve the interpolation problem for large data sets in three dimensions, this approach was not further studied.

The preservation of non-negativity is a challenging requirement, as discussed above. In the following, we investigate sufficient conditions for the non-negativity to hold. The algorithm consisting of Eqs. (21), (22), and (23) (function values f_{i+1} determined via the geometric mean) is considered as the base. It has the strong advantage not to require a case distinction between vanishing and positive values. It shall be combined with the minimum value approach which guarantees the non-negativity of the solutions. We thus now investigate which conditions on generally prescribed non-negative function values f_i are required to guarantee that also the sub-grid function values are non-negative, i. e. $f_i^{(1)} \geq 0 \wedge f_i^{(2)} \geq 0$. For the central slope being defined as the mean Eq. (17), the sub-grid function values are given by Eq. (21) and Eq. (22). The requirement of non-negativity of $f_i^{(1)}$ and $f_i^{(2)}$ then amounts to

$$18\, g_i \geq \max\{f_i + 5f_{i+1}, 5f_i + f_{i+1}\}\,. \tag{24}$$

Thus, a sufficient condition for the algorithm to preserve non-negativity is the restriction

$$f_{i+1} \leq 3\min\{g_i, g_{i+1}\}, \quad \forall i \in \mathcal{I}\,. \tag{25}$$

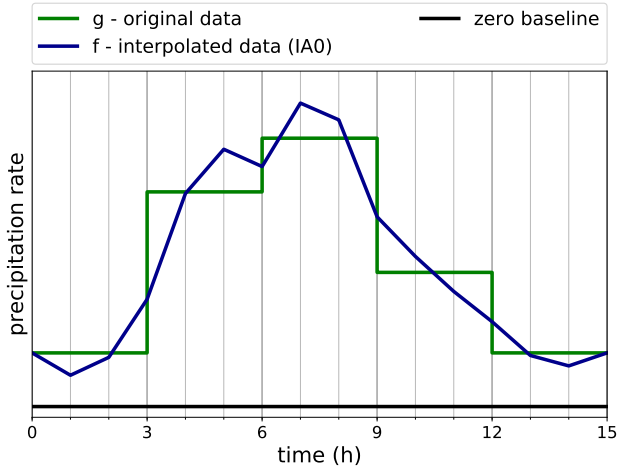

Figure 7. Results with the IA0 algorithm. The original precipitation rate g is shown in green with a 3-hourly resolution. The IA0 interpolated data f on the new sub-grid with 1-hourly resolution is given in dark blue. A zero baseline is shown in black.

The same condition for non-negativity is obtained if the central slope is prescribed as zero, $k_i^{(2)} = 0$ (option (i) in Sect. 3.3.2). With the data-driven definition Eq. (19) for the central slope, option (ii), the additional restriction

$$18\, g_i \leq |g_{i+1} - g_{i-1}| \tag{26}$$

would have to be fulfilled by the input data. However, this relation is often violated in realistic precipitation data, which justifies the decision to discard this approach.

We thus return to the geometric mean Eq. (23) method and combine it with the non-negativity constraint Eq. (25) which results in

$$f_{i+1} = \min\{3g_i, 3g_{i+1}, \sqrt{g_i g_{i+1}}\}\,. \tag{27}$$

We have thus obtained a piecewise-linear interpolation function determined by Eq. (27), (21) and (22) which defines a non-negative, continuous and area-preserving algorithm, called *Interpolation Algorithm 0* (IA0) henceforth.

3.3.5 Monotonicity filter as a postprocessing step

Figure 7 illustrates the IA0 algorithm with a practical example. It fulfils the mentioned requirements, but it appears to be not sufficiently realistic, as it does not preserve the monotonicity as present in the input data (minimum at $t = 6$ h).

In response to this problem, we introduce a monotonicity filter which is active only in the regions where the graph of f resembles the shape of an 'M' or 'W', or where

$$\mathrm{sgn}(k_i^{(2)}) \cdot \mathrm{sgn}(k_i^{(3)}) = -1 \quad \wedge \quad \mathrm{sgn}(k_i^{(3)}) \cdot \mathrm{sgn}(k_{i+1}^{(1)}) = -1 \quad \wedge \quad \mathrm{sgn}(k_{i+1}^{(1)}) \cdot \mathrm{sgn}(k_{i+1}^{(2)}) = -1\,. \tag{28}$$

In this case, we replace the function value f_{i+1} by

$$f_{i+1}^{\text{mon}} = \min\left\{3g_i, 3g_{i+1}, \sqrt{(f_{i+1}^{\diamond} f_{i+1}^{\diamond\diamond})_+}\right\} \tag{29}$$

where now $f_{i+1}^{\diamond}$ and $f_{i+1}^{\diamond\diamond}$ are the values at t_{i+1} obtained by taking either $k_i^{(3)} = 0$ in I_i, starting from f_i, or $k_{i+1}^{(1)} = 0$ in I_{i+1}, ending at f_{i+2} respectively. Since these values can become negative, we just set negative values to zero, as indicated by the shorthand notation $F_+ := \max\{F, 0\}$ for some value F. The function values f_i and f_{i+2} thereby remain unchanged while the sub-grid function values in the intervals I_i and I_{i+1} are recomputed accordingly to satisfy the equal-area condition (see Fig. 8).

More precisely, given f_i, we compute f_{i+1} by setting $k_i^{(3)} = 0$, further on denoted by $f_{i+1}^{\diamond}$. With Eq. (9), this amounts to $f_i^{(2)} = f_{i+1}^{\diamond}$. According to Eq. (22), the function value $f_{i+1}^{\diamond}$ becomes

$$f_{i+1}^{\diamond} = \frac{18}{13} g_i - \frac{5}{13} f_i \,. \tag{30}$$

On the other hand, given f_{i+2}, the function value $f_{i+1}^{\diamond\diamond}$ corresponds to the one obtained from setting $k_{i+1}^{(1)} = 0$, such that $f_{i+1}^{(1)} = f_{i+1}^{\diamond\diamond}$ (with Eq. (9)) while leaving f_{i+2} unchanged. Then, from Eq. (21) it follows that

$$f_{i+1}^{\diamond\diamond} = \frac{18}{13} g_{i+1} - \frac{5}{13} f_{i+2} \,. \tag{31}$$

The monotonicity filter then recomputes $f_i^{(1)}, f_i^{(2)}, f_{i+1}^{(1)}, f_{i+1}^{(2)}$ according to Eq. (21) and Eq. (22) with f_{i+1} being replaced by f_{i+1}^{mon} from Eq. (29). The interpolation algorithm which uses the monotonicity filter as a postprocessing step henceforth is called *Interpolation Algorithm 1* (IA1) and is summarised in Table 1.

3.3.6 Alternative monotonicity filter yielding a single-sweep algorithm

It is also possible to construct an algorithm which directly incorporates the idea from the monotonicity filter introduced above. In order to apply the filter in a single sweep, we need a kind of educated guess for f_{i+2} as this appears in Eq. (31). We estimate it similar to Eq. (29) as

$$\widetilde{f}_{i+2} = \min\{3g_{i+1}, 3g_{i+2}, \sqrt{g_{i+1}\, g_{i+2}}\} \tag{32}$$

where the tilde indicates that it is a preliminary estimate. We can now proceed analogously to above by constructing

$$f_{i+1}^{\diamond} = \frac{18}{13} g_i - \frac{5}{13} f_i \,, \tag{33}$$

$$f_{i+1}^{\diamond\diamond} = \frac{18}{13} g_{i+1} - \frac{5}{13} \widetilde{f}_{i+2} \,, \tag{34}$$

and determine f_{i+1} as

$$f_{i+1} = \min\left\{3g_i, 3g_{i+1}, \sqrt{(f_{i+1}^{\diamond} f_{i+1}^{\diamond\diamond})_+}\right\}, \quad \text{for} \quad i \in \mathcal{I}\,, \tag{35}$$

respecting again the sufficient condition for non-negativity. Having obtained f_{i+1}, the sub-grid function values in I_i are determined as before by Eq. (21) and Eq. (22), thus closing the algorithm. This improved version of the interpolation algorithm, with the monotonicity filter directly built in, is called *Interpolation Algorithm 2* (IA2) henceforth and is summarised in Table 1. It applies the filter to all the intervals rather than to 'M'- or 'W'-shaped parts of the graph only, as it is the case in IA1.

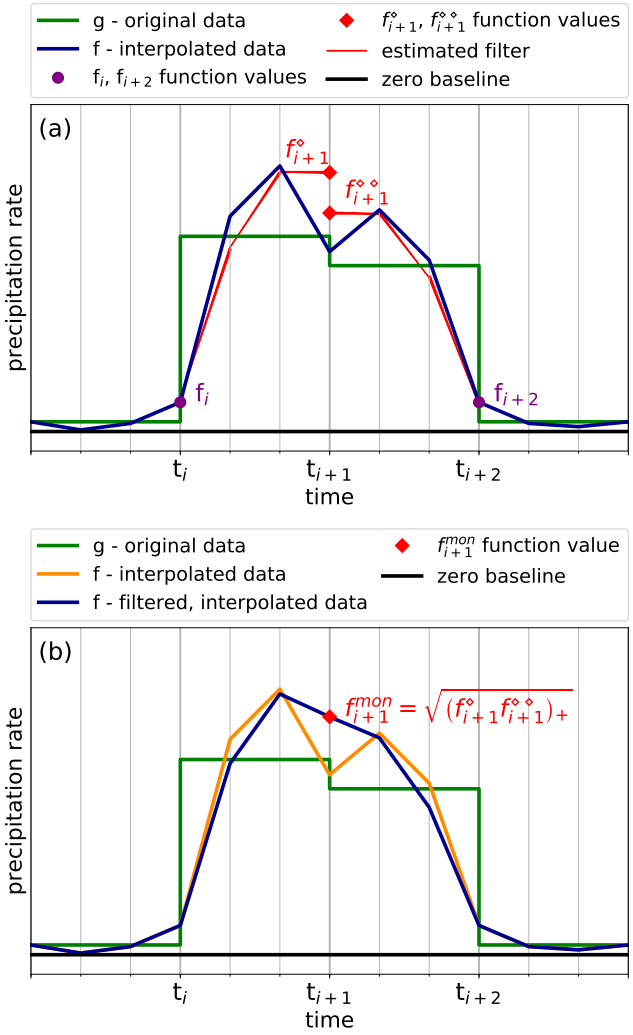

Figure 8. Illustration of the monotonicity filter construction. The original precipitation rate g is shown in green with a 3-hourly resolution. The interpolated series f from IA0, using the new sub-grid with 1-hourly resolution, is given in dark blue. A zero baseline is shown in black. **(a)** At first, the function value f_{i+1} is split in $f^{\diamond}_{i+1}$ and $f^{\diamond\diamond}_{i+1}$ shown as red squares. Other function values are recomputed as shown with the red line where f_i and f_{i+2} remain unchanged as shown with purple circles. **(b)** Second, the function value f_{i+1} is substituted by the new function value f^{mon}_{i+1} as marked with a red square. The unfiltered graph f derived by IA0 is shown in orange here, while the filtered graph resulting after recomputation of the neighbouring values is shown in dark blue.

3.4 Summary of the interpolation algorithms IA1 and IA2

Three interpolation algorithms, IA0, IA1 and IA2, were developed. They were introduced on an additional sub-grid based on the geometric mean and fulfil the conditions to be non-negative, continuous and area-conserving. The basic algorithm is called IA0. A monotonicity filter was then introduced to improve the realism of the reconstructed function. The IA1 algorithm

Table 1. Overview of the two algorithms IA1 and IA2. The second calculation of f_{i+1} in algorithm IA1 corresponds to the equation of f_{i+1}^{mon}. The superscript "mon" in the original form of Eq. (29) is omitted here for simplicity.

	IA1	Ref.		IA2	Ref.
$f_i^{(1)} =$	$\frac{3}{2}g_i - \frac{1}{12}f_i - \frac{5}{12}f_{i+1}$	(21)	$f_i^{(1)} =$	$\frac{3}{2}g_i - \frac{1}{12}f_i - \frac{5}{12}f_{i+1}$	(21)
$f_i^{(2)} =$	$\frac{3}{2}g_i - \frac{5}{12}f_i - \frac{1}{12}f_{i+1}$	(22)	$f_i^{(2)} =$	$\frac{3}{2}g_i - \frac{5}{12}f_i - \frac{1}{12}f_{i+1}$	(22)
$f_{i+1} =$	$\min\{3g_i, 3g_{i+1}, \sqrt{g_i g_{i+1}}\}$	(27)	$\widetilde{f}_{i+2} =$	$\min\{3g_{i+1}, 3g_{i+2}, \sqrt{g_{i+1}\, g_{i+2}}\}$	(32)
if	$\mathrm{sgn}(k_i^{(2)}) \cdot \mathrm{sgn}(k_i^{(3)}) = -1 \quad \wedge$	(28)			
	$\mathrm{sgn}(k_i^{(3)}) \cdot \mathrm{sgn}(k_{i+1}^{(1)}) = -1 \quad \wedge$				
	$\mathrm{sgn}(k_{i+1}^{(1)}) \cdot \mathrm{sgn}(k_{i+1}^{(2)}) = -1$ **then**				
$f_{i+1}^{\diamond} =$	$\frac{18}{13}g_i - \frac{5}{13}f_i$	(30)	$f_{i+1}^{\diamond} =$	$\frac{18}{13}g_i - \frac{5}{13}f_i$	(33)
$f_{i+1}^{\diamond\diamond} =$	$\frac{18}{13}g_{i+1} - \frac{5}{13}f_{i+2}$	(31)	$f_{i+1}^{\diamond\diamond} =$	$\frac{18}{13}g_{i+1} - \frac{5}{13}\widetilde{f}_{i+2}$	(34)
$f_{i+1} =$	$\min\left\{3g_i, 3g_{i+1}, \sqrt{(f_{i+1}^{\diamond} f_{i+1}^{\diamond\diamond})_+}\right\}$	$(29)_*$	$f_{i+1} =$	$\min\left\{3g_i, 3g_{i+1}, \sqrt{(f_{i+1}^{\diamond} f_{i+1}^{\diamond\diamond})_+}\right\}$	(35)
$f_i^{(1)} =$	$\frac{3}{2}g_i - \frac{1}{12}f_i - \frac{5}{12}f_{i+1}^{\mathrm{mon}}$	(21)			
$f_i^{(2)} =$	$\frac{3}{2}g_i - \frac{5}{12}f_i - \frac{1}{12}f_{i+1}^{\mathrm{mon}}$	(22)			
endif					

requires a second sweep through the data, while IA2 has a monotonicity filter already built into the main algorithm. The equations defining IA1 and IA2 are listed in Table 1, and Fig. 9 illustrates all three with an example. The algorithms were realised in Python and can be downloaded from the supplementary material.

3.5 The two-dimensional case

We have also carried out a preliminary investigation of the two-dimensional case. In the case of precipitation, this could be used for horizontal interpolation. We follow the same approach and introduce a sub-grid with two additional grid points, now for both directions.

The isolated two-dimensional precipitation event can then easily be represented on the sub-grid as a truncated pyramid. For multiple adjacent cells with non-zero data, this type of interpolation is, however, not suitable due to the non-vanishing values at the boundaries of the grids which would be difficult to formulate.

A more advantageous approach is the bilinear interpolation, which defines the function in a square uniquely through its four corner values. (Note that we assume the grid spacing equal in both directions, without loss of generality, as this can always be achieved by simple scaling). The main idea here is to apply the bilinear interpolation in each of the nine sub-squares. We recall that for given function values $F(X_i, Y_j) := F_{ij}$, with $i, j \in \{1, 2\}$, at the corners of an area $A = [X_1, X_2] \times [Y_1, Y_2]$, the bilinear interpolation amounts to

$$F(X,Y) \;=\; \frac{Y_2 - Y}{Y_2 - Y_1}\left(\frac{X_2 - X}{X_2 - X_1}F_{11} + \frac{X - X_1}{X_2 - X_1}F_{21}\right) + \frac{Y - Y_1}{Y_2 - Y_1}\left(\frac{X_2 - X}{X_2 - X_1}F_{12} + \frac{X - X_1}{X_2 - X_1}F_{22}\right), \tag{36}$$

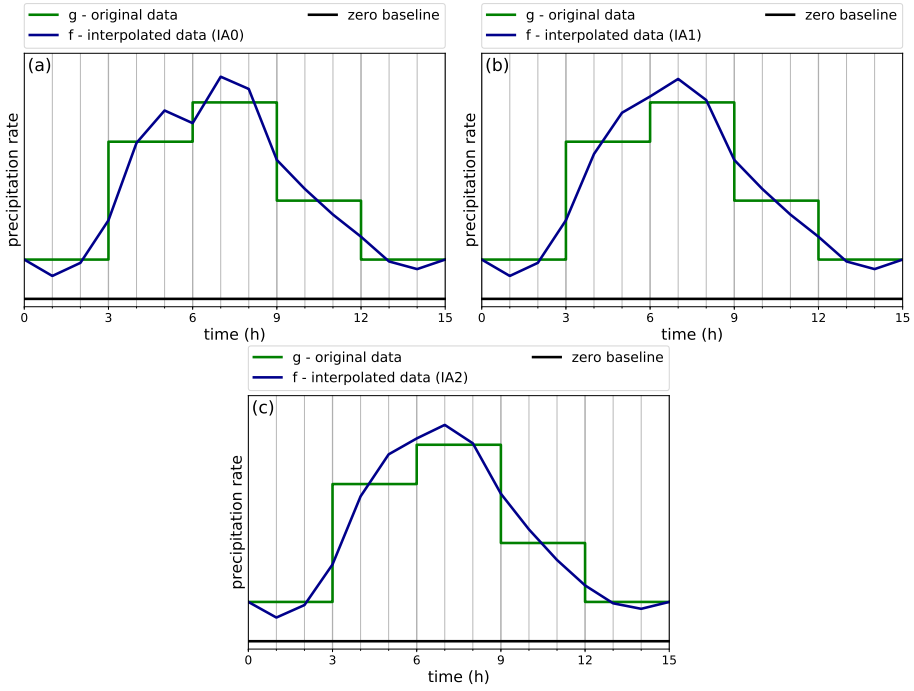

Figure 9. Illustration of the basic IA0 algorithm in **(a)**, with the additional monotonicity filter in algorithm IA1 in **(b)** and the directly implemented filter in the algorithm IA2 in **(c)**. The original precipitation rate g is shown in green with a 3-hourly resolution. The IA0 interpolated data f on the new sub-grid with 1-hourly resolution is given in dark blue. A zero baseline is shown in black.

which corresponds to interpolating first in X at $Y = Y_1$ and $Y = Y_2$ and then performing another interpolation in Y (or vice versa). Thus, the algorithm is closed if the 16 sub-grid function values in each grid cell are known, where again only one is determined by the conservation of mass. The case of the isolated precipitation event with vanishing boundary values is again easily solved, since only the corner values of the centred sub-square with constant height need to be determined, amounting to one degree of freedom. This in particular demonstrates that the bilinear interpolation algorithm is a natural extension from the one-dimensional case, since the function value in the centred sub-square of such an isolated precipitation event turns out to be

$$f_{ij}^0 = \frac{9}{4} g_{ij} = \left(\frac{3}{2}\right)^2 g_{ij} \,. \tag{37}$$

For the general case involving larger precipitation areas many different possibilities for prescribing the slopes and function values arise at the sub-grid points. The geometric-mean based approach can be extended to the two-dimensional setting with corresponding restrictions guaranteeing non-negativity. The derivation of a full solution for the two-dimensional case is reserved for future work.

Table 2. Classification of requirements for the interpolation algorithm. They are classified into strict requirements (stRE), which are essential and need to be fulfilled, and soft requirements (soRE), which are desirable but not absolutely necessary.

	Requirements
stRE1	Mass shall be conserved in each single time interval.
stRE2	The interpolated function shall preserve non-negativity.
stRE3	The boundary transitions shall be continuous.
soRE1	The interpolated function shall remain monotonic where input data are.
soRE2	Symmetric structures shall remain symmetric.
soRE3	The interpolated curve shall be realistic and accurate.
soRE4	The algorithm shall be computationally efficient and easy to implement into the existing framework of the FLEXPART model.

4 Evaluation of interpolation algorithms

The evaluation of the new algorithms IA1 and IA2 was carried out in three steps. At first, the interpolation algorithms were applied to ideal, synthetic time series to verify the fulfilment of the requirements. Next, they were validated with ECMWF data. Short sample sections were analysed visually. The main validation is then based on statistical metrics. The original algorithm from the ECMWF data extraction (flex_extract) for FLEXPART was also included in the evaluation. In the following, it is referred to as *Interpolation algorithm FLEXPART* (IFP). This allows to see and quantify the improvements through the new algorithms. The IFP is not published, but it is included in the flex_extract download on the FLEXPART web site (http://flexpart.eu/) and a Python version of it is included in the Supplementary Material.

4.1 Verification of algorithms with synthetic data

Verification is the part of evaluation where the algorithm is tested against the requirements to show whether it is doing what it is supposed to do. These requirements, mentioned in the previous sections, are classified into strict requirements (main conditions, stRE) and soft requirements (soRE), as formulated in Table 2.

The synthetic time series for the first tests is specified with 3-hourly resolution. It consists of four isolated precipitation events, with constant precipitation rates during the events and durations which increase from one to four 3-h intervals. As the variation within each 3-h interval is unknown, it is visualised as a step function. We refer to it as the *Synthesised 3-hourly* (S3h) time series. Both new algorithms IA1 and IA2 and the currently used IFP were applied to these data. The IFP produces 3-hourly disaggregated output which is divided into 1-hour segments by the usual linear interpolation between the supporting points. As all three algorithms are intended to be used in connection with linear interpolation, they are visualised by connecting the resulting supporting points with straight lines. Figure 10 shows the input data set together with the results from the reconstruction algorithms.

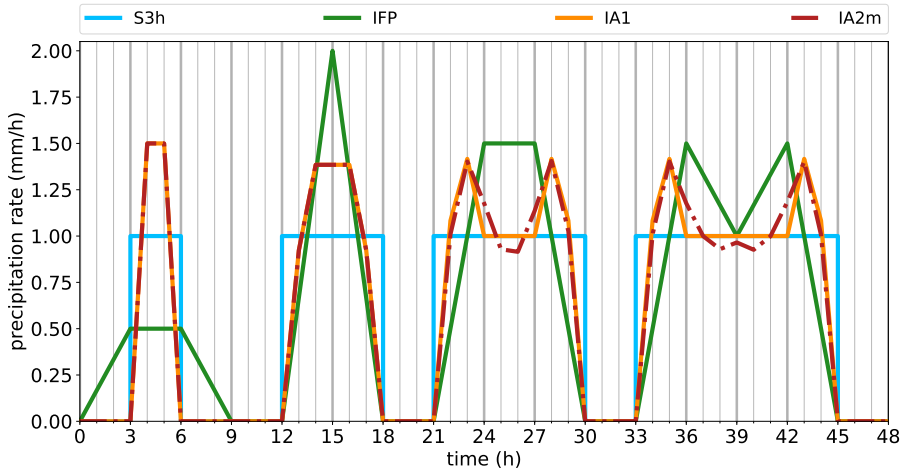

Figure 10. Verification of the interpolation algorithms for four simple precipitation events. The 3-hourly synthetic precipitation rate (S3h) is illustrated as a step function in light blue. Reconstructions are shown as linear connections of their respective supporting points, with the current FLEXPART algorithm (IFP) in green, newly developed algorithm IA1 in orange, and IA2 (also new) in red (dashed-dotted).

It is easy to see that IFP violates requirement stRE1 (cf. Table 2): the mass of the first precipitation event is spread over three intervals instead of one in the first event. This leads to a reduction of the precipitation intensity in the originally rainy interval while precipitation appears also in adjacent, originally dry intervals, constituting the basic problem introduced in Sect. 1. The second event is the only one in this synthetic time series where IFP conserves mass within each interval. The peak value constructed is twice the input precipitation rate. In the third event, having a duration of three times 3-h, mass is shifted from the two outer intervals into the middle one. Similarly, in the fourth event, mass is shifted from the border to the central intervals, however, here two local maxima separated by a local minimum are created. In all these events, IFP produces a non-negative and continuous reconstruction, hence, requirements stRE2 and stRE3 are fulfilled. Concerning the soft requirements, the monotonicity condition (soRE1) may appear to be violated in the fourth event. However, some overshooting is necessary unless an instantaneous onset of the precipitation at the full rate is postulated. The symmetry condition (soRE2) is fulfilled. Requirements soRE3 and soRE4 cannot be tested well with this short and simple case.

Concerning the behaviour of the two newly developed algorithms IA1 and IA2, we clearly see in Fig. 10 that no mass is spread from the wet intervals into the dry neighbourhood. For a strict verification of local mass conservation (stRE1), we compared the integral values in each interval of the interpolated time series and S3h numerically and found that mass is conserved perfectly for both algorithms. Moreover, non-negativity (stRE2) as well as the continuity requirement (stRE3) are also fulfilled.

With respect to soRE1 (monotonicity), we are faced with the already-mentioned overshooting behaviour. In the events lasting three and four intervals, the new algorithms introduce a local minimum in the centre of the event. As directly intelligible, it is not possible for the interpolated curve to turn into a constant value without overshoot. This would either lead to excess mass in the

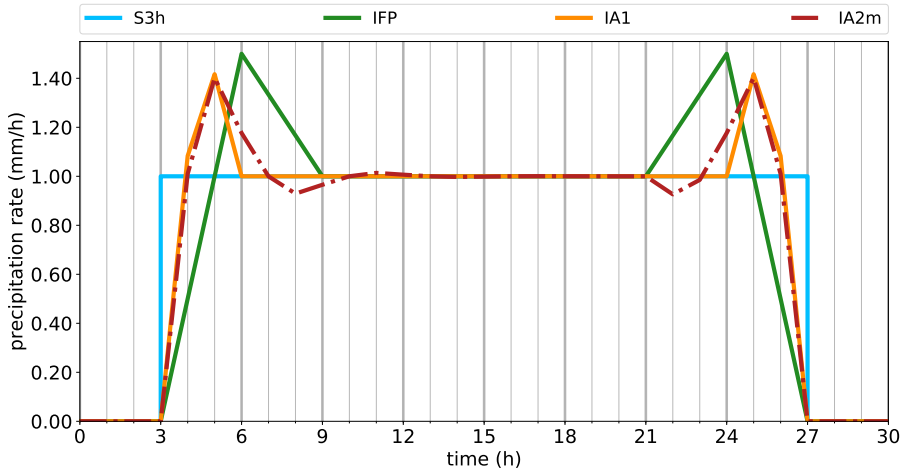

Figure 11. Verification of the different behaviour of the interpolation algorithms for a longer constant precipitation event plotted in $\mathrm{mm\,h^{-1}}$. The 3-hourly synthesised precipitation rate (S3h) is illustrated as a step function in light blue, the old interpolation algorithm of FLEXPART (IFP) in green and the newly developed algorithms IA1 and IA2 are shown in orange and red (dashed-dotted), respectively.

inner period as seen in the IFP algorithm, or to a lack in the outermost periods. Obviously, interpolated curves have to overshoot to compensate the gradual rise (or fall, respectively) near the borders of precipitation periods. While IA1 accomplishes this within a single three hour interval, algorithm IA2 falls off more slowly towards the middle with the consequence of requiring another interval on each side for compensation. In order to investigate how these wiggles would develop in an even longer event, a case with eight constant values, lasting 24 hours, was constructed (Fig. 11). It shows that the amplitude of the wiggles in IA2 falls off rapidly.

The symmetry condition (soRE2) is satisfied by IA1 but not by IA2. The wiggles in the 24-hour event (Fig. 11) spread beyond the second rainy interval on the left, but not on the right side. A tiny asymmetry is visible also in Fig. 10. The reason for this difference between IA1 and IA2 is the way in which the monotonicity filter is applied. In the IA1 algorithm, the filter is only applied if the curve is 'M'- or 'W'-shaped (Eq. 28), while in IA2 the filter is applied for each time interval. We would see in IA1 exactly the same behaviour as in IA2 if we would remove the condition of Eq. (28). In the context of investigating symmetry, we have also run the reconstruction algorithms in the reverse direction. This produced identical results for IA1, but different results for IA2. We have also tried to run IA2 in both directions, taking the mean of the resulting values for the supporting points. This yields in a nearly symmetrical solution (Fig. 12) and fulfils the symmetry requirement soRE2. Thus, from now on, we use this version of IA2, calling it *Interpolation Algorithm 2 modified* (IA2m). The question of monotonicity will be revisited with more realistic cases. As mentioned above, this idealised test case is not suitable for judging the fulfilment of soRE3 and soRE4.

In the next step, we extended the verification to a case with more realistic, but still synthetic data (Fig. 13). Again, we can see the non-conservative behaviour of the IFP algorithm. The new algorithms IA1 and IA2m conserve the mass within each single

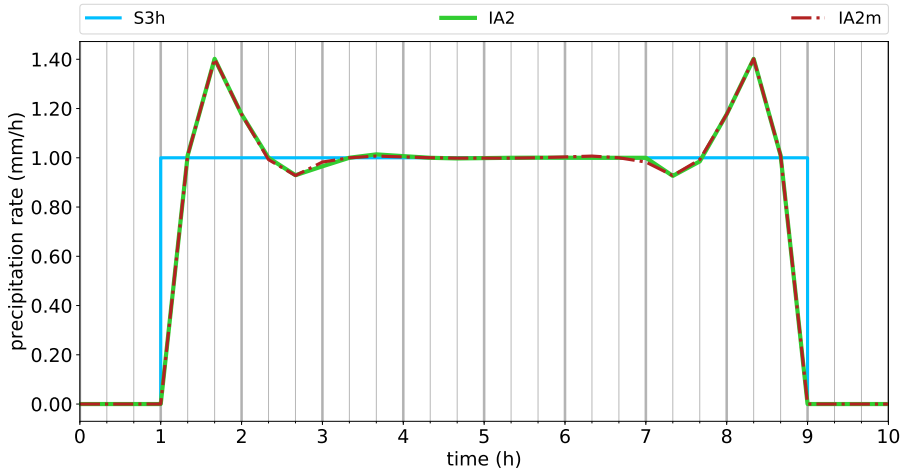

Figure 12. Same as Fig. 11, but comparing only IA2 (green line) and the modified version IA2m (red; dashed-dotted). It is recommended to zoom in to see the differences clearly.

interval within machine accuracy (ca. $\pm 10^{-15}$). As even spurious negative values need to be avoided for certain applications, values of supporting points resulting from IA1 and IA2m within the range between -10^{-12} and zero were set to zero. All of the three strict requirements (mass conservation, non-negativity, continuity) are fulfilled.

This case provides more interesting structures for looking at monotonicity than the idealised case with constant precipitation values in each rainy period. For the new algorithms, minor violations of monotonicity can be observed, e. g. around hour 30, and a smaller one after hour 3. They occur when a strong increase of the precipitation rate is followed by a weaker one or vice versa. Thus, they represent a transition to the situation discussed above where the overshooting is unavoidable, and it is difficult to judge which overshoot is possibly still realistic. Subjectively, we would prefer an algorithm that would be less prone to this phenomenon, however, we consider this deviation from soRE1 as tolerable. The symmetry requirement (soRE2) is not strictly tested here, as no symmetric structure was prescribed as input, but it can be noted that gross asymmetries as we see in IFP as shifting of peaks to the border of an interval do not occur in IA1 and IA2m.

The reconstructed precipitation curves resulting from algorithms IA1 and IA2m have a more realistic shape (soRE3) than that from IFP. Due to the two additional supporting points per interval, they are able to adapt better to strong variations. Computational efficiency (soRE4) is not tested for this still short test period.

Summarising the verification with synthetic cases, it is confirmed that IA1 and IA2m fulfill all of the strict requirements whereas IFP does not. The soft requirements are fulfilled by the new algorithms with a minor deficiency for the monotonicity condition. Next, they will be validated with real data.

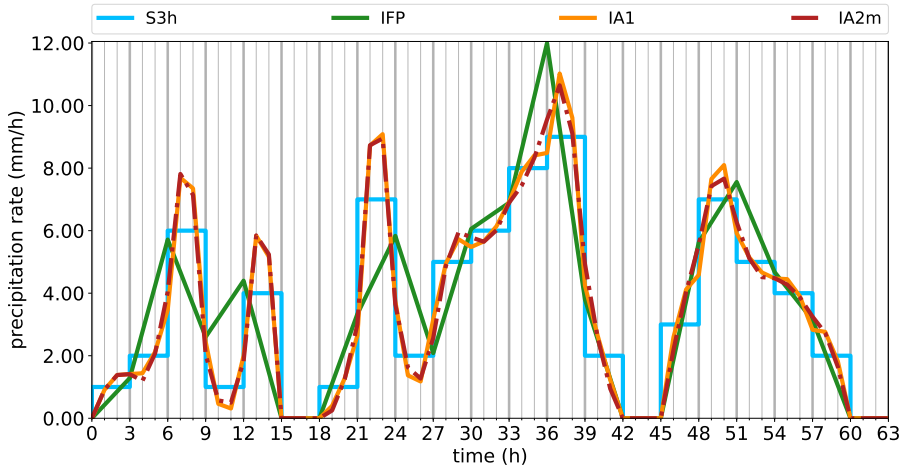

Figure 13. Verification of the different behaviour of the interpolation algorithms for a complex synthetic precipitation time series. S3h is the input data series with 3-h resolution (light blue), IFP is the linearly interpolated curve according to the current scheme in FLEXPART (green), while IA1 (orange) and IA2m (red; dashed-dotted) are the reconstructions using the new algorithms.

4.2 Validation with ECWMF data

The validation with ECMWF data makes use of precipitation data retrieved with 1-hourly and 3-hourly time resolution. The 3-hourly data serve as input to the algorithms, while the 1-hourly data are used to validate the reconstructed 1-hourly precipitation amounts. In this way, the improvement of replacing IFP by one of the new algorithms can be quantified. By using a large set of data, robust results are obtained.

4.2.1 ECWMF precipitation data

Fields of both large-scale and convective precipitation in the operational deterministic forecasts were extracted from ECMWF's MARS archive with 0.5° resolution for the whole year 2014 and the whole globe, thus yielding approximately $2.28 \cdot 10^9$ one-hourly data values (720 grid points in E–W direction, 361 in N–S direction,[3] 8761 hours including the last hour of 2013). They were extracted as 3-hourly and as 1-hourly fields. ECMWF output distinguishes these two precipitation types, derived from the grid-scale cloud microphysics scheme in the case of large-scale precipitation and from the convection scheme in the case of convective precipitation. Note that parameterised convection by definition is a sub-gridscale process, while reported precipitation intensities are averaged over the grid cell. Precipitation data are accumulated from the start of each forecast at 00 and 12 UTC. We used both these forecasts, so that the forecast lead time is at most 12 h. This is in line with typical data use in FLEXPART. Data were immediately de-accumulated to 1-h and 3-h sums (see Sect. 7 on the data availability for more details).

[3] As explained in footnote 2 on page 3, currently the precipitation extracted on a lat-lon grid is a point value. As the global grid includes the poles, there are 361 points per meridian. Nevertheless, as explained, we use the concept of a cell also horizontally, as does also FLEXPART.

4.2.2 Visual analysis of sample period

Two short periods in January 2014 were selected for visual inspection at a grid cell with significant precipitation, one dominated by large-scale and another one by convective precipitation. Convective precipitation occurs less frequently (cf. Table 5) and its variability is higher (cf. Table 3) than in the case of large-scale precipitation which is more continuous and homogeneous. We are interested in the performance of the reconstruction algorithms for both of these types. Furthermore, a criterion for the selection of the sample was that it should exhibit monotonicity problems as discussed above. The two days are typical; they do not represent a rare or extreme situation. The results are shown in Fig. 14 including the reference 1-hourly and 3-hourly ECMWF data, called R1h and R3h, respectively. Note that the same input, namely R3h, is used for all the algorithms; R1h serves for validation.

Similar to the synthetic cases, large discrepancies between the real ECMWF data and the interpolated data from the IFP algorithm can be found. This is true especially for the convective precipitation, where frequently the real peaks clipped and the mass is instead redistributed to neighbouring time intervals with lower values, leading to a significant positive bias there. The function curves of IA1 and IA2m follow the R3h signal and are even able to capture the tendency of the R1h signal as long as R1h does not have too much variability within the 3-h intervals of R3h. Again, in the convective part there is an interval where monotonicity is violated, near 11 January, 12 UTC. The secondary minimum occurs a bit earlier in the IA1 algorithm than in the IA2m algorithm, which seems typical for the case of the ascending graph (vice versa in the descending sections; see also Fig. 13).

The large-scale precipitation rate time series is smoother and precipitation events last longer (Fig. 14). This is easier for the reconstruction with all three algorithms. Nevertheless, there are occasions which show a clear improvement compared to the old IFP algorithm, for example, the double-peak structure of the precipitation event between 15 January, 18 UTC, and 16 January, 09 UTC, which is missing in the IFP curve but reconstructed by the new algorithms. Obviously, single-hour interruptions of precipitation cannot be exactly reconstructed and it is not possible to reproduce all the little detainls of the R1h time series.

Regarding the monotonicity, the large-scale precipitation time series produces a few instances with unsatisfactory monotonic behaviour, for example on 14 January 12 UTC or on 14 January 21 UTC, in the IA1 curve while the IA2m algorithm avoids the secondary minima in these cases (there are other cases where the behaviour is vice versa, not shown). The double peak structure in the IA1 and IA2m reconstructions on 15 January between 6 and 15 UTC are similar to the plateau-like ideal cases where overshooting is unavoidable.

Notwithstanding the minor problems with the monotonicity condition, the reconstructed precipitation curve from IA1 and IA2m are much closer to the real ones than the IFP curve. Therefore, we consider requirement soRE3 as basically fulfilled. This example raises the expectation that the new algorithms will be capable to improve the performance of the FLEXPART model.

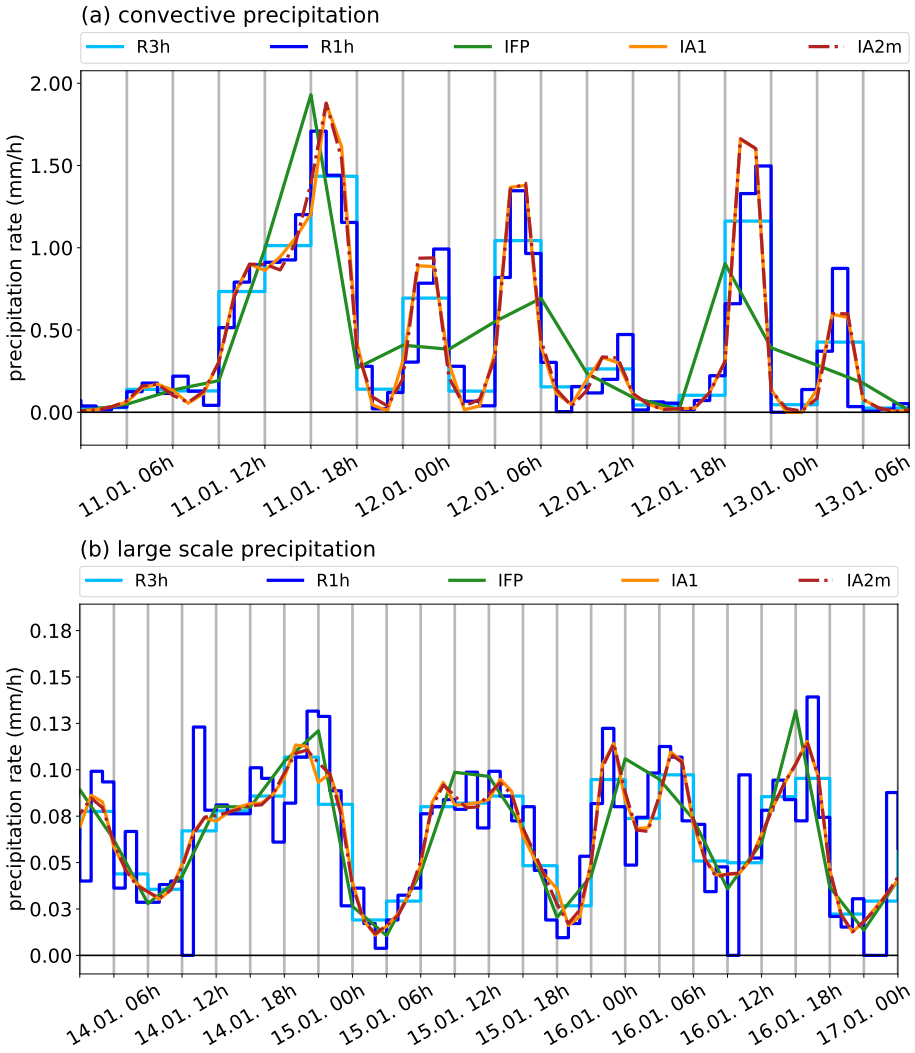

Figure 14. Sample periods in January 2014 at the grid cell centred on 48°N, 16.5°W. (a) 11 January 00 UTC to 13 January 06 UTC for convective precipitation intensity, and (b) 14 January 00 UTC to 17 January 00 UTC for large-scale precipitation intensity. Original ECMWF data are shown as step functions (1-hourly: R1h (dark blue), 3-hourly: R3h (light blue)), while the interpolation resulting from the reconstruction algorithms (new algorithms: IA1 (orange) and IA2m (red; dashed-dotted), current FLEXPART algorithm: IFP (green)) are piecewise linear between the respective supporting points. A baseline is drawn in black at intensity value zero.

4.2.3 Statistical validation

A statistical evaluation comparing the 1-hourly precipitation reconstructed by the new IA1 and IA2m algorithms as well as by the old IFP algorithm from 3-hourly input data (R3h) to the reference 1-hourly data (R1h) was carried out. While the R1h data directly represent the amount of precipitation in the respective hour, the output of the algorithms represents precipitation

Table 3. Statistical metrics for the global large-scale and convective precipitation rates ($\mathrm{mm\,h^{-1}}$) of the ECMWF data set for the year 2014. It comprises the minimum (MIN), mean of event maxima (MEX, see text for definition), mean value (MEAN), standard deviation (STD), skewness (SKEW), and kurtosis (KURT) of the true ECMWF R3h and R1h data as well as of the data reconstructed by the current FLEXPART algorithm IFP and the two new algorithms IA1 and IA2m. Among the reconstructed data, those being closest to R1h have been marked by printing in bold.

	large-scale precipitation						convective precipitation					
	MIN	MEX	MEAN	STD	SKEW	KURT	MIN	MEX	MEAN	STD	SKEW	KURT
R3h	0.0	0.75	0.0443	0.2139	22.39	1021.20	0.0	0.85	0.0567	0.2503	9.91	155.08
R1h	0.0	1.02	0.0444	0.2282	23.98	1144.16	0.0	1.53	0.0567	0.3017	12.86	275.50
IFP	0.0	0.81	0.0444	0.2172	23.66	1211.27	0.0	0.86	0.0567	0.2462	9.95	159.43
IA1	0.0	**0.91**	0.0444	0.2195	**23.84**	1188.61	0.0	**1.09**	0.0567	0.2584	**10.70**	**190.07**
IA2m	0.0	0.89	0.0444	**0.2200**	23.72	**1168.20**	0.0	1.07	0.0567	**0.2591**	10.65	187.13

rates at the supporting points of the time axis, and the hourly integrals had to be calculated, under the assumption of linear interpolation. The data set comprises the whole year of 2014 and all grid cells on the globe as described in Sect. 4.2.1. All evaluations were carried out separately for large-scale and convective precipitation.

A set of basic metrics is presented in Table 3. Since all the reconstruction algorithms are conservative, either globally (IFP) or locally (IA), the overall means must be identical. This is the case for all of the 1-hourly time series. However, the average of R3h large-scale data slightly deviates from the R1h data (fourth decimal place). This can be explained as a numerical effect, as R3h averages were calculated from fewer data. All the data sets fulfil the non-negativity requirement as indicated by a minimum value of zero. The column MEX in Table 3 contains the means of the maxima of all distinct precipitation events. They were derived from R3h and are defined as consecutive intervals with precipitation rate of at least 0.2 $\mathrm{mm\,h^{-1}}$ in each interval bounded by at least one interval with less than 0.2 $\mathrm{mm\,h^{-1}}$. The periods thus derived are also used for the 1-hourly time series. The mean of all event maxima in R1h is best reproduced by the IA1 algorithm whereas IFP underestimates it by about 20 % for large-scale and 45 % for convective precipitation. The respective numbers for IA1 are 10 % and 30 %. The higher-order moments (standard deviation, skewness, kurtosis) are generally underestimated by the reconstruction algorithms. However, the new algorithms are always closer to the R1h values than the IFP values. An exception is the kurtosis of the large-scale precipitation which is overestimated (again, less by IA1, IA2m than by IFP).

The root mean square error (RMSE), the normalised root mean square error (NMSE) and the correlation coefficient (R) between the R1h and the reconstructed data are listed in Table 4. The NMSE is calculated as

$$\mathrm{NMSE} = \sqrt{\frac{1}{N}\sum_i \frac{(\mathrm{R1h}_i - \mathrm{IA}_i)^2}{[(\mathrm{R1h}_i + \mathrm{IA}_i)/2]^2}} \tag{38}$$

where only cases with $(\mathrm{R1h}_i + \mathrm{IA}_i)/2 > 0.1$ are considered, N being the total number of these cases. The new methods represent a clear improvement compared to the IFP method, with IA2m being slightly better than IA1 with respect to all

Table 4. Root-mean-square error (RMSE, $\mathrm{mm\,h^{-1}}$), normalised root-mean-square error (NMSE, $\mathrm{mm\,h^{-1}}$) and correlation coefficient (R) between the interpolated data sets IFP, IA1 and IA2m and the true ECMWF data R1h, based on the global data set for the year 2014, large-scale and convective precipitation. Note that NMSE is calculated only for data pairs clearly different from zero as described in the text.

	large-scale precipitation			convective precipitation		
	RMSE	NMSE	R	RMSE	NMSE	R
IFP_R1h	0.0860	0.4270	0.927	0.1896	0.9091	0.779
IA1_R1h	0.0630	0.3330	0.961	0.1605	0.7995	0.847
IA2m_R1h	**0.0610**	**0.3241**	**0.964**	**0.1597**	**0.7970**	**0.849**

parameters. The large-scale precipitation reconstruction is obviously more accurate than that of the convective precipitation, even though this gap is reduced by IA1 and IA2m.

Another aspect is the ability of the algorithms to conserve the ratio of dry and wet intervals (Table 5). Two different thresholds of the precipitation intensity were chosen to separate 'dry' and 'wet'. The lower one, 0.002 $\mathrm{mm\,h^{-1}}$ corresponds to about 0.05 mm per day (rounded 0.1 mm, the lowest non-zero value reported by meteorological stations). The higher one is 0.2 $\mathrm{mm\,h^{-1}}$ and indicates substantial rain or snowfall. Again, the results are reported separately for large-scale and convective precipitation. In all cases, the reconstructions produce too many wet intervals. The relative deviations are larger for the lower threshold, and for convective precipitation in comparison to large-scale precipitation. In all cases, the new algorithms result in a clear improvement compared to the current IFP algorithm. For the high threshold and large-scale precipitation, however, already IFP deviates only by 1.9 %; for convective precipitation, the relative deviation is improved from 15 % to 11 %. In the case of the lower threshold, the improvement is from 18 % to 13 % and 35 % to 23 %, the latter for convective precipitation. The differences between IA1 and IA2m are marginal, with the latter being better in three of the four situations.

Finally, two-dimensional histograms (relative frequency distributions) are provided for a more detailed insight into the relationship between the reconstructed and the true R1h values (Fig. 15). The larger scatter in the convective precipitation compared to the large-scale one is striking. The distributions are clearly asymmetric with respect to the diagonal, especially for the convective precipitation. One has to be careful in the interpretation, however, because most cases are concentrated in the lower left corner (log scale for the frequencies, spanning many orders of magnitude!). Thus, at least for the high values, more points fall below the diagonal, indicating more frequent underprediction. This might be due to the short duration of peaks with the highest intensity. For both precipitation types, but especially for convective precipitation, an overestimation of very low intensities is noticeable. Zooming in, the first R1h bin for the convective precipitation shows enhanced values corresponding to the bias towards wet cases in Table 5. This is continued as a general levelling off of (imagined) frequency isolines towards the y-axis, which corresponds to the weaker, but still present dry-wet bias with higher thresholds. Another feature for the convective precipitation is the structure noticeable in the sector below the diagonal. Especially in the IFP plot, a secondary maximum is visible in the light-blue area, indicating a characteristic severe underprediction. This is less pronounced for IA1

Table 5. Frequencies of dry (h_d) and wet (h_w) intervals in the reference (R3h, R1h) and interpolated (IFP, IA1, IA2m) precipitation data (upper part), based on the global data set for the year 2014. Relative deviations (δ_d and δ_w) between the three interpolations and R1h are shown in the lower part. Two different thresholds (0.2 $\mathrm{mm\,h^{-1}}$ and 0.002 $\mathrm{mm\,h^{-1}}$) were used to separate 'wet' and 'dry'. Large-scale and convective precipitation were analysed separately. All values are in percent. The reconstructed values that match best the true R1h data are printed in bold.

	threshold = 0.002 $\mathrm{mm\,h^{-1}}$				threshold = 0.200 $\mathrm{mm\,h^{-1}}$			
	large-scale precipitation		convective precipitation		large-scale precipitation		convective precipitation	
	h_d	h_w	h_d	h_w	h_d	h_w	h_d	h_w
R3h	59.74	40.26	72.11	27.89	95.39	4.61	94.06	5.94
R1h	64.72	35.28	77.62	22.38	95.49	4.51	94.72	5.28
IFP	58.21	41.79	69.79	30.21	95.40	4.60	93.93	6.07
IA1	60.16	39.84	**72.51**	**27.49**	95.42	4.58	94.13	5.87
IA2m	**60.24**	**39.76**	72.46	27.54	**95.43**	**4.57**	**94.15**	**5.85**
	δ_d	δ_w	δ_d	δ_w	δ_d	δ_w	δ_d	δ_w
IFP_R1h	-10.06	18.46	-10.08	34.95	-0.09	1.87	-0.83	14.97
IA1_R1h	-7.05	12.94	**-6.59**	**22.84**	-0.08	1.59	-0.63	11.29
IA2m_R1h	**-6.93**	**12.71**	-6.65	23.04	**-0.07**	**1.41**	**-0.60**	**10.78**

and almost absent for IA2m. Summarising, the scatter plots indicate an improvement from IFP towards IA1 and IA2m. A near-perfect agreement obviously cannot be expected as the information content in the 3-h input data is of course less than in the 1-h data. This information gap is larger for the convective precipitation which obviously has a shorter autocorrelation time scale.

4.3 Performance

Potential applications for the new algorithm include situations where computational performance is relevant. For the precipitation (and possibly other input data, see Sect. 5), both the time for the preprocessing software flex_extract (which includes reconstruction algorithm to calculate the supporting points) and time for interpolation in FLEXPART itself are relevant, and they should not significantly exceed the current computational efforts.

During the evaluation process a computationally more efficient version of the IA1 algorithm was developed. It applies the monotonicity filter within one sweep through the time series (filter trailing behind the reconstruction) rather than processing the series twice. The algorithmic equations are unchanged. We refer to this version as *Interpolation Algorithm IA1 modified* (IA1m). It was verified that results are not different from the standard IA1.

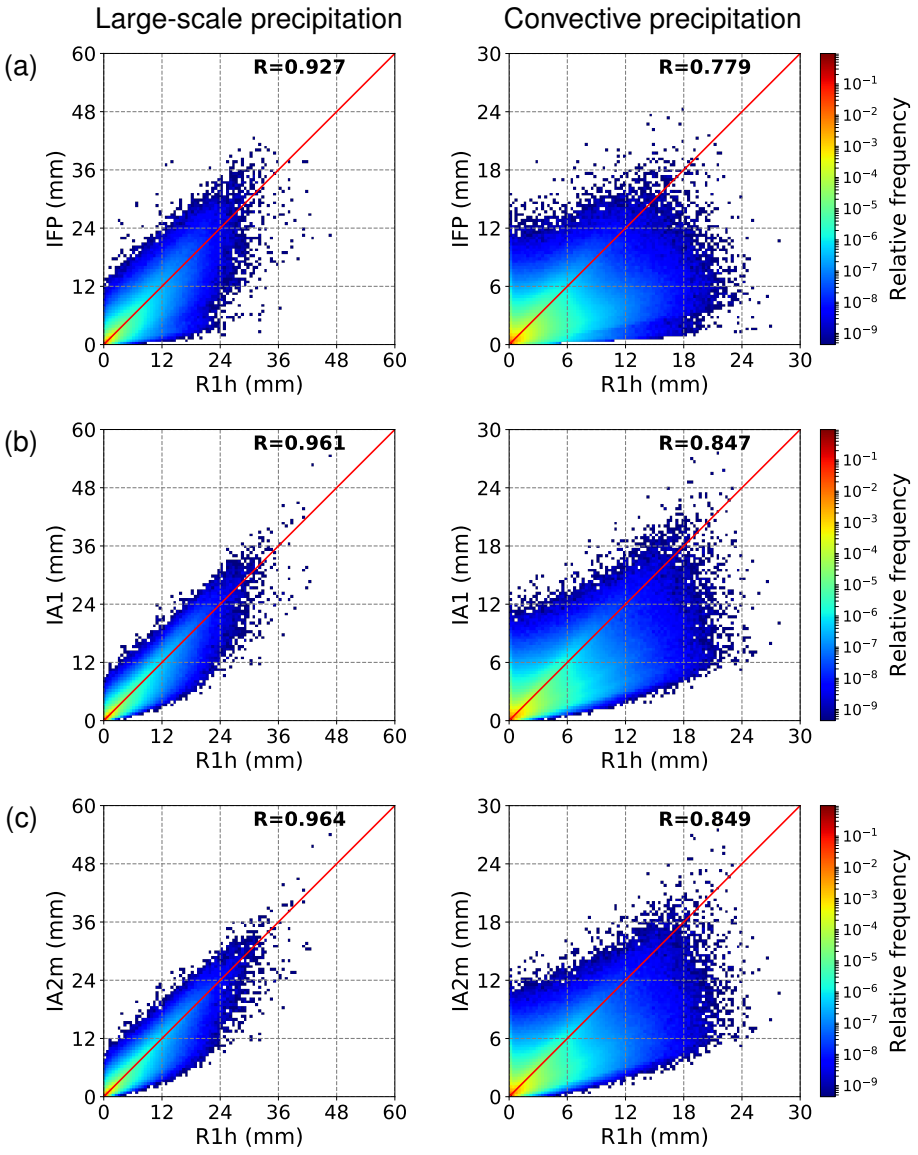

Figure 15. Two-dimensional histogram (relative frequencies) showing the relationship between the hourly precipitation reconstructed by (a) IFP, (b) IA1 and (c) IA2m and the true R1h precipitation, based on the global data set for the year 2014. For each axis, 100 equally-distributed bins were used. The correlation coefficient is annotated to each plot, and the one-to-one line is shown in red.

The wall clock time for the application of each of the algorithms to the 1-year global test data set is listed in Table 6. This is the time needed to reconstruct the new time series with Python and to save the data in the npz format provided by Python's NumPy package, which is the most efficient way to write them out. The computing time for all the new algorithms is similar

Table 6. Computing time (wall-clock) for the processing of one year of global data (ECMWF test data used in this paper) with the old IFP algorithm and the new IA1, IA1m and IA2m algorithms, on a Linux server with Intel(R) Xeon(R) E5-2690 @ 2.90GHz CPU, single thread.

Algorithm	Wall-clock time
IFP	2 h 29 min
IA1	2 h 24 min
IA1m	**2 h 06 min**
IA2m	2 h 56 min

to that of the old algorithm. The fastest version is IA1m, which needs about 85 % of the time required by IFP, while the IA2m, the slowest version, takes about 118 % of IFP.

5 Conclusions and Outlook

5.1 Conclusions

We have provided a one-dimensional, conservative and positive-definite reconstruction algorithm suitable for the interpolation of a gridded function whose grid values represent integrals over the grid cell, such as precipitation output from numerical models. The approach is based on a one-dimensional piecewise-linear function with two additional supporting points within each grid cell, dividing the interval into three pieces.

This approach has three degrees of freedom, similar to a piecewise parabolic polynomial. They are fixed through the mass conservation condition, the slope of the central interval which is taken as the average of the slopes of the two outer subintervals, and the left and right border grid points (each counting as half a degree of freedom). For the latter, the geometric mean value of the bordering integral values is chosen. Its main advantage is that the function values vanish automatically if one of the involved values is zero, which is a necessary condition for continuity. However, the geometric mean in general converges too slowly with vanishing values to prevent negative values under all conditions. This led us further to derive a sufficient condition for non-negativity and restrict the function values accordingly by these upper bounds.

This non-negative geometric-mean-based algorithm, however, still violates monotonicity. Therefore, we further introduced a (conservative) filter for regions where the remapping function takes an 'M'- or 'W'-like shape, requiring a second run through the data (IA1). Alternatively, the filter can be applied during the first sweep immediately after the construction of the next interval (IA1m). We also showed how this basic idea of the monotonicity filter can be directly incorporated into the construction of an algorithm (IA2). As in this case the algorithm is not symmetric, we apply it a second time in the other direction and average the results (IA2m).

The evaluation, consisting of verification and validation, confirmed the advantages of the new algorithms IA1 (including IA1m) and IA2m. After the verification of our requirements, each evaluation step revealed a significant improvement by the new algorithms as compared to the algorithm currently used for the FLEXPART model. Nevertheless, the soft requirement of

monotonicity has not been fulfilled perfectly but the deviation is considered to be acceptable. The modified version of IA1, IA1m, yields identical results to IA1 and is quite fast. However, the results of the quantitative statistical validation would slightly favour the IA2m algorithm whose computational performance is still acceptable, even though in this modified form the original IA2 algorithm is applied twice.

5.2 Outlook

The next steps will be the integration of the method into the preprocessing of the meteorological input data for the FLEXPART Lagrangian dispersion model and the model itself for the temporal interpolation of precipitation. The application to two dimensions, intended for spatial interpolation, is also under investigation. Options include the straightforward operator-splitting approach as well as an extension based on bilinear interpolation with additional supporting points. As the monotonicity filter appears to be not yet perfect, this may also be revisited.

5.3 Possible other applications of the new piecewise-linear reconstruction method

It may be noted that there is a wide range of useful applications of such conservative reconstructions. Interestingly, at least in the geoscientific modelling community, they have largely remained restricted to the specific problem of semi-Lagrangian advection schemes. Therefore, we are sketching below more possible use cases.

In typical LPDMs, other extensive quantities which are being used, apart from precipitation, are surface fluxes of heat and momentum which enter boundary-layer parameter calculations and which could be treated similarly, especially for temporal interpolation. The often-used three-hourly input interval is quite coarse and may clip, for example, the peak values of the turbulent heat flux.

In many applications, output is required for single points representing measurement stations or, in the case of backward runs (Seibert and Frank, 2004), point emitters. While FLEXPART has the option of calculating concentrations at point receptors with a parabolic sampling kernel, the results have often been similar to simple bilinear interpolation of gridded output, probably because of the difficulty to determine an optimum kernel width; therefore many users produce only the gridded output and take the point values through a nearest-neighbour or a bilinear interpolation approach. The piecewise-linear interpolation with additional supporting points as introduced here, or one of the higher-order methods discussed in Sect. 2, would probably provide an improvement.

This latter example could be easily extended to all kinds of model output postprocessing, where currently too simple methods often prevail. It should be clear that applying naive bilinear interpolation to gridded output of precipitation and other extensive quantities, including fluxes, introduces systematic errors as highlighted in Sect. 1.

Finally, this also includes contouring software. Contouring involves interpolation between neighbouring supporting points to determine where the contour line should intersect the cell boundaries. It is obvious that linear interpolation is inadequate for extensive quantities whose values represent grid averages. This holds in particular for precipitation, energy fluxes, and trace species concentrations. While we cannot expect the many contouring packages to be rewritten with an option for conservative interpolation, our method (once extended to two dimensions) provides an easy implementation through preprocessing resulting

in an auxiliary grid with triple (one-dimensional) resolution that then could be linearly interpolated without violating mass conservation, thus being able to be used with standard contouring software.

6 Code availability

The piecewise-linear reconstruction routines IA1, IA2 and IA1m are written in Python2. The code is included in the supplementary material and is licensed under the Creative Commons Attribution 4.0 International License. For IA2m, IA2 has to be called with the original and the reversed time series and the results have to be averaged. The software for the statistical evaluation (written in Python2) is available on request by contacting the second author, A. Philipp (anne.philipp /at/ univie.ac.at). It relies on the NumPy (Walt et al., 2011) and SciPy (Jones et al., 2001–2017) libraries for data handling and statistics, and on Matplotlib (Hunter, 2007) for the visualisation. The FLEXPART model as well as the accompanying data extraction software can be downloaded from the community site http://flexpart.eu/.

7 Data availability

The precipitation data used for evaluating the interpolation algorithms were extracted from ECMWF through MARS retrievals (ECMWF, 2017). For easy reproducibility of the results, we provide the MARS retrieval routines in the supplementary material, licensed under the Creative Commons Attribution 4.0 International License. Note that only authorised ECMWF users have access to the operational archive data.

Appendix A: Theoretical aspects of the monotonicity filter

We show in the following that the monotonicity filter as introduced in Sect. 3.3.5 improves the monotonicity behaviour and also does not destroy non-negativity. The conservation of mass (equal-area condition) is clearly preserved by construction.

(i) The "M"-shape: This corresponds to the case

$$k_i^{(2)} > 0 \quad \wedge \quad k_i^{(3)} < 0 \quad \wedge \quad k_{i+1}^{(1)} > 0 \quad \wedge \quad k_{i+1}^{(2)} < 0\,. \tag{A1}$$

Then, the filter improves the monotonicity behaviour in the sense that

$$f_{i+1}^{\mathrm{mon}} > f_{i+1} \qquad \text{for all } i \in \mathcal{I}\,. \tag{A2}$$

To see this we note that according to Eq. (17), the conditions $k_i^{(2)} > 0$ and $k_{i+1}^{(2)} < 0$ are equivalent to

$$f_i < f_{i+1} \qquad \text{and} \qquad f_{i+2} < f_{i+1}\,. \tag{A3}$$

Furthermore, from the condition $k_i^{(3)} < 0$ we can deduce

$$f_{i+1} < f_i^{(2)} = \frac{3}{2} g_i - \frac{5}{12} f_i - \frac{1}{12} f_{i+1} \qquad \text{and thus} \qquad g_i > \frac{13}{18} f_{i+1} + \frac{5}{18} f_i\,,$$

implying in particular also

$$f_{i+1} < 3g_i\,, \qquad \text{and furthermore} \qquad f^{\diamond}_{i+1} \;=\; \frac{18}{13}g_i - \frac{5}{13}f_i > f_{i+1}\,. \tag{A4}$$

From the condition $k^{(1)}_{i+1} > 0$ we can deduce in a similar fashion that

$$f_{i+1} < f^{(1)}_{i+1} = \frac{3}{2}g_{i+1} - \frac{1}{12}f_{i+1} - \frac{5}{12}f_{i+2} \qquad \text{and thus} \qquad g_{i+1} > \frac{13}{18}f_{i+1} + \frac{5}{18}f_{i+2}\,,$$

such that in particular also

$$f_{i+1} < 3g_{i+1}\,, \qquad \text{and moreover} \qquad f^{\diamond\diamond}_{i+1} \;=\; \frac{18}{13}g_{i+1} - \frac{5}{13}f_{i+2} > f_{i+1}\,. \tag{A5}$$

Making use of all derived inequalities we can deduce

$$f^{\text{mon}}_{i+1} = \min\left\{3g_i, 3g_{i+1}, \sqrt{(f^{\diamond}_{i+1}\, f^{\diamond\diamond}_{i+1})_+}\right\} > f_{i+1}\,.$$

(ii) The "W"-shape. This corresponds to the case

$$k^{(2)}_i < 0 \quad \wedge \quad k^{(3)}_i > 0 \quad \wedge \quad k^{(1)}_{i+1} < 0 \quad \wedge \quad k^{(2)}_{i+1} > 0\,. \tag{A6}$$

Again the filter improves the monotonicity behaviour in the sense that

$$f^{\text{mon}}_{i+1} < f_{i+1} \qquad \text{for all } i \in \mathcal{I}\,. \tag{A7}$$

Arguing similarly, now the conditions $k^{(2)}_i < 0$ and $k^{(2)}_{i+1} > 0$ imply

$$f_{i+1} < f_i \qquad \text{and} \qquad f_{i+1} < f_{i+2} \tag{A8}$$

and thus in particular

$$f_{i+1} < \min\{3g_i, 3g_{i+1}\}\,. \tag{A9}$$

This can be easily proven by contradiction, since otherwise if $f_{i+1} = 3g_i$ (or $f_{i+1} = 3g_{i+1}$), then also $f_i \le 3g_i$ (or $f_{i+2} \le 3g_{i+1}$ respectively) by construction, implying furthermore $f_i \le f_{i+1}$ (or $f_{i+2} \le f_{i+1}$), which obviously contradicts (A8). From the condition $k^{(3)}_i > 0$ we can deduce similar to above

$$g_i < \frac{13}{18}f_{i+1} + \frac{5}{18}f_i\,, \qquad \text{and therefore} \qquad f^{\diamond}_{i+1} \;=\; \frac{18}{13}g_i - \frac{5}{13}f_i < f_{i+1}\,. \tag{A10}$$

From $k^{(1)}_{i+1} < 0$ we obtain accordingly

$$g_{i+1} < \frac{13}{18}f_{i+1} + \frac{5}{18}f_{i+2} \qquad \text{such that} \qquad f^{\diamond\diamond}_{i+1} \;=\; \frac{18}{13}g_{i+1} - \frac{5}{13}f_{i+2} < f_{i+1}\,. \tag{A11}$$

Note that in comparison to the "M" - shape above there is no lower bound for the different constructed values $f^{\diamond}_{i+1}, f^{\diamond\diamond}_{i+1}$. Thus they can take possibly negative values and care has to be taken here to ensure that the filter still preserves non-negativity. Combining now (A9), (A10) and (A11) we can deduce

$$f^{\text{mon}}_{i+1} = \min\left\{3g_i, 3g_{i+1}, \sqrt{(f^{\diamond}_{i+1} f^{\diamond\diamond}_{i+1})_+}\right\} = \sqrt{(f^{\diamond}_{i+1} f^{\diamond\diamond}_{i+1})_+} < f_{i+1}\,. \tag{A12}$$

Author contributions. P. Seibert formulated the problem in Sec. 1, made initial suggestions and guided the work. Section 2 was jointly written by S.Hittmeir and P. Seibert. Section 3 and Appendix A have been contributed mainly by S. Hittmeir (with additions by A. Philipp), Section 4 by A. Philipp. Section 5 has been written jointly by all authors. The supplementary material was contributed by A. Philipp.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. S. Hittmeir thanks the Austrian Science Fund (FWF) for the support via the Hertha-Firnberg project T-764-N32 who also finances the open-access publication. We thank the Austrian Meteorological Service ZAMG for the access to the ECMWF data. The second and third author are grateful to Christoph Erath (then Department of Mathematics, University of Vienna; now at TU Darmstadt, Germany) for an early-stage discussion of the problem and useful hints about literature from the semi-Lagrangian advection community. We also thank the reviewers for careful reading, thus helping to improve the paper.

References

ECMWF: MARS User Documentation, https://software.ecmwf.int/wiki/display/UDOC/MARS+user+documentation, [Online; 10.08.2017], 2017.

Hämmerlin, G. and Hoffmann, K.-H.: Numerische Mathematik, Springer, 4th edn., doi:10.1007/978-3-642-57894-6, 1994.

Hermann, M.: Numerische Mathematik, München Oldenburg, 3rd edn., 2011.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in Science & Engineering, 9, 90–95, doi:10.1109/MCSE.2007.55, 2007.

Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python, http://www.scipy.org/, [Online; 10.08.2017], 2001–2017.

Lauritzen, P. H., Ullrich, P. A., and Nair, R. D.: Atmospheric transport schemes: Desirable properties and a semi-Lagrangian view on finite-volume discretizations, in: Numerical Techniques for Global Atmospheric Models, edited by Lauritzen, P. H., Jablonski, C., Taylor, M. A., and Nair, R. D., vol. 80 of *Lecture Notes in Computational Science and Engineernig*, chap. 8, pp. 185–250, Springer, doi:10.1007/978-3-642-11640-7_8, 2010.

Lin, J. C., Brunner, D., Gerbig, C., Stohl, A., Luhar, A., and Webley, P., eds.: Lagrangian Modeling of the Atmosphere, vol. 26 of *AGU Geophysical Monograph*, American Geophysical Union, doi:10.1029/2012GM001294, 2013.

Schmidt, J. W. and Heß, W.: Positivity of cubic polynomials on intervals and positive spline interpolation., BIT Numer. Math., 28, 340–352, doi:10.1007/BF01934097, 1988.

Seibert, P. and Frank, A.: Source-receptor matrix calculation with a Lagrangian particle dispersion model in backward mode, Atmos. Chem. Phys., 4, 51–63, http://www.atmos-chem-phys.net/4/51/2004/, 2004.

Stohl, A., Forster, C., Frank, A., Seibert, P., and Wotawa, G.: Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2, Atmos. Chem. Phys., 5, 2461–2474, doi:10.5194/acp-5-2461-2005, 2005.

Walt, S. v. d., Colbert, S. C., and Varoquaux, G.: The NumPy array: A structure for efficient numerical computation, Computing in Science and Engineering, 13, 22–30, doi:10.1109/MCSE.2011.37, 2011.

White, L., Adcroft, A., and Hallbert, R.: High-order regridding-remapping schemes for continuous isopycnal and generalized coordinates in ocean models, J. Comp. Phys., 228, 8665–8692, doi:10.1016/j.jcp.2009.08.016, 2009.

Zerroukat, M., Wood, N., and Staniforth, A.: SLICE: A Semi-Lagrangian Inherently Conserving and Efficient scheme for transport problems, Q. J. Roy. Meteorol. Soc., 128, 2801–2820, doi:10.1256/qj.02.69, 2002.

Zerroukat, M., Wood, N., and Staniforth, A.: A monotone and positive-definite filter for a Semi-Lagrangian Inherently Conserving and Efficient (SLICE) scheme, Q. J. Roy. Meteorol. Soc., 131, 2923–2936, doi:10.1256/qj.04.97, 2005a.

Zerroukat, M., Wood, N., and Staniforth, A.: The Parabolic Spline Method (PSM) for conservative transport problems, Int. J. Numer. Meth. Fluids, doi:10.1002/fld.1154, 2005b.