Solar Energy 248 (2022) 121–127

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener

Mean-preserving interpolation with splines for solar radiation modeling

José A. Ruiz-Arias *

Applied Physics I, University of Málaga, Málaga, Spain
Solargis s.r.o., Bratislava, Slovakia

ARTICLE INFO

Dataset link: https://github.com/jararias/mpsplines/tree/paper

Keywords:
Solar models coupling
Splines interpolation
Mean preservation
Periodic boundaries

ABSTRACT

Interpolation is a fundamental process in solar resource assessment that glues consecutive components of the modeling chain. Most interpolation techniques assume that the interpolating function must go through the interpolation points. However, this assumption does not fit with averaged datasets or variables that must be conserved across interpolation. Here I present a mean-preserving splines method for interpolating one-dimensional data that conserves the interpolated field and is appropriate for averaged datasets. It uses second-order polynomial splines to minimize the fluctuations of the interpolated field, restricts the interpolation results to user-provided limits to prevent unphysical values, deals with periodic boundary conditions in the interpolated field, and can work with non-uniform averaging grids. The validity and performance of the method are illustrated against regular second- and third-order splines using relevant case examples in the solar resource assessment realm.

1. Introduction

The countless and disparate data streams available nowadays, directly observed or modeled, make data interpolation a critical step to couple processes in earth system modeling (Warner et al., 2008), hydrological modeling (Gupta and Tarboton, 2016) and landscape models (De Caceres et al., 2018), to name a few. The domain of solar models for renewable energy is not an exception.

Evaluating solar irradiance at the ground surface requires ancillary variables that are not necessarily available at the temporal resolution needed for solar applications. An example is the evaluation of clear-sky solar irradiance from clear-sky models (Ruiz-Arias and Gueymard, 2018a,b; Ruiz-Arias et al., 2019; Sun et al., 2019, 2021) that require various atmospheric variables such as aerosol optical depth (AOD), precipitable water, total column ozone content, and ground albedo, among others. In most current applications, these variables are retrieved from numerical weather datasets (e.g., Gelaro et al., 2017; Inness et al., 2019). In the best case, these datasets are available at hourly time steps although clear-sky solar irradiance may be needed at a different resolution (e.g., every 10-min to match the satellite imagery retrieval times). In addition, the intrinsic uncertainty of these hourly weather variables, possibly combined with a moderate impact on solar irradiance, might bring with none or little benefit compared to coarser scales, such as daily, as it is pointed out by Ruiz-Arias (2020) for AOD predictions of current state-of-the-art weather models. Hence, in some cases, even daily data would be a reasonable choice for weather variables because daily data is easier to archive, retrieve and maintain. Be that as it may, the weather data must be first interpolated to the time grid at which solar radiation is needed. Only afterwards, they are used in solar models to evaluate solar radiation.

Splines interpolation is a popular interpolation technique because of its high adaptability to multiple problems and ease of use. It evaluates the interpolated data from a continuous and differentiable piecewise function. Generally, with $n + 1$ interpolation nodes (i.e., the known data samples to infer the interpolated values), the piecewise function is made up of n polynomials, known as *splines*, that collide at each interpolation node. The polynomials' coefficients are chosen such that the interpolating piecewise function is continuous and the interpolation exact, that is, it goes through all the interpolation nodes (Knott, 2000; Beu, 2015). The exactness constraint is appropriate when the interpolation nodes result directly from a sampling process. For instance, we can gather global horizontal irradiance (GHI) measurements every second but retain only the samples at multiples of 60 s. As they are actual measurements drawn from the 1 s time series, splines interpolation as defined above is appropriate to reconstruct the 1 s time series from the 60 s samples. Alternatively, however, we might have retained 60 s averages of GHI measurements. As they are not necessarily actual values drawn from the 1 s time series, the exactness constraint in splines interpolation will not be appropriate.

In addition to clear-sky solar irradiance models, many other applications use interpolated average datasets. For instance, those that utilize monthly climatologies (e.g., temperature and humidity), daily

* Correspondence to: Applied Physics I, University of Málaga, Málaga, Spain.
E-mail addresses: jararias@uma.es, jose.ruiz-arias@solargis.com.

https://doi.org/10.1016/j.solener.2022.10.038
Received 15 August 2022; Received in revised form 29 September 2022; Accepted 18 October 2022
Available online 17 November 2022

averages (e.g., AOD and precipitable water), or even accumulated variables, such as precipitation and solar irradiance predictions. In such cases, using exact splines to interpolate these variables is inappropriate. Being aware of this issue, Delhez (2003) replaced the usual third-order splines with fourth-order splines and updated the splines' constraints to ensure that the average of the splines matches the interpolation node values. However, the high order of the splines may yield large fluctuations and non-physical values (e.g., negative precipitation rate). Very recently, Lai and Kaplan (2022) have proposed a new method based on cubic splines that preserves the mean and places bounds to prevent non-physical interpolated values. Nevertheless, the code is not readily usable into general purpose time series applications. Taking a different approach, Rymes and Myers (2001) proposed a recursive algorithm based on simplified Markov autoregressive processes. However, it was specifically devised for solar irradiance data, but not for ancillary variables.

In view of the limitations of the existing solutions, this work proposes a general-purpose mean-preserving splines interpolation method, hereafter referred to as *mp*-splines, in which the average of each spline matches the value of the corresponding interpolation node. It uses second-order polynomial splines to minimize the fluctuations in the interpolated values and provides a means to prevent non-physical results. The method is implemented in the ubiquitous Python language, and is shared publicly in https://github.com/jararias/mpsplines.

2. Mean-preserving splines interpolation

Let $\{x_i, y_i\}_{i=1,2,\ldots,n}$ be n known pairs of values (the interpolation nodes) of an explanatory process x and the unknown process y, respectively. They verify that y_i is the average of y throughout the interval $[x_{l,i}, x_{u,i}]$ for arbitrary known values $x_{l,i}$ and $x_{u,i}$ such that $x_i \in [x_{l,i}, x_{u,i}]$ and $x_{u,i} = x_{l,i+1}$. Note that, conversely to other methods, x_i is not necessarily ascribed to the center of the interval $[x_{l,i}, x_{u,i}]$. In practice, this fact generalizes the application of the algorithm to averaged series with arbitrary non-uniform integration limits.

The interpolating piecewise function that reconstructs y is made up of n second-order polynomials defined as:

$$S_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i, \quad x_{l,i} \le x \le x_{u,i}, \tag{1}$$

where a_i, b_i and c_i are arbitrary coefficients. The interpolating function is subject to the following constraints:

$$\int_{x_{l,i}}^{x_{u,i}} S_i(x)\,dx = y_i\left(x_{u,i} - x_{l,i}\right), \quad i = 1, 2, \ldots, n \tag{2a}$$

$$S_i(x_{u,i}) = S_{i+1}(x_{l,i+1}), \quad i = 1, 2, \ldots, n-1 \tag{2b}$$

$$S_i'(x_{u,i}) = S_{i+1}'(x_{l,i+1}), \quad i = 1, 2, \ldots, n-1 \tag{2c}$$

where $S_i'(x)$ is the first derivative of $S_i(x)$.

The first constraint ensures that the average of the interpolating function throughout the interval $[x_{l,i}, x_{u,i}]$ is y_i, and translates into the following set of n equations:

$$\frac{1}{3}\left(\Delta_{u,i}^3 - \Delta_{l,i}^3\right)a_i + \frac{1}{2}\left(\Delta_{u,i}^2 - \Delta_{l,i}^2\right)b_i + \left(\Delta_{u,i} - \Delta_{l,i}\right)c_i - \left(\Delta_{u,i} - \Delta_{l,i}\right)y_i = 0, \quad i = 1, 2, \ldots, n \tag{3}$$

where $\Delta_{l,i} = x_{l,i} - x_i$ and $\Delta_{u,i} = x_{u,i} - x_i$.

The second and third constraints ensure the continuity of the interpolating function, and translates, respectively, into the following sets of equations:

$$\Delta_{u,i}^2 a_i + \Delta_{u,i} b_i + c_i - \Delta_{l,i+1}^2 a_{i+1} - \Delta_{l,i+1} b_{i+1} - c_{i+1} = 0, \tag{4a}$$

$$2\Delta_{u,i} a_i + b_i - 2\Delta_{l,i+1} a_{i+1} - b_{i+1} = 0 \tag{4b}$$

with $i = 1, 2, \ldots, n-1$.

In total, the constraints make up a system of $3n - 2$ linear and independent equations, but the interpolating function is defined by $3n$ coefficients. Hence, two more equations are needed.

If the process y is periodic, such as it may be the case for long-term monthly climatologies, the first and last splines should be continuous. To that aim, two additional constraints are $S_1(x_{l,1}) = S_n(x_{u,n})$ and $S_1'(x_{l,1}) = S_n'(x_{u,n})$, which translates respectively into:

$$\Delta_{u,n}^2 a_n + \Delta_{u,n} b_n + c_n - \Delta_{l,1}^2 a_1 - \Delta_{l,1} b_1 - c_1 = 0 \tag{5a}$$

$$2\Delta_{u,n} a_n + b_n - 2\Delta_{l,1} a_1 - b_1 = 0. \tag{5b}$$

If the process y is not periodic, the two additional constraints are defined by imposing continuity of the second derivative of the interpolating function in the upper edge of the first spline and the lower edge of the last spline, that is, $S_1''(x_{u,1}) = S_2''(x_{l,2})$ and $S_{n-1}''(x_{u,n-1}) = S_n''(x_{l,n})$. They turn into the following equations:

$$a_1 - a_2 = 0 \tag{6a}$$

$$a_{n-1} - a_n = 0. \tag{6b}$$

The choice of these constraints is not unique, and others might have been implemented. However, these translate into simple equations that lead to reasonable results.

All together, Eq. (3) and (4) combined, plus Eq. (5) for periodic boundary conditions, or Eq. (6) for free boundary conditions, make a linear system of $3n$ independent equations with $3n$ unknowns that can be solved using conventional methods. The solution of the system fully defines the interpolating function.

Fig. 1 illustrates the working principles of *mp*-splines interpolation. Unlike regular splines, the interpolated curve does not necessarily pass through the interpolation nodes. The preservation of the average implies that the red and orange areas under and over each spline are equal.

3. Constrained interpolation

The mean-preserving splines algorithm imposes constraints for smoothness, mean state, and continuity. However, none of them prevents the interpolated values to take non-physical results. A typical case is that of monthly precipitation rates interpolated to daily steps. Since the precipitation rate may be quite low during the dry season, the interpolation algorithm may fall into negative values. This issue is common to other splines interpolation algorithms. Nonetheless, the risk of falling into non-physical values is higher with *mp*-splines precisely because of the averaging constraint enforced by Eq. (3).

As a countermeasure, the *mp*-splines algorithm detects departures of the interpolated values into the non-permitted domain and relaxes the constraints in a neighborhood of the culprit spline. In particular, the infringing spline and its neighbors (by default, four on each side) are replaced by third-order polynomials. The additional degree of freedom in each spline allows for a readjustment of the data while still respecting the averaging constraint. The domain of permitted values is delimited by a minimum threshold that can be set by the user. In the precipitation rate example, the user would set this threshold to 0. A maximum threshold is not explicitly considered because it rarely happens in practical solar applications. However, in such a situation, the sign of the interpolated variable can be switched so that the maximum threshold becomes a minimum. The current implementation of *mp*-splines does not allow for minimum and maximum thresholds simultaneously.

A downside of the local relaxation is that it requires a numerical iteration, in this case using Sequential Least Squares Programming (Kraft, 1988; Virtanen et al., 2020), that increases the interpolation time, especially when many interpolation violations occur. Fortunately, the latter is not usual. In the uncommon case when the iterative process does not reach a solution, the interpolated values are clipped on the minimum permitted value.

Fig. 2 shows the interpolation of a monthly rainfall climatology to daily steps using multiple splines interpolation approaches. Both regular 2nd- and 3rd-order splines interpolation, and even the regular

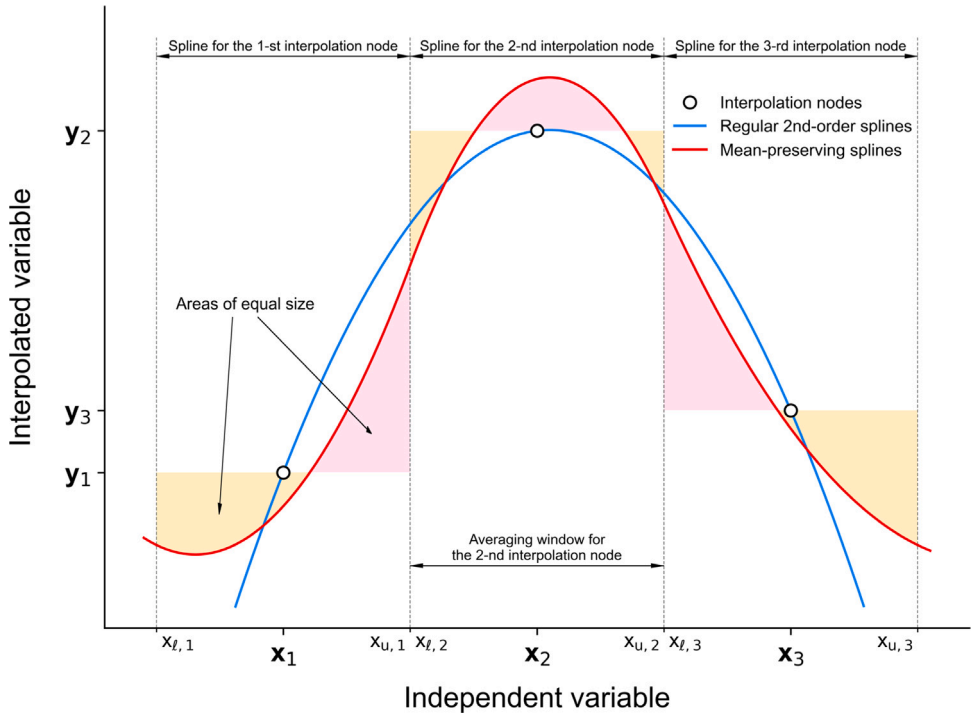

Fig. 1. Illustration example of the *mp*-splines interpolation working principles for three interpolation nodes (markers) with periodic boundary conditions. The color shaded areas below and above each mean-preserving spline have equal size to preserve the mean.

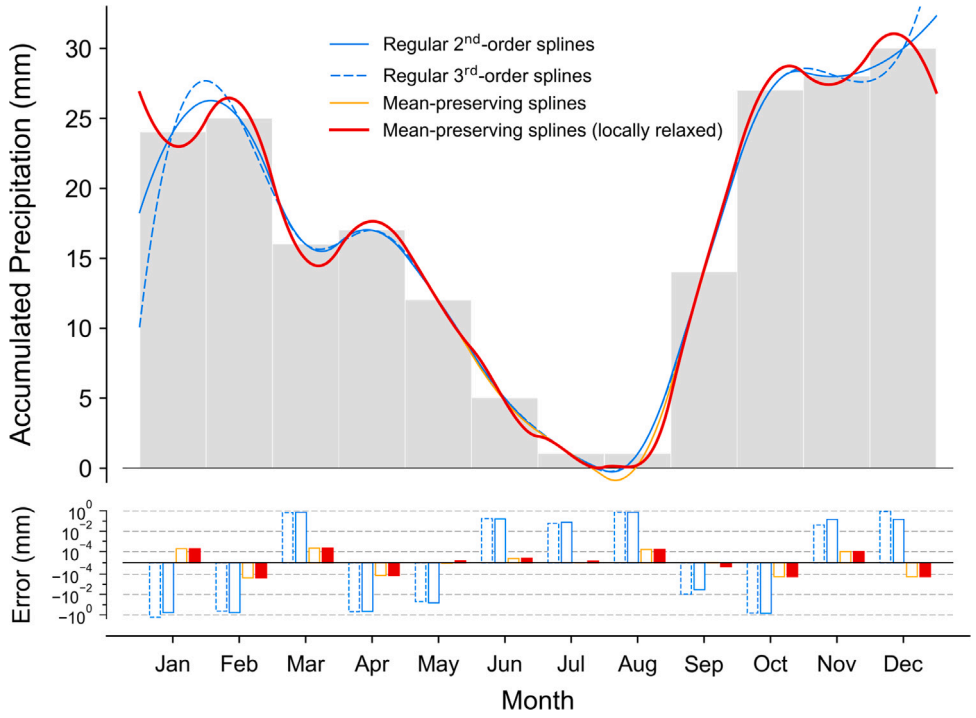

Fig. 2. Mean-preserving splines interpolation with local relaxation compared against regular splines interpolation and regular *mp*-splines. Data is a monthly climatology of rainfall at the Airport of Almería (Spain) retrieved from the Spanish Weather Agency website (http://www.aemet.es). The upper panel shows the actual data (bars) and the daily interpolated values with the various interpolation methods. The lower panel shows the differences between the source and the interpolated data in each case. Note the logarithmic scale in *y*-axis.

mp-splines interpolation, yield negative results for July and August. This situation is reversed using the local relaxation with *mp*-splines by setting a minimum allowed value of zero. In this particular example, the relaxation is enabled only for two neighboring splines (i.e., May through Oct) because there are only 12 interpolation nodes. Both *mp*-splines with and without local relaxation differ within the relaxed period but coincide beyond. The bottom panel shows the differences between the original monthly data and the ones averaged from the interpolated curves. The deviations with the regular splines interpolations are much greater than with *mp*-splines (note the logarithmic scale).

Indeed, the deviations with *mp*-splines are negligible, as expected from the mean preservation, less than 1 % in the worst case (August), while it is up to 100 % in the worst case for regular splines (August).

4. Python code

The *mp*-splines algorithm has been implemented in Python and is publicly available at https://github.com/jararias/mpsplines/. The library's interface follows a similar approach to that of `scipy.interpolate` (https://docs.scipy.org/doc/scipy/reference/interpolate.html). Particularly, first, an interpolation instance is created from the interpolation nodes and additional optional inputs (e.g., whether boundary conditions are periodic or not). Then, the instance is used as a regular function to perform the interpolations.

For illustration purposes, the code for the constrained interpolation of the monthly rainfall data shown in Fig. 2 is as follows:

```
# import required libraries and
    classes
import numpy as np
from mpsplines import (
        MeanPreservingInterpolation
             as MPI
)

# interpolation nodes data
xi = np.array(
[1, 2, 3, 4, 5, 6,
 7, 8, 9, 10, 11, 12])
yi = np.array(
[24, 25, 16, 17, 12, 5,
 1, 1, 14, 27, 28, 30])

# create the mp-splines
    interpolation instance
mpi = MPI(xi, yi, periodic=True,
    min_val=0.)

# load the interpolation targets
x = np.linspace(0.5, 12.5, 365)

# perform interpolation
y = mpi(x)
```

For detailed usage information and installation directions, see information at https://github.com/jararias/mpsplines. A brief performance analysis of *mp*-splines, including a comparison against other splines methods available in Python, is presented in Appendix.

5. Example use cases

This section illustrates the application and performance of *mp*-splines interpolation with three example use cases: sub-daily interpolation of clear-sky solar irradiance data (Section 5.1), daily-to-hourly interpolation of aerosol optical depth and precipitable water (Section 5.2), and daily interpolation of long-term monthly average ground albedo data (Section 5.3).

5.1. Interpolation of sub-daily clear-sky solar irradiance data

Global horizontal irradiance is available at hourly and coarser sub-daily time steps from multiple sources, such as observational networks (e.g., SIAR, 2022; MRCC, 2022; WRDC, 2022) and numerical weather datasets (e.g., Hersbach et al., 2018; MERRA-2 GHI, 2015), but they must be interpolated when solar applications require data at a finer time step. Depending on the intended application, the general case of sub-daily interpolations in cloudy or patchy-cloudy days, for which solar irradiance is expected highly variable, might require a method that not only preserves the mean, but also accounts for the fine-scale variability. However, simple interpolation methods, such as *mp*-splines, are suitable for overcast and cloudless days.

Table 1
Interpolation root mean squared difference (in Wm^{-2}) for various interpolation approaches (SP2: 2nd-order splines; SP3: 3rd-order splines; KT: KT-based linear; MPI: *mp*-splines) using averages of the 1-min GHI data shown in Fig. 3 for multiple time steps. The performance is evaluated against the source 1-min GHI data.

Time step	SP2	SP3	KT	MPI
1 h	3.6	3.5	2.1	2.0
2 h	11.8	11.5	6.6	4.3
3 h	31.6	32.1	11.6	20.8
4 h	40.3	32.8	17.6	19.2

This section benchmarks *mp*-splines against other methods by interpolating a 2-hr average GHI time series into 1-min time steps, as shown in Fig. 3. The regular 2nd-order splines interpolation is simple but results in deviations from the actual 1-min data at sunrise, noon and sunset. The linear interpolation of clearness index (that is, the ratio of GHI to extraterrestrial horizontal irradiance, KT), which is a customary approach in forecasting applications (Sengupta et al., 2021), improves the regular splines results, but it requires a transformation of GHI to KT, then a linear interpolation of KT into the 1-min time grid, and then transforming KT back to GHI. The *mp*-splines interpolation provides similar results, but slightly better at noon.

Table 1 shows the root mean squared difference of the previous interpolation approaches, plus regular 3rd-order splines, when they are used to interpolate the clear-sky GHI data shown in Fig. 3 averaged over multiple time steps, from 1 h to 4 h. The results reveal that the KT-based and *mp*-splines interpolation methods outperform regular splines for all time resolutions. The *mp*-splines interpolation even improves the results of the linear interpolation of KT for 1- and 2-hr time steps, but the latter is better for coarser time resolutions.

5.2. Daily-to-hourly interpolation of weather data

The simulation of clear-sky solar irradiance requires ancillary variables that are not necessarily available at the temporal resolution that is needed for solar irradiance. When these variables are defined as averaged amounts over a time period (e.g., hourly or daily), *mp*-splines arises as a logical interpolation approach to accommodate the input data to the time resolution required by the specific solar application. Fig. 4 shows interpolation results of hourly AOD data (MERRA-2 AOD, 2015) and hourly precipitable water data (MERRA-2 P.W., 2015) that have been first averaged to daily scale and then interpolated back to hourly time steps. The time series are taken from a Modern-Era Retrospective analysis for Research and Applications v2, (MERRA-2; Gelaro et al., 2017) grid-cell near the Penn State University SURFRAD site (SURFRAD, 2022), and span the entire year 2014. Two interpolation approaches are considered, regular 2nd-order splines and *mp*-splines. As a reference, Fig. 4 also shows the original data at hourly resolution. The results with the two interpolation approaches are barely distinguishable to the naked eye, at least at the zoom level shown in Fig. 4. As expected, the comparison of the interpolated hourly values against the original hourly data reveals no bias, and similar root mean squared deviations for the two methods, although slightly lower for *mp*-splines, particularly for precipitable water. Interestingly, these results also show evidence that none of the interpolation methods is able to reproduce the full sub-daily variability of the interpolated variables. This limitation is remarkable for daily maxima of AOD, that are not captured in the interpolated data at their full extent. This issue might be alleviated with a specialized approach that considered not only the average during the day, but maybe also a measure of the data spread (e.g., standard deviation). A more detailed analysis, not shown here, reveals that *mp*-splines is consistently better than regular 2nd-order splines for the 10 % of days with higher turbidity (0.99 RMSD for 2nd-order splines and 0.94 RMSD for *mp*-splines), while they behave alike for the rest. With precipitable water, *mp*-splines is always consistently better.

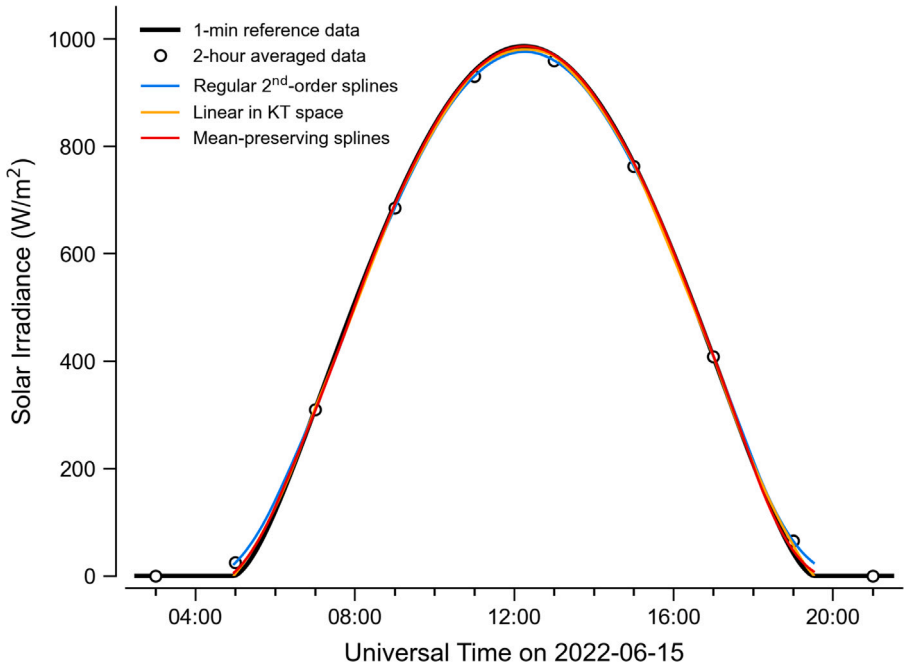

Fig. 3. Interpolation of 2-hr clear-sky average GHI data (markers) using three different interpolation methods: regular 2nd-order splines (blue), linear interpolation of KT (orange), and *mp*-splines (red). The reference 1-min data (black) from which the 2-hr averages are calculated is also shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

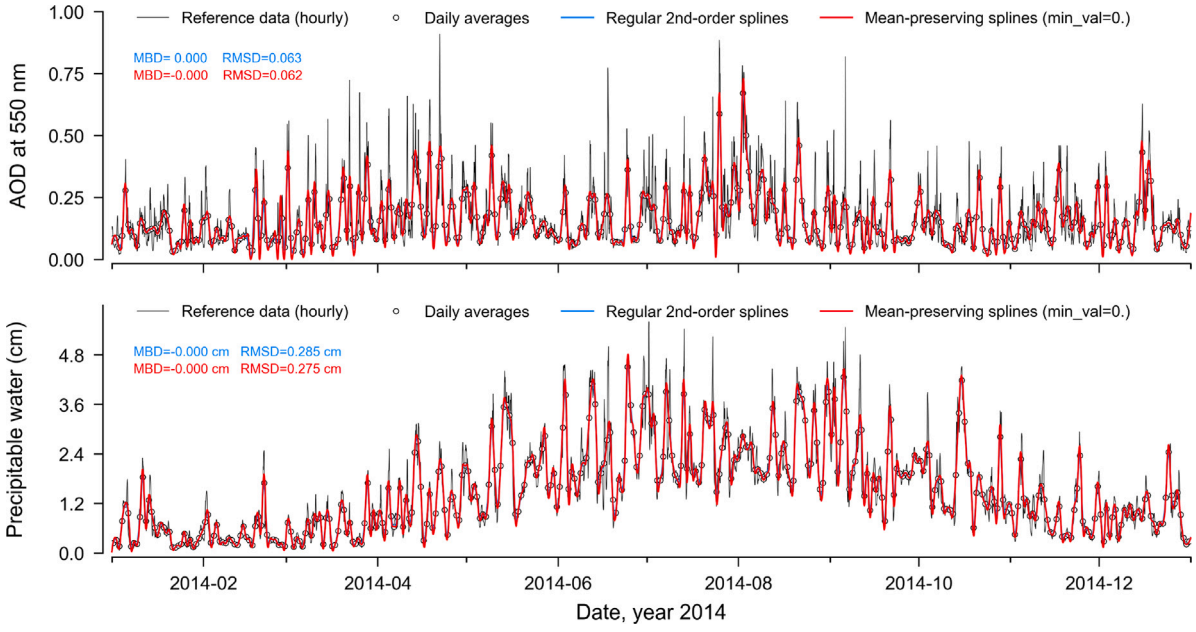

Fig. 4. Interpolation of daily averages of MERRA-2 AOD at 550 nm (top panel) and precipitable water (bottom panel) data to hourly time steps. The daily averages (markers) are calculated from hourly values, which are also shown for visual reference (black solid line). Two interpolation methods are considered, 2nd-order splines (blue solid line) and *mp*-splines (red solid line). The mean bias difference (MBD) and root mean squared difference (RMSD) of the interpolated data compared to the original hourly values are shown on the panels (blue for 2nd-order splines; red for *mp*-splines). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.3. Interpolation of long-term monthly average ground albedo

Sometimes weather variables are available only as a long-term average (LTA), maybe because they change only at long time scales, or because there is limited knowledge that prevents a finer representation. Some examples that are relevant to solar irradiance assessment are total-column ozone content and ground albedo. Long-term average databases are frequently represented at monthly scale, which naturally highlights their periodic boundary conditions.

Fig. 5 shows the interpolation of a monthly LTA version of the MERRA-2 ground albedo (MERRA-2 GHI, 2015) at daily steps throughout three consecutive years using both regular 2nd-order splines and *mp*-splines. The central year is highlighted in color to show the differential treatment of boundary conditions in the two interpolation methods, since *mp*-splines is provisioned with a specific mode for periodic boundary conditions, but regular splines is not. The library *mpsplines* has special functionalities for the interpolation of LTA data.

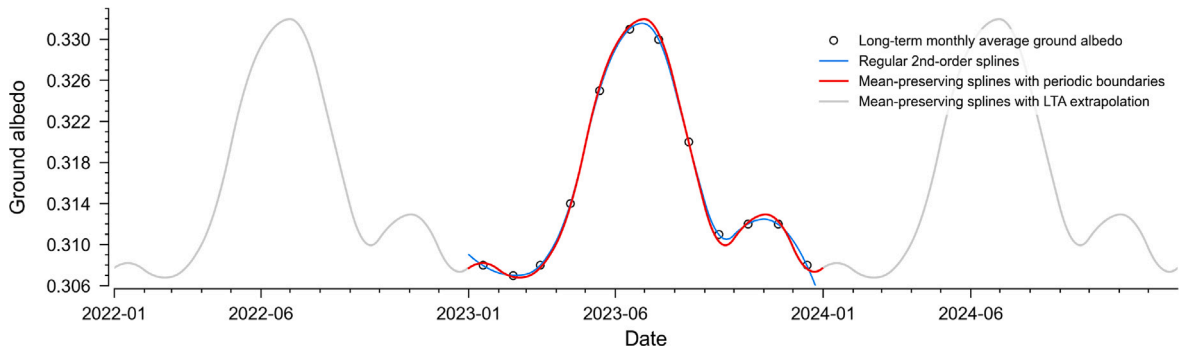

Fig. 5. Interpolation of MERRA-2 LTA monthly ground albedo data to daily time steps using regular 2nd-order splines and *mp*-splines. The LTA monthly time series is drawn from the MERRA-2 grid dataset at the Gobabeb BSRN site's location (−23.56140°N, 15.04198°E).

6. Conclusions

A general-purpose 1-D mean-preserving splines interpolation approach (*mp*-splines) has been presented. Unlike regular splines, it is suitable for averaged datasets and processes that require ensuring the conservation of mass or energy budgets during interpolation, that is a typical situation in the processes usually involved in solar modeling. However, as with regular splines, it can result in unphysical values (e.g., negative precipitation rates). To prevent it, *mp*-splines implements a local relaxation of the splines that yield unphysical values to readjust the interpolation and keep the values within the permitted domain. In addition, the algorithm can deal with data with periodic boundary conditions, for instance, with monthly climatologies.

The performance of the new interpolation method has been benchmarked against regular splines in three use cases. First, for the interpolation of sub-daily GHI averages during a cloudless day, where it proved superior to regular second- and third-order splines, and like linear interpolation in the KT space, for 1-hr and 2-hr average GHI. Second, for hourly interpolation of daily AOD and precipitable water data from an atmospheric reanalysis. The new interpolation approach proved better than regular second-order splines for precipitable water, and similar for AOD, except for the days with the highest turbidity, when *mp*-splines consistently improved regular second-order splines. Third, for long-term monthly average ground albedo, which is a dataset with periodic boundary conditions and thus, suitable for *mp*-splines.

CRediT authorship contribution statement

José A. Ruiz-Arias: Conceptualization, Methodology, Data processing, Coding, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data and code that support the findings of this study are openly available at https://github.com/jararias/mpsplines/tree/paper. See the file README.md there for a description of data and code usage.

Acknowledgments

This work was supported by the FEDER 2014–2020 Operative Program, and the Consejería de Economía *y* Conocimiento of Junta de Andalucía through the research project UMA20-FEDERJA-134, and the Agencia Estatal de Investigación through the research projects PID2019-107455RB-C21 and PID2019-107455RB-C22, Spain. The author is grateful to the scientists and personnel of the Spanish Meteorological Agency and the Global Modeling and Assimilation Office (GMAO) at NASA Goddard Space Flight Center who provided the data used in this study. The University of Málaga/CBUA provided the funding for open access.

Table 2
Execution time (milliseconds) of various interpolation methods for 1 year and 10 years of daily values interpolated to hourly and minutely time steps. Scipy splines refers to the function `scipy.interpolate.interp1d`, with *kind* being the splines' order. The figures in parentheses in the last row refer to the number of splines that violate min_val in *mp*-splines. The test cases are run in single-thread mode in a processor Intel Core i7-101710U, 1.10 GHz.

Source	1 year of daily values		10 years of daily values	
Target	Hourly	Minutely	Hourly	Minutely
scipy splines, kind=1	0.6	14.7	3.4	97
scipy splines, kind=2	0.9	21.1	4.6	163
scipy splines, kind=3	0.9	33.9	6.2	269
mp-splines, min_val=None	30	150	900	18305
mp-splines, min_val=0	875 (2)	996 (2)	9496 (20)	26957 (20)

Appendix. Comparative performance analysis

Although the current implementation of *mp*-splines is not aimed at meeting stringent performance criteria, this section shows a brief comparative performance analysis against splines interpolation methods implemented in Scipy (https://docs.scipy.org/doc/scipy/reference/interpolate.html). To that aim, the AOD dataset shown in Fig. 4 is used. The results are summarized in Table 2.

mp-splines is much slower than the splines implementation in Scipy and scales up less efficiently. This is expected because Scipy runs a wrapper over a Fortran implementation to optimize the performance. This is not the case with *mp*-splines, but it could be done in the future if foreseen applications require faster execution times. So far, the speed of the current implementation is deemed sufficient for most intended applications. Indeed, ten years of daily values are interpolated to hourly time steps in less than 1 s. This is equivalent to interpolate about 150 days of hourly values into 2.5 s time steps.

When *mp*-splines finds splines that yield values beyond the physical limits, the execution time blows up proportionally to the number of violating splines. Among the variables tested in this work, only AOD incurs routinely in un-physical values, but for very few splines. In the example shown in Table 2, only 2 splines per year yield un-physical values.

References

Beu, T.A., 2015. Introduction to Numerical Programming. A Pratical Guide for Scientists and Engineers Using Python and C/C++. CRC Press, Taylor and Francis Group, Boca Raton, FL, USA.

De Caceres, M., Martin-StPaul, N., Turco, M., Cabon, A., Granda, V., 2018. Estimating daily meteorological data and downscaling climate models over landscapes. Environ. Modell. Softw. 108, 186–196. http://dx.doi.org/10.1016/j.envsoft.2018.08.003.

Delhez, É., 2003. A spline interpolation technique that preserves mass budgets. Appl. Math. Lett. 16, 17–26. http://dx.doi.org/10.1016/S0893-9659(02)00139-8.

Gelaro, R., McCarty, W., Suárez, M.J., Todling, R., Molod, A., Takacs, L., Randles, C.A., Darmenov, A., Bosilovich, M.G., Reichle, R., et al., 2017. The modern-era retrospective analysis for research and applications, version 2 (MERRA-2). J. Clim. 30 (14), 5419–5454. http://dx.doi.org/10.1175/JCLI-D-16-0758.1.

Gupta, A.S., Tarboton, D.G., 2016. A tool for downscaling weather data from large-grid reanalysis products to finer spatial scales for distributed hydrological applications. Environ. Modell. Softw. 84, 50–69. http://dx.doi.org/10.1016/j.envsoft.2016.06.014.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, J.N., 2018. ERA5 hourly data on single levels from 1959 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). http://dx.doi.org/10.24381/cds.adbb2d47, Accessed 12 August 2022.

Inness, A., Ades, M., Agusti Panareda, A., Barré, J., Benedictow, A., Blechschmidt, A.M., Dominguez, J.J., Engelen, R., Eskes, H., Flemming, J., et al., 2019. The CAMS reanalysis of atmospheric composition. Atm. Chem. Phys. 19 (6), 3515–3556. http://dx.doi.org/10.5194/acp-19-3515-2019.

Knott, G.D., 2000. Interpolating Cubic Splines. In: Progress in Computer Science and Applied Logic, vol. 18, Birkhäuser, Boston, USA.

Kraft, D.A., 1988. A Software Package for Sequential Quadratic Programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany.

Lai, L.O., Kaplan, J.O., 2022. A fast mean-preserving spline for interpolating interval data. J. Atmos. Ocean. Tech. 39 (4), 503–512. http://dx.doi.org/10.1175/JTECH-D-21-0154.1.

MERRA-2 AOD, 2015. tavg1_2d_aer_Nx: 2d, 1-Hourly, Time-averaged, Single-Level, Assimilation, Aerosol Diagnostics V5.12.4. Global Modeling and Assimilation Office (GMAO), Greenbelt, MD, USA: Goddard Space Flight Center Distributed Active Archive Center (GSFC DAAC), http://dx.doi.org/10.5067/KLICLTZ8EM9D, (Accessed 12 August 2012).

MERRA-2 GHI, 2015. tavg1_2d_rad_Nx: 2d,1-Hourly, Time-Averaged, Single-Level, Assimilation, Radiation Diagnostics V5.12.4. Global Modeling and Assimilation Office (GMAO), Greenbelt, MD, USA: Goddard Space Flight Center Distributed Active Archive Center (GSFC DAAC), http://dx.doi.org/10.5067/Q9QMY5PBNV1T, (Accessed 12 August 2012).

MERRA-2 P.W., 2015. tavg1_2d_slv_Nx: 2d, 1-Hourly, Time-Averaged, Single-Level, Assimilation, Single-Level Diagnostics V5.12.4. Global Modeling and Assimilation Office (GMAO), Greenbelt, MD, USA: Goddard Space Flight Center Distributed Active Archive Center (GSFC DAAC), http://dx.doi.org/10.5067/VJAFPLI1CSIV, (Accessed 12 August 2012).

MRCC, 2022. Midwestern Regional Climate Center. https://mrcc.purdue.edu. (Accessed 12 August 2022).

Ruiz-Arias, J.A., 2020. Bias in modeled solar radiation by non-resolved intra-daily AOD variability. Sol. Energy 205, 221–229. http://dx.doi.org/10.1016/j.solener.2020.04.082.

Ruiz-Arias, J.A., Gueymard, C.A., 2018a. A multi-model benchmarking of direct and global clear-sky solar irradiance predictions at arid sites using a reference physical radiative transfer model. Sol. Energy 171, 447–465. http://dx.doi.org/10.1016/j.solener.2018.06.048.

Ruiz-Arias, J.A., Gueymard, C.A., 2018b. Worldwide inter-comparison of clear-sky solar radiation models: Consensus-based review of direct and global irradiance components simulated at the earth surface. Sol. Energy 168, 10–29. http://dx.doi.org/10.1016/j.solener.2018.02.008.

Ruiz-Arias, J.A., Gueymard, C.A., Cebecauer, T., 2019. Direct normal irradiance modeling: Evaluating the impact on accuracy of worldwide gridded aerosol databases. AIP Conf. Proc. 2126 (1), 190013. http://dx.doi.org/10.1063/1.5117710.

Rymes, M., Myers, D., 2001. Mean preserving algorithm for smoothly interpolating averaged data. Sol. Energy 71 (4), 225–231. http://dx.doi.org/10.1016/S0038-092X(01)00052-4.

Sengupta, M., Habte, A., Wilbert, S., Gueymard, C.A., Remund, J., 2021. Best Practices Handbook for the Collection and Use of Solar Resource Data for Solar Energy Applications: Third Edition, no. NREL/TP-5D00-77635. National Renewable Energy Laboratory, chapter 8.

SIAR, 2022. The agrolimatic information system for irrigation. https://datos.gob.es/en/blog/agroclimatic-information-system-irrigation-siar. (Accessed 12 August 2022).

Sun, X., Bright, J.M., Gueymard, C.A., Acord, B., Wang, P., Engerer, N.A., 2019. Worldwide performance assessment of 75 global clear-sky irradiance models using principal component analysis. Renew. Sust. Energ. Rev. 111, 550–570. http://dx.doi.org/10.1016/j.rser.2019.04.006.

Sun, X., Bright, J.M., Gueymard, C.A., Bai, X., Acord, B., Wang, P., 2021. Worldwide performance assessment of 95 direct and diffuse clear-sky irradiance models using principal component analysis. Renew. Sust. Energ. Rev. 135, 110087. http://dx.doi.org/10.1016/j.rser.2020.110087.

SURFRAD, 2022. Penn State University (PSU) station, SURFRAD Network, Global Monitoring Laboratory, National Oceanic and Atmospheric Administration. https://gml.noaa.gov/grad/surfrad/pennstat.html. (Accessed 12 August 2022).

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E.W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1. 0 Contributors, 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nat. Methods 17, 261–272. http://dx.doi.org/10.1038/s41592-019-0686-2.

Warner, J.C., Perlin, N., Skyllingstad, E.D., 2008. Using the model coupling toolkit to couple earth system models. Environ. Modell. Softw. 23 (10–11), 1240–1249. http://dx.doi.org/10.1016/j.envsoft.2008.03.002.

WRDC, 2022. World radiation data center. World meteorological organization. http://wrdc.mgo.rssi.ru/. (Accessed 12 August 2022).