

A N-dimensional Fortran Interpolation Program (NterGeo.v2020a) for Geophysics Sciences - Application to a back-trajectory program (BACKPLUMES.v2020r1) using CHIMERE or WRF outputs

Bessagnet, B., Menut, L., M.Beauchamps

5

<https://www.geosci-model-dev-discuss.net/gmd-2019-337/>

Dear Editor and reviewers,

We acknowledge the reviewers for the time spent to evaluate our work and for their minor revisions. We also acknowledge the Editor and we made all proposed changes in the revised manuscript. [Please note that answers are in blue and after each reviewer's remark.](#) When a large paragraph is added in the manuscript, it is here described in a grey box. 10

All reviewers remarks were taken into account and are detailed in this letter.

Text, references and Figures (captions and labels) were checked and corrected as requested.

Best regards,
Laurent Menut 15
July 15, 2020

1 Reviewer #1

This manuscript presents an useful tool to support the scientific analysis of atmospheric model outputs. This tool allows the estimation of back trajectories of plumes and it is directly linked to commonly-used regional atmospheric and chemistry-transport models, such as WRF or CHIMERE. The fact that the tool is directly linked to these models allows a total consistence between forward and backward estimates, as the wind field and grid are the same in both cases. The methodology is well described, with a clear and well-structured overall presentation. The code is available, through a link provided in the manuscript.

I strongly recommend the publication of this manuscript in GMD. Here some minor comments that I consider that could improve the manuscript (but not necessary for publication).

Answer:

We acknowledge the reviewer for these interesting comments. There is also some questions, certainly because our text was not clear enough. Some minor corrections are then put into the manuscript as suggested by the reviewer.

1. In page 36, the authors mention that BACKPLUMES is different than other back-trajectories models, such as Hysplit or Flexpart. Could the authors explain more the differences with the before-mentioned models? As the authors mention more processes than atmospheric motions, such as chemistry and deposition processes, can they be more precise, indicating which models consider those processes (further than only atmospheric motions)?

Answer:

There is no page 36 but we assume it is probably page 2 (introduction) or page 11. The paragraph describing the backplumes model was completely changed to answer these questions. The introduction is simplified (because it is not the place for a model description) and the 'backplumes' section is enriched with more details about the model. The new part in the introduction is:

In order to quantify the impact of such new interpolation program and show examples of its use, it is implemented in the back-tajjectory model Backplumes, developed in the same team than the CHIMERE model, Mailler et al. (2017). This host model is well dedicated for this implementation, because the most important part of its calculation is an interpolation of a point in a model grid box.

and the new paragraph for the presentation of the backplumes model is:

7.1 The Backplumes model

In order to test this new interpolation program, it is implemented in a backtrajectories model called "Backplumes". This model was already used in some studies such as (Mailler et al., 2016) and (Flamant et al., 2018) for example. Backplumes is open source and is available on the CHIMERE web site. Backplumes calculates backtrajectories from a starting point and a starting date. It is different from other 'backtrajectories' models, such as HYSPLIT (Stein et al., 2015), STILT (Lin et al., 2003), (Nehrkorn et al., 2010) and Flexpart (Pisso et al., 2019), because it is launching hundreds of passive tracers and plot as outputs all trajectories. Thus the answer is complementary compared to the other models: the output results is all possible trajectories, and not only the most probable.

An advantage of Backplumes for the WRF and CHIMERE users is that the code is dedicated to directly read output results of these models. Being developed by the CHIMERE developers teams, the code is completely homogeneous with CHIMERE in term of numerical libraries. Another advantage is that the code is very fast and calculates hundreds of trajectories in a few minutes. Using the wind fields of WRF or CHIMERE, and running on the same grid, the results of backtrajectories are fully consistent with the simulations done by the models.

The model is dedicated to calculate transport but not chemistry: only passive tracers are released. But a distinction could be made between gaseous or aerosol tracer: for the latter one,

scavenging is calculated to have a more realistic trajectory. The model is easy to use and light because a small set of meteorological parameters is required. These meteorological parameters are described in Table ?? for WRF and CHIMERE.

2. In the comparison with Hysplit, could the authors indicate if their methodology consider the same meteorological parameters?

Answer:

Yes, the same meteorological parameters are used and it is now explain in the new 'backplumes' section. 5

3. It would be appreciated to include a comment (or to highlight if already included; apparently it is not included) about the target pollutants, if used for chemistry-transport models; if back trajectories are mainly estimated considering atmospheric motions this code can be used mainly for non-reactive pollutants. 10

Answer:

Backplumes can only calculates the transport of passive tracers. It was added in the new paragraph.

4. The authors mention through the paper "particles". Please clarify this more (or if it is a general pollutant, not necessarily a particle)

Answer:

Yes, it was corrected by 'tracer'. 15

5. Could it be possible (not necessary for publication) to have an example of the comparison with Hysplit and Python for the WRF and CHIMERE applications? It could be useful for potential users.

Answer:

There is no comparison with Hysplit because this is not the same kind of trajectories which are calculated. But there is a comparison of 'Backplumes' used with WRF and CHIMERE. There is an interest to compare the same kind of calculation with two different forcings since the goal of this paper is to present a new interpolation algorithm. 20

References

- Flamant, C., Deroubaix, A., Chazette, P., Brito, J., Gaetani, M., Knippertz, P., Fink, A. H., de Coetlogon, G., Menut, L., Colomb, A., Denjean, C., Meynadier, R., Rosenberg, P., Dupuy, R., Dominutti, P., Duplissy, J., Bourriane, T., Schwarzenboeck, A., Ramonet, M., and Totems, J.: Aerosol distribution in the northern Gulf of Guinea: local anthropogenic sources, long-range transport, and the role of coastal shallow circulations, *Atmospheric Chemistry and Physics*, 18, 12 363–12 389, <https://doi.org/10.5194/acp-18-12363-2018>, <https://www.atmos-chem-phys.net/18/12363/2018/>, 2018. 25 30
- Lin, J. C., Gerbig, C., Wofsy, S. C., Andrews, A. E., Daube, B. C., Davis, K. J., and Grainger, C. A.: A near-field tool for simulating the upstream influence of atmospheric observations: The Stochastic Time-Inverted Lagrangian Transport (STILT) model, *Journal of Geophysical Research: Atmospheres*, 108, n/a–n/a, <https://doi.org/10.1029/2002JD003161>, <http://dx.doi.org/10.1029/2002JD003161>, 4493, 2003.
- Mailler, S., Menut, L., di Sarra, A. G., Becagli, S., Di Iorio, T., Bessagnet, B., Briant, R., Formenti, P., Doussin, J.-F., Gómez-Amo, J. L., Mallet, M., Rea, G., Siour, G., Sferlazzo, D. M., Traversi, R., Udisti, R., and Turquety, S.: On the radiative impact of aerosols on photolysis rates: comparison of simulations and observations in the Lampedusa island during the ChArMEx/ADRIMED campaign, *Atmospheric Chemistry and Physics*, 16, 1219–1244, 2016. 35
- Mailler, S., Menut, L., Khvorostyanov, D., Valari, M., Couvidat, F., Siour, G., Turquety, S., Briant, R., Tuccella, P., Bessagnet, B., Colette, A., Létinois, L., Markakis, K., and Meleux, F.: CHIMERE-2017: from urban to hemispheric chemistry-transport modeling, *Geoscientific Model Development*, 10, 2397–2423, 2017. 40
- Nehrkorn, T., Eluszkiewicz, J., Wofsy, S., Lin, J., Gerbig, C., Longo, M., and Freitas, S.: Coupled weather research and forecasting-stochastic time-inverted lagrangian transport (WRF-STILT) model, *Meteorology and Atmospheric Physics*, 107, 51–64, <https://doi.org/10.1007/s00703-010-0068-x>, 2010.

- Pisso, I., Sollum, E., Grythe, H., Kristiansen, N. I., Cassiani, M., Eckhardt, S., Arnold, D., Morton, D., Thompson, R. L., Groot Zwaaftink, C. D., Evangeliou, N., Sodemann, H., Haimberger, L., Henne, S., Brunner, D., Burkhart, J. F., Fouilloux, A., Brioude, J., Philipp, A., Seibert, P., and Stohl, A.: The Lagrangian particle dispersion model FLEXPART version 10.4, *Geoscientific Model Development*, 12, 4955–4997, [https://doi.org/10.5194/gmd-12-4955-](https://doi.org/10.5194/gmd-12-4955-2019)
5 2019, <https://www.geosci-model-dev.net/12/4955/2019/>, 2019.
- Stein, A. F., Draxler, R. R., Rolph, G. D., Stunder, B. J. B., Cohen, M. D., and Ngan, F.: NOAA's HYSPLIT Atmospheric Transport and Dispersion Modeling System, *Bulletin of the American Meteorological Society*, 96, 2059–2077, <https://doi.org/10.1175/BAMS-D-14-00110.1>, 2015.

Dear Editor and Anonymous reviewer,

We acknowledge the reviewer for the time spent to evaluate our work and for their important revisions which to improve the manuscript. We made most of proposed changes in the revised manuscript. Please note that answers are in blue and after each reviewer's remark or questions. The reviewer remarks were taken into account and are detailed in this letter. We have checked the English level.

Bertrand Bessagnet on behalf of Co-authors

Answers to interactive comment on “A N-dimensional Fortran Interpolation Program (NterGeo.v2020a) for Geophysics Sciences – Application to a back-trajectory program (BACKPLUMES.v2020r1) using CHIMERE or WRF outputs” by Bertrand Bessagnet et al.

Anonymous Referee #2

Received and published: 20 July 2020

Review for <https://www.geosci-model-dev-discuss.net/gmd-2020-88/gmd-2020-88.pdf>

The manuscript "A N-dimensional Fortran Interpolation Program (NterGeo.v2020a) for Geophysics Sciences – Application to a back-trajectory program (BACKPLUMES.v2020r1) using CHIMERE or WRF outputs" by Bertrand Bessagnet et al., describes an interpolation subroutine written in Fortran and its application to analysis of CHIMERE data. The topic lies within the scope of GMD, but the manuscript in the current state needs additional work in order to be considered for publication. The writing is not sufficiently clear, which makes the reading difficult. The current level of English is inadequate for publication in GMD. It needs to be revised and edited by a proficient speaker. It would be ideal if a native English speaker could language edit the manuscript.

Specific comments are listed below:

Title "A N-dimensional Fortran Interpolation Program (NterGeo.v2020a) for Geophysics Sciences - Application to a back-trajectory program (BACKPLUMES.v2020r1) using CHIMERE or WRF outputs" triggers a question. CHIMERE being an Eulerian off-line chemistry-transport model, it doesn't generate wind fields. Offline trajectory models require wind fields in order to run. In what sense BACKPLUMES uses CHIMERE

output? CHIMERE outputs are mainly composition fields. Does it mean that BACK-PLUMES uses CHIMERE interpolated meteorological fields from WRF?

CHIMERE is off-line and can work with any meteorological model. CHIMERE requires wind fields. In most cases they come from WRF but they could come from IFS (ECMWF). Of course if CHIMERE uses WRF it will be better to directly uses WRF to avoid interpolation issues.

Abstract

L4 "Fortran is a powerful and fast language, highly portable and easy to interface with other existing Fortran models." This phrase is problematic: Fortran is not a model. Most languages should allow interfacing code. Is the point that there are many existing geophysical models written in Fortran? It could be replaced by something like: "Fortran is a fast and powerful language highly portable. It is easy to interface models written in Fortran with each other."

Yes you are right we reformulate according your suggestion.

L5 "any Fortran compiler": did the authors tried all possible options? Why don't just say that is written in standard Fortran and tested with two compilers? In fact there are KIND statements and those can be dialect dependent.

Right, we have changed to "*Our program does not need any libraries, it is written in standard Fortran and tested with two usual compilers.*"

L6 "novel optional parameter" This is unclear: novel with respect to what? Optional with respect to what?

We have changed to "*A parameter (normalisation option) is provided...*"

L6 "For the general program, the inverse distance is used for the weight calculation with a distance defined as a p-distance". This phrase is undefined at this point. Which weights? What is p? The abstract should be self contained.

We have decided to remove this technical sentence

L8 "Moreover, a real case of geophysics application embedding this interpolation program is provided and discussed" Is it a "geophysical application"?

Yes we have changed accordingly

L8 "it consists in determining back-trajectories using atmospheric dispersion or mesoscale meteorological model outputs, respectively from the widely used models CHIMERE and WRF." Here "atmospheric dispersion" is unclear when applied to CHIMERE, which as far as I know is an Eulerian chemistry model itself. "Dispersion" usually refers to Lagrangian models. The phrase reads as : "NterGeo.v2020a determines back-trajectories using atmospheric dispersion (?) from CHIMERE and meteorology from WRF." Is that the intended meaning?

Yes "Dispersion" is not appropriate. We have changed to "*it consists in determining back trajectories using chemistry transport or mesoscale meteorological model outputs*"

L13 "used in its recent version V4.9.2 to propose horizontal and vertical interpolations" "used to propose interpolations" please revise the English expression.

We have changed to "*is commonly used in its recent version V4.9.2 for horizontal and vertical interpolations to manage climate models outputs*"

L17 "full coupling strategy": Full coupling of what?

We changed to "*of a full coupling strategy between these modules*"

L18 Is ISORROPIA the chemistry and thermodynamics module? then please place the reference at the end of the phrase. Or it is an interpolation routine?

ISORROPIA is a chemistry and thermodynamics module. Yes we have moved the reference at the end of the sentence.

L19 "In this case" do you mean in such cases? Is this related to what is said above? If so please do not open a new paragraph which implies a new topic.

Yes the reviewer is right, we have corrected.

L25 "In some studies," is too informal, please revise.

We have removed "*In some studies*,"

L31 (Scipy, 2014), unfortunately -> (Scipy, 2014),]. Unfortunately...

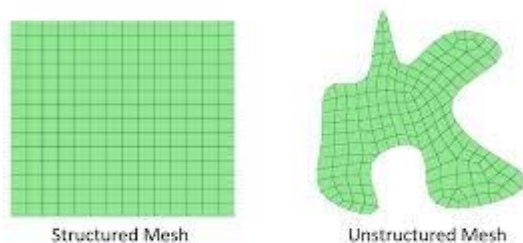
Right, we have corrected.

L32 "not enough optimized for our objective": Please make explicit which objective.

We have changed to "Unfortunately this program is not really adapted to our problem, it could be not enough optimized for our objective as it can manage fully unstructured datasets."

L33 "irregular (varying intervals) but structured" is unclear. What is the structure? Do you mean a square array?

Yes, the difference is a bit tricky to explain but usually in the community of 3D modellers it is quite well apprehended. In 2D, "Structured" means you have the same number of grid point number on y direction for a given x value (idem for the other direction). "irregular" means you can have different delta x that can depend on y coordinates.



L35 It is still unclear what the program precisely does and why the existing interpolation algorithms are not sufficient.

We have added this sentence to clarify : "*In short, the novelty of this program is to fill the gap of interpolation issues between the treatment of very complex unstructured meshes and simple regular grid for a general dimension N.*"

L36 "Atmospheric models (physics and/or chemistry) are commonly used in the Geophysics community." This phrase as it is contains little information. Mainly Lagrangian models are named later in the paragraph, maybe the authors mean "Lagrangian models (physics and/or chemistry) are commonly used in the Atmospheric Science community?"

L39 It is "atmospheric motion" not "motions".

This section has been rewritten according the suggestion of the previous reviewer and put later in the description of the back trajectory model.

L40 "A new back-trajectory plume has been developed" do you mean "A new back-trajectory plume model has been developed"?

L41 spatial interpolation of what? Please specify.

L41 "the other codes": please specify which codes.

L41 Unspecified: "some additional characteristics are implemented," please list the most important.

For these 4 comments, it has completely been rewritten as : *"In order to quantify the impact of such new interpolation program and show examples of its use, it is implemented in the back-trajectory model Backplumes, developed in the same team than the CHIMERE model (Mailler et al., 2017). This host model is well dedicated for this implementation, because the most important part of its calculation is an interpolation of a point in a model grid box. This paper describes (i) the methodology and the content of the interpolation program package NterGeo, and (ii) an application of this program embedded in the new "back-trajectory" program BACKPLUMES. These two codes are freely available (see code availability section)."*

Message of section 1: The introduction suggests that the aim is to interpolate 3d aerosol data on trajectories calculated with BACKPLUMES.

No in fact as it is written "In order to quantify the impact of such new interpolation program and show examples of its use, it is implemented in the back-trajectory model Backplumes, developed in the same team than the CHIMERE model". However, the program can be used for any application that requires to interpolate.

Section 2)

L51 "The program supports": which program, NterGeo?

Yes we have clarified like this *"The program NterGeo is fit for exploring irregular but structured grids or look-up tables defined by a unique size for each dimension which can be of course different from one to another dimension."*

here it is still not completely clear what "irregular structured" mean. This may suggest an arbitrary triangular mesh. But it appears that what is meant is that the nodes are in multidimensional grid array with varying edge lengths, i.e., there is an $N_1 \times \dots \times N_n$ array of cells that are hexahedrons with arbitrary trapezoids as faces.

See my answer above regarding this point.

L57 "As it includes not specific options or function, version of a compiler, there is no reason to have limitations or errors with other compilers." Which compilers did the authors try? Better simply to say that "for portability, the code is written in standard Fortran without elements from any particular dialect."?

Yes we have changed to *"The code does not need any libraries and is written in standard Fortran."*

L58 "The top shell calling script in the package provide two sets of options for ": provide*s* two sets of options...

Corrected

L58 "Assuming the X array, the result of the function f transforming X to Y array in R can be expressed as: " is confusing, please rewrite. It is stated that Y is a scalar, not an array, otherwise express the suitable space R^m .

Corrected we have changed Y to an array structure.

L62: Typography, please use `"\verb|"` or the verbatim environment and not italics for file names.

Corrected

L63 "a classical bilinear interpolation" before you mentioned arbitrary dimensions, do you mean multilinear?

Yes 'multilinear' is the right word.

L70 Style: please don't start a phrase with a symbol.

Yes we have corrected

L79 1-dimensionAL array

Corrected

Eq 5 please check and define all variables involved. Is the summation from $k=1$ or from $j=1$?

Corrected it was $j=1$

L85 "closest neighbours" or vertices of containing grid cell? It may be that the closest vertices correspond to a neighbouring cell

We have changed to "*closest vertice*"

L94 an hard coded option -> A hard coded option

Corrected

L99 in a 1D array. -> into a 1D array

Corrected

L125 "If the distance is too high compared to the characteristic" -> "If the distance is larger than the characteristic"...

Corrected

L141 "The time consuming is somehow proportional to the number of points in which to interpolate." This phrase is not grammatically correct and as stated does not add any important information. It is trivial that performing more operations takes longer time, the authors could replace "somehow" for something more precise.

We have changed to "As expected, the time consuming is obviously proportional to the number of points in which to interpolate."

L143 "There is a discontinuity of the NMSE from $p = 1$ to $p = 1+$ ": why?
I cannot explain this discontinuity that could be related to some rounding processes in fortran.

L144 "The NMSE decreases with the number of points but a slight increase is observed 145 from 200×200 from 300×300 ." why?

Increasing the number of points can produces noisy patterns, produce some instabilities that could impair the results.

L153 This could certainly depends -> This could certainly depend

Corrected

L175 "can use output files": what kind of data? CHIMERE and WRF output contains a very large number of variables, which are necessary?

This part has been rewritten thanks to reviewer 1 comments

"This kind of model has mainly one calculation to do several times" please revise this phrase. It is difficult to make sense out of it. Do you mean: This kind of models (Lagrangian) contain a time loop that performs many times the same kind of calculation?

This part has been rewritten thanks to reviewer 1 comments

L178 "The first advantage of BACKPLUMES is to use the results of a simulation already performed." do you mean that BACKPLUMES is an offline model?

This part has been rewritten thanks to reviewer 1 comments as *"An advantage of Backplumes for the WRF and CHIMERE users is that the code is dedicated to directly read output results of these models. Being developed by the CHIMERE developers teams, the code is completely homogeneous with CHIMERE in term of numerical libraries. Another advantage is that the code is very fast and calculates hundreds of trajectories in a few minutes."*

Yes BACKPLUME is offline, it can be run when CHIMERE AND WRF outputs are ready.

L179 "homogeneous with the "direct" simulation" do you mean consistent with the Eulerian forward simulation?

No it refers to the way the code is written. It has been rewritten in a new section. See herebelow.

L180 "BACKPLUMES is different than other back-trajectories models, such as Hysplit or Flexpart." In what is different?

It has been changed to *"In order to quantify the impact of such new interpolation program and show examples of its use, it is implemented in the back-trajectory model Backplumes, developed in the same team than the CHIMERE model, Mailler et al. (2017). This host model is well dedicated for this implementation, because the most important part of its calculation is an interpolation of a point in a model grid box."*

Please correct the English: different than other -> different to other

L181 "Since it is difficult to calculate correct probabilities back in time, the choice was made to randomly launch numerous trajectories and try to cover all possible origins." This phrase is very unclear, please clarify.

L181 Flexpart's goal is not to "estimate the most likely trajectory as an envelope of numerous possible trajectories

enveloppe -> envelope

L185: "The BACKPLUMES model is an opensource code and is available on the CHIMERE model web site." You should add a link with this statement. I looked in the website <https://www.lmd.polytechnique.fr/chimere/> and I couldn't find any reference to BACKPLUMES.

To answer all previous comments, the section has been completely rewritten as *"In order to test this new interpolation program, it is implemented in a backtrajectories model called "Backplumes". This model was already used in some studies such as (Mailler et al., 2016) and (Flamant et al., 2018) for example. Backplumes is open source and is available on the CHIMERE web site. Backplumes calculates backtrajectories from a starting point and a starting date. It is different from other 'backtrajectories' models, such as HYSPLIT (Stein et al., 2015), STILT (Lin et al., 2003), (Nehrkorn et al., 2010) and Flexpart (Pisso et al., 2019), because it is launching hundreds of passive tracers and plot as outputs all trajectories. Thus the answer is complementary compared to the other models: the output results is all possible trajectories, and not only the most probable. An advantage of Backplumes for the WRF and CHIMERE users is that the code is*

dedicated to directly read output results of these models. Being developed by the CHIMERE developers teams, the code is completely homogeneous with CHIMERE in term of numerical libraries. Another advantage is that the code is very fast and calculates hundreds of trajectories in a few minutes. Using the wind fields of WRF or CHIMERE, and running on the same grid, the results of backtrajectories are fully consistent with the simulations done by the models. The model is dedicated to calculate transport but not chemistry: only passive tracers are released. But a distinction could be made between gaseous or aerosol tracer: for the latter one, scavenging is calculated to have a more realistic trajectory. The model is easy to use and light because a small set of meteorological parameters is required. These meteorological parameters are described in Table \$\$ for WRF and CHIMERE."

L186-189 This paragraph is useful but incomplete. Since it is not described before (although it has been used in a couple of publications) you should specify better the numerics of the model.

This paragraph is just a summary of all useful variables. Then, their interest and use is explained later in the text (lines 215 for the vertical mixing for example). But, for clarity, the whole section was rewritten and a large part of explanations, not really mandatory for this paper about the interpolation routine, are put in Appendix.

L192 " The number of trajectories is a up to the user and may be from one to several hundred of tracers.": "tracers" usually denote chemical species. Do you mean particles?

Yes we have changed to "air particles" but also kept the concept of tracer.

L193 This paragraph starts with a discussion of the time step and the CFL condition and then jumps to the WRF specific calculation of the z coordinate. It is difficult to follow the logic and has many English language errors. Please rewrite paying attention to the logical order in the definition of the variables for the equations.

We have changed to "In order to respect the Courant Friedrich Levy (CFL) number, a sub-time step may be calculated. If the input data are hourly provided (as in many regional models), the meteorological variables are interpolated between the two consecutive hours to obtain refined input data."

L202 "the horizontal grid cell" do you mean dependent on the horizontal coordinate?

Eq 15: does 1 correspond to the lowest level?

It corresponds to the ground level pressure. Then we have changed to "on the first level grid"

L207 " is eta values on full (w) levels " please revise the English.

We have changed to "where $\eta_{\{M\}}$ is its value on full (w) levels (ZNW) and $\eta_{\{F\}}$ is the eta value on half (mass) levels (ZNU)".

"The layer thicknesses varying in space but also in time, this calculation is done for all trajectories and all time-step. " please revise the English.

We have changed to "The layer thicknesses is space and time dependent, this calculation is performed for all trajectories and all time-step"

L200-208: This section is very unclear. You have to better explain the rationale of the log pressure vertical interpolation and possibly move the technical details (maybe expanded) to an appendix.

The use of pressure as vertical coordinate is classical in meteorological modeling. And when you have to interpolate between two altitudes, it is better to linearize the values using the log of the pressure before to use a linear interpolation. The reference to [Pielke, 1984] was added being the best way to see this point. Some sentences were added to clarify the text: the calculations are done in pressure vertical coordinates, but the input (the altitude of the starting point) and the outputs (the backplumes) are in

meters. Following the reviewer suggestion, some technical details are moved to an appendix.

Figure 5: "Methodology for the vertical distribution as a function of the diurnal cycle of the boundary layer height." Please define the circles, squares and the black thick line in the caption.

The Figure was removed because the explanation in the text is enough.

L210-215: How do you know that this Euler scheme is accurate enough?

The question is not clear

L219-220 There is also mixing in a neutrally stable boundary layer, even in the absence of thermal convection. How do you treat such cases?

This case is not treated. But it could be in a future version, the reviewer is right.

L225 "Therefore a random function is applied..." Please define Rand.

Yes, the definition is now in the manuscript

L228 "It is considered that 15 mn is representative of a well-mixed convective layer." Why? Can you add e.g. a reference?

Yes, it is in many articles about thermals and convective mixing. We prefer here to add the reference book explaining that, [Stull, 1988].

L233 "its possible vertical motion with values between 0 and $w/2$ m s⁻¹." Where do these values come from?

Same reference, [Stull, 1988].

L237 Define Rand (as all symbols) the first time is mentioned.

Yes, it was done

The treatment of mixing is very simplistic. The well mixed condition is not even mentioned. Although simplified approaches may find application in certain situations, its limitations and shortcomings should be better presented and discussed. There is abundant literature on Lagrangian methods in the atmosphere to refer to.

We don't think it is simplistic: it is statistical. And it is the key point of this little backplumes model. It is only the assumption that, in a well convective mixed layer, you don't know at what altitude the tracer was within the boundary layer and 15mn ago.

L239 "The difference between the two models is 240 the number of vertical levels." How many in each case?

It depends on the user's choice when he made the simulation. But in our case, it was added in the text. But it is just an example, it is not very important regarding the goal of the paper

L241 "The wind field is also the same, WRF sending this information to CHIMERE." Please rewrite this phrase revising the English.

Ok, the sentence was replaced by: "The wind field is the same for the two models: CHIMERE using directly the wind field calculated by WRF."

L241 " WRF using the (Hong et al., 2006) schemes and CHIMERE using the (Troen and Mahrt, 1986) scheme." Are there many Hong et al. schemes? Please use \citet and not \citep in order to place the parenthesis in the right place.

Ok it was done

L245 Please use a proper LaTeX degree symbol.

\circ was replaced by \circ in the whole manuscript.

L259 But, in both cases, the answer in our case is clearly that the main contribution of the air masses located at the starting point are mainly coming from the North." There is very little quantitative information in this statement.

Yes, it is right. Mainly because the spread between the two models is not the main goal of this article. We do not want to present a geophysical analysis of the result but just illustrate the fact that the new interpolation scheme works well with two different models. However, the paragraph was updated to present more details with: "This illustrates the sensitivity of the result to the forcing model. But, in both cases, the answer in our case is clearly that the main contribution of the air masses located at the starting point are mainly coming from the North-East, crossing Tunisia, then the Mediterranean Sea and Europe. The main difference between the two calculations is the eastern part of the plume, more intense with WRF than CHIMERE."

L267 "This interpolation program can be used for any application in Geophysics and Engineering Sciences but also to explore large structured matrices for Machine Learning applications." This conclusion is not supported by the main body of the manuscript. Especially the mention to Machine Learning. There are no results nor discussion of this topic. Please rework it in the discussion or remove from the conclusions.

The reviewer is right we have removed the mention to Machine Learning.

Answers to Interactive comment on Geosci. Model Dev. Discuss.,
<https://doi.org/10.5194/gmd-2020-88>, 2020.

A N-dimensional Fortran Interpolation Program (NterGeo.v2020a) for Geophysics Sciences - Application to a ~~back-trajectory~~ back trajectory program (~~BACKPLUMES~~ Backplumes.v2020r1) using CHIMERE or WRF outputs

Bertrand Bessagnet^{1,2}, Laurent Menut¹, and Maxime Beauchamp³

¹LMD/IPSL, École Polytechnique, Institut Polytechnique de Paris, ENS, PSL Université, Sorbonne Université, CNRS, 91128 Palaiseau, France

²Citepa, Technical Reference Center for Air Pollution and Climate Change, 42, rue de Paradis 75010 Paris, France

³IMT Atlantique, Lab-STICC UMR CNRS, 655 Avenue du Technopôle, 29280 Plouzané, France

Correspondence: Bertrand BESSAGNET (bertrand.bessagnet@lmd.polytechnique.fr)

Abstract. An interpolation program coded in Fortran for irregular N-dimensional cases is presented and freely available. Needs of interpolation ~~procedure~~ procedures over irregular meshes or matrixes with interdependent input data dimensions is frequent in geophysical models. Also, these models often embed look-up tables of physics/chemistry modules. Fortran is a ~~powerful and fast language, highly portable~~ and fast and powerful language highly portable. It is easy to interface ~~with other existing~~ Fortran models ~~models written in Fortran with each other~~. Our program does not need any libraries ~~and can be compiled with any Fortran compiler, it is written in standard Fortran and tested with two usual compilers~~. The program is fast and competitive compared to current Python libraries. A ~~novel optional parameter (normalisation option)~~ normalisation option parameter is provided when considering different types of units on each dimension. ~~For the general program, the inverse distance is used for the weight calculation with a distance defined as a p-distance~~. Some tests and examples are provided and available in the code package. Moreover, a ~~real case of geophysics~~ geophysical application embedding this interpolation program is provided and discussed, it consists in determining ~~back-trajectories using atmospheric dispersion~~ back trajectories using chemistry transport or mesoscale meteorological model outputs, respectively from the widely used models CHIMERE and WRF.

1 Introduction

Interpolation is commonly used in geophysical sciences for post-treatment ~~operations~~ processing to evaluate model performances ~~at point against ground station~~ observations. The NCO library (?) is commonly used in its recent version V4.9.2 ~~to propose for~~ horizontal and vertical interpolations to manage climate models outputs. The most frequent need is to interpolate in 3D spatial dimension and time therefore in 4 dimensions. Fortran is ~~used extensively for weather and climate-related software~~ extensively used for atmosphere modelling softwares (? , e.g. WRF - ?; GFDL AM3 - ?). ~~Geophysical~~ More generally, geophysical models can use look-up tables of complex modules instead of a full coupling strategy between these modules which

20 is the case of the CHIMERE model (?) with the embedded ISORROPIA (??)-module dealing with chemistry and thermodynamics -

~~In this case(??). In such case,~~ the look-up table can easily exceed 5 dimensions to approximate the model. In parallel, Artificial Intelligence methods are developed and can explore the behaviour of complex model outputs that requires fast interpolation methods. While more recent modern languages like Python are used in the scientific community, Fortran remains widely in
25 the geophysics / engineering community and is known as one of the faster language in time execution, very good on array handling, parallelisation and above all portability. Some benchmarks are available on web site to evaluate the performances of languages on simple to complex operations (?).

~~In some studies, the~~The parameterization techniques proposed to manage aerosol/droplet microphysical schemes (?) can employ either the modified Shepard interpolation method (?) or the Hardy multiquadrics interpolation method (??), and the
30 numerical results obtained show that both methods provide realistic results for a wide range of aerosol mass loadings. For the climate community, a comparison of six methods for the interpolation of daily European climate data are proposed by (?), some of these methods use ~~kriging-like~~ kriging-like methods with the capability to use co-predictors like the topography.

A python procedure called *scipy.interpolate.griddata* is freely available (?), ~~unfortunately.~~ Unfortunately this program is ~~too~~
~~general handling fully unstructured datasets and then not~~ not really adapted to our problem, it could be not enough optimized
35 for our objective as it can manage fully unstructured datasets. The goal of this paper is to present a program to interpolate in a grid or a matrix which can be irregular (varying intervals) but structured with the possibility to have interdependent dimensions (e.g. a longitude interval edges which depend on longitude, latitude, altitude and time). We think this type of program can be easily implemented within models or to manage model outputs for post-treatment issues. In short, the novelty of this program is to fill the gap of interpolation issues between the treatment of very complex unstructured meshes and simple regular grids
40 for a general dimension N.

~~Atmospheric models (physics and/or chemistry) are commonly used in the Geophysics community. Among all existing models, HYSPLIT (?), STILT (?) and its WRF coupled version, (?) and Flexpart (?) are widely used. These models have different levels of complexity and are able to transport backward in time air masses by accounting for atmospheric motions, chemistry and deposition processes.~~

~~A new back-trajectory plume has been developed taking advantage of this new interpolation program used to perform a 3D spatial interpolation at each time step. Compared to the other codes some additional characteristics are implemented, the code is very light and does not required a lot of computer resources and libraries. This code is also very fast and enables to calculate numerous trajectories in a few minutes. Finally, and probably the most important point, the code is dedicated to run on the models outputs already calculated: trajectories are thus estimated over exactly the same grid than the one used for the direct
50 Eulerian simulation. This model is thus particularly suitable for users already having an Eulerian simulation and who want supplementary information about their studied case.~~

In order to quantify the impact of such a new interpolation program and show examples of its use, it is implemented in the back-tajjectory model Backplumes, developed in the same team than the CHIMERE model, ?. This host model is well fit for this implementation, because the most important part of its calculation is an interpolation of a point in a model grid box. This

55 paper describes (i) the methodology and the content of the interpolation program package **NterGeo**, and (ii) an application of this program embedded in the new ~~"back-trajectory"-program~~ [back trajectories program](#) **BACKPLUMES**[Backplumes](#). These two codes are freely available (see code availability section).

2 Development of the interpolation program

The program ~~supports the exploration of~~ [NterGeo is fit for exploring](#) irregular but structured grids or look-up tables defined by
60 a ~~size in unique size for~~ each dimension which can be of course different [from one to another dimension](#). The space intervals can vary along a dimension and the grid interval edges in each dimension can depend on other dimensions. Two versions have been developed, (i) a version for a "regular" arrays with independent dimensions and, (ii) a "general" version for possible inter-dependent dimensions, e.g. to handle 3D meshes which have time varying spatial coordinates. The code does not need any libraries and ~~can be easily compiled with any Fortran compiler is~~ [written in standard Fortran](#). Our interpolation code was
65 tested with [gfortran-gfortran](#) (GNU Fortran project) and [ifort-ifort](#) (Intel). As it includes not specific options or function, version of a compiler, there is no reason to have limitations or errors with other compilers. The top shell calling script in the package ~~provide~~ [provides](#) two sets of options for "production" and "debugging" modes. Assuming the X array, the result of the function f transforming X to Y array in \mathbb{R} can be expressed as:

$$Y(\underline{x_1, \dots, x_N}) = f(X(x_1, \dots, x_N)) \quad (1)$$

70 N is the dimension of the array, x_i is the coordinates at dimension $i \in [1, N]$ of the point X we want to interpolate.

2.1 The program for regular grids

A program ~~interpolation_regular.F90~~ [interpolation_regular.F90](#) for regular grids (i.e. with independent dimensions) is available. To handle this type of grids a classical [bilinear-multilinear](#) interpolation is performed. Figure 1 shows the variables for $N = 3$ defined hereafter in the section.

75 For the particular case of a regular grid with independent dimensions the result \tilde{Y} of the [bilinear-multilinear](#) interpolation of the 2^N identified neighbours can be expressed as:

$$\begin{aligned} \tilde{Y} = & w_N^0 \dots w_i^0 \dots w_1^0 \times Y_0(0 \dots 0 \dots 0) \\ & + w_N^0 \dots w_i^0 \dots w_1^1 \times Y_1(0 \dots 0 \dots 1) \\ & + \dots \\ & + w_N^{\delta_N} \dots w_i^{\delta_i} \dots w_1^{\delta_1} \times Y_k(\delta_N \dots \delta_i \dots \delta_1) \\ & + \dots \\ & + w_N^1 \dots w_i^1 \dots w_1^1 \times Y_{2^N-1}(1 \dots 1 \dots 1) \end{aligned} \quad (2)$$

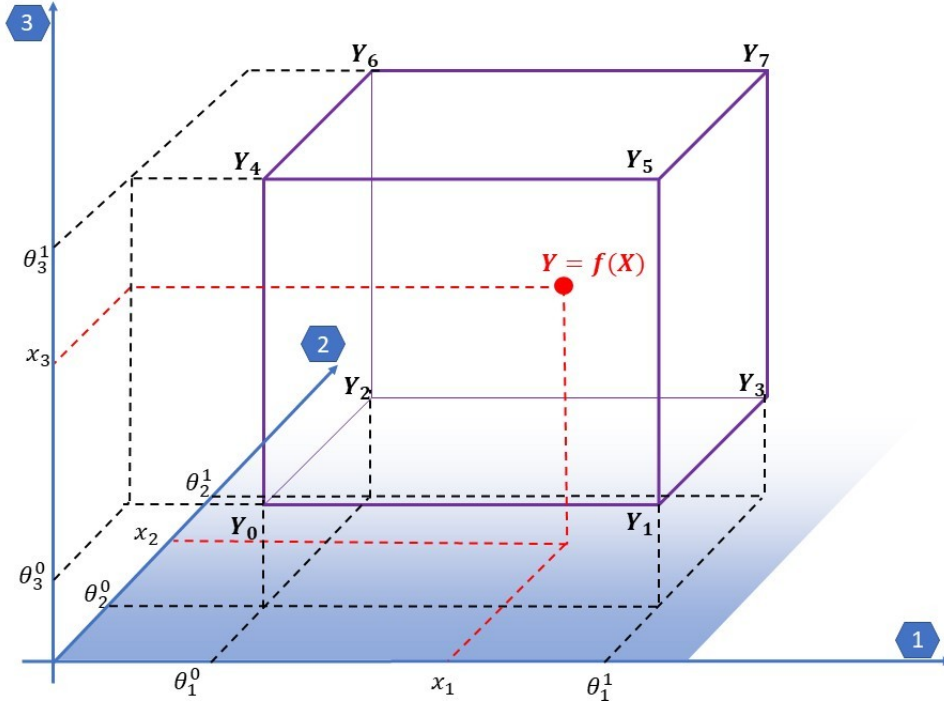


Figure 1. Description of variables for $N=3$ with a regular grid case

with δ_i the binary digit equal to 0 or 1, and the weights $w_i^{\delta_i}$ for $i \in [1, N]$ defined as:

$$w_i^0 = \frac{\theta_i^1 - x_i}{\theta_i^1 - \theta_i^0} \quad w_i^1 = 1 - w_i^0 \quad (3)$$

80 **Variable** Θ_i is the list of interval edges on each dimension i and does not depend on other dimensions. $\theta_i^{\delta_i}$ is the bottom ($\delta_i = 0$) and top ($\delta_i = 1$) edges on each dimension $i \in [1 \dots N]$ so that $x_i \in]\theta_i^0, \theta_i^1]$. Y_k is a 1-dimensional array with 2^N elements storing the value Y of the function at the identified neighbours Ψ on each dimension:

$$Y_k(\delta_N \dots \delta_i \dots \delta_1) = f(\Psi(\theta_N^{\delta_N}, \dots, \theta_i^{\delta_i}, \dots, \theta_1^{\delta_1})) \quad (4)$$

with $k \in [0, 2^N - 1]$

85 The tuple $(\delta_N \dots \delta_i \dots \delta_1)$ is the binary transformation of integer k defined as $\sum_{i=0}^{N-1} (\delta_i \times 2^i)$. The coefficients $\Gamma_k = w_N^{\delta_N} \dots w_i^{\delta_i} \dots w_1^{\delta_1}$ as a product of weighting factors on each direction can be seen as a binary suite that is convenient to handle in a compacted and optimized Fortran programming strategy for the *regular* grid version of the code (Appendix B).

2.2 The general program

Considering the *general* program called `interpolation_general.F90` `interpolation_general.F90`, the coordinates
90 of edge points are stored in a 1-dimension-1-dimensional array of $n = \prod_{i=1}^N I_i$ elements with I_i the number of edges on each

dimension i . The tuple of coordinates (j_1, \dots, j_N) of an interval edge θ_k^i , with j_i the indexed coordinate on dimension i , is transformed in a **1-dimension 1-dimensional** array indexed on $k \in [1, n]$ by:

$$k = \sum_{\substack{k=1 \\ j=1}}^N \left((i_j - 1) \prod_{l=0}^{j-1} I_l \right) + 1 \quad (5)$$

with $I_0 = 1$ for initialisation.

- 95 Once the nearest neighbour is found, the result \tilde{Y} of the interpolation is a weighting procedure of the 2^N closest **neighbours** **vertice** using a Shepard interpolation (?) based on the inverse distance calculations:

$$\tilde{Y} = \sum_{k=0}^{2^N-1} (\Gamma_k \times Y_k) \quad (6)$$

with $Y_k = f(\Upsilon_k)$ the value of the function f at neighbour Υ_k of coordinates $(\theta_k^1, \dots, \theta_k^N)$, and:

$$\Gamma_k = \frac{1/d_k}{\sum_{k=0}^{2^N-1} (1/d_k)} \quad (7)$$

- 100 The distance d_k between the point of interest of coordinates (x_1, \dots, x_N) to the neighbour $k \in [1, n]$ is calculated as:

$$d_k = \left(\sum_{i=1}^N |x_i - \theta_k^i|^p \right)^{\frac{1}{p}} \quad (8)$$

The previous formula are valid for $d_k \neq 0$, in the case of $d_k = 0$ the procedure stops and exit returning the exact value of the corresponding data of the nearest neighbour. For distorted meshed or matrix, or dimensions with different units (e.g mixing time with length), **an a** hard coded option $norm = .true.$ or $.false.$ is also available to normalize the intervals with an average

- 105 interval Δ_i value for the calculation of distances so that:

$$d_k = \left(\sum_{i=1}^N \left(\frac{|x_i - \theta_k^i|}{|\Delta_i|} \right)^p \right)^{\frac{1}{p}} \quad (9)$$

3 Computation strategy for the general program

The list of input/output arguments is provided in Appendix C. In the main program calling the subroutine the key point is to transform first the N-dimension matrix **in into** a 1D array. An example of **main program a main program calling the subroutine**

- 110 is provided in the code package. The computation strategy in the subroutine can be broken down into the sequential steps as follows:

- (I) Find the nearest neighbour of the input data by minimizing a distance with a simple incremental method stepping every ± 1 coordinates on each dimension (detailed later in this section).
- 115 (II) Scan the surroundings of the nearest point within the matrix on ± 1 step on each dimension and store the corresponding block of input data to be tested. The size of the block is therefore $(1 + 2 \times 1)^N$ but can be extended to $(1 + 2 \times 2)^N$ if we increase the scanning process to ± 2 on each dimension (hard coded option *iconf*=1 or 2 in the declaration block).
- (III) Calculate the distance to the previously selected input data. A *p*-distance concept is adopted (hard coded option *pnum* in the declaration block). The *pnum* value *p* should be superior or equal to 1 to verify the Minkowski inequality and be considered as a metric.
- 120 (IV) Sort the previous block of data in ascending order and stop the sorting process when the first 2^N point are selected. The code offers the possibility to use only the first $N + 1$ neighbours (hard coded option *neighb* in the declaration block) that is sufficient and faster in most cases.
- (V) Calculate the weights_{*x*} and then the final result.

The first step consisting in finding the first neighbour is the trickiest and is broken down into several steps. Figure 2 displays
 125 an example in 2D of the step by step procedure to find the nearest neighbour.

- (i) The procedure initializes the process starting from the first point of the input data grid or taken from the last closest point if given in argument as a non-null value.
- (ii) A delta of coordinates is applied based on an average delta on each dimension to improve the initialisation. This computation step of delta is externalized as it can be time consuming and should be done once for all target points at which we
 130 want to interpolate.
- (iii) A test between the target value and the input data grid points coordinates determines the ± 1 steps to add on each dimension (see Figure 2 for an example in 2D).
- (iv) If the grid point falls on the edges or outside the borders the closest coordinates within the matrix is selected.
- (v) A test on the ~~p-distance~~ p-distance computation between the running point and the target is performed so that if the
 135 distance calculated at Iteration N_{it} is equal to distance at Iteration $N_{it} - 2$ the closest point is found.
- (vi) If the distance is ~~too high compared to~~ larger than the characteristic distance of the cell, the point is considered to be outside the borders of the input grid data. Therefore, the code allows a slight extrapolation if the target point is not too far from the borders.
- (vii) At this stage, the procedure can stop if the distance to the closest ~~neighbour~~ vertice is 0 returning to the main program
 140 with the exact value of the input data grid.

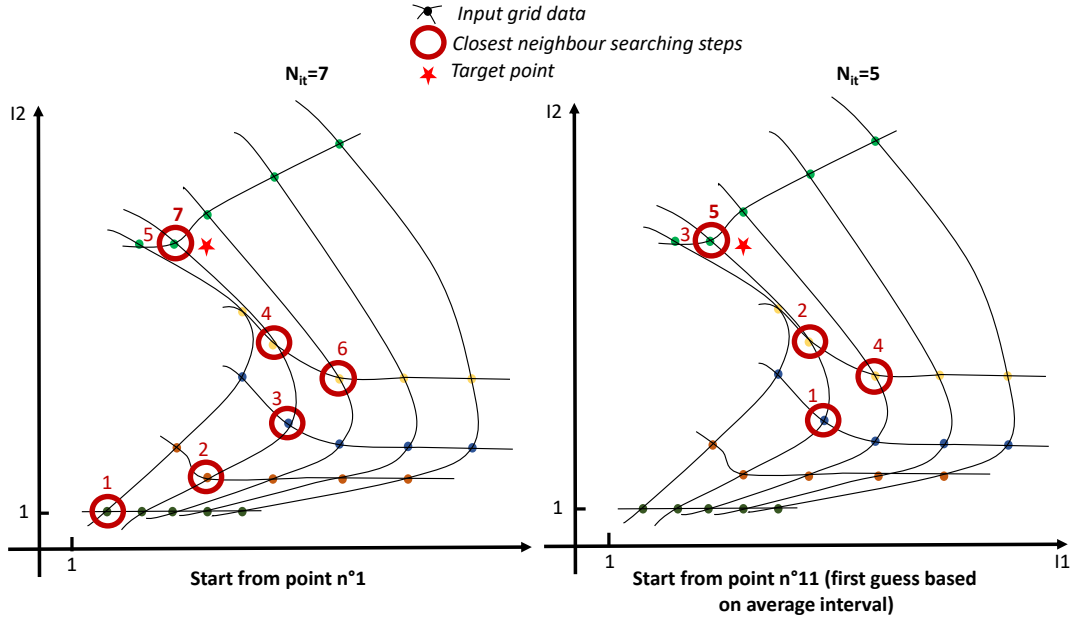


Figure 2. Real example in 2D of the step by step procedure to find the nearest neighbour of a target point for an irregular but structured 5×5 grid. Left panel when starting the process from the 1^{st} point of the grid on the lowest left corner, Right panel when starting with a first guess based on an average delta computed for each dimension.

4 Visual example in 2D for a regular grid

As an example to visualize the capacity of the *general* program, the 2D function used in ? is used to test our procedure. The function is:

$$Y = f(X) = x_1 \times (1 - x_1) \times \cos(4\pi x_1) \times \sin(4\pi x_2^2)^2 \quad (10)$$

145 with $x_1, x_2 \in [0, 1]$.

Our input grid data is a regular grid with regular intervals of 0.02 from 0 to 1 for x_1 and x_2 with therefore 51 points on each dimension. We propose to interpolate on a finer regular grid with $n=100 \times 100$, 200×200 and 300×300 points on each dimension. For these three interpolations cases a normalized root mean square error (NMSE) of the result \tilde{Y}_j for the full grid point number j can be calculated against the true value Y_j of the function as:

$$150 \quad NMSE = \frac{\frac{1}{n} \sum_{j=1}^n (\tilde{Y}_j - Y_j)^2}{\frac{1}{n-1} \sum_{j=1}^n (Y_j - \bar{Y}_j)^2} \quad (11)$$

with \bar{Y}_j the mean value Y_j as $\frac{1}{n} \sum_{j=1}^n Y_j$.

For the three cases the CPU time for the interpolation is evaluated and displayed in Table 1 for Machine 1 (Appendix E). **The** As expected, the time consuming is somehow obviously proportional to the number of points in which to interpolate. Figure 3 displays the evolution of the NMSE with the parameter p of the p-distance p-distance definition. There is a discontinuity of the NMSE from $p = 1$ to $p = 1^+$ with a slight increase with p in an asymptotic way. The NMSE decreases with the number of points but a slight increase is observed from 200×200 from 300×300 .

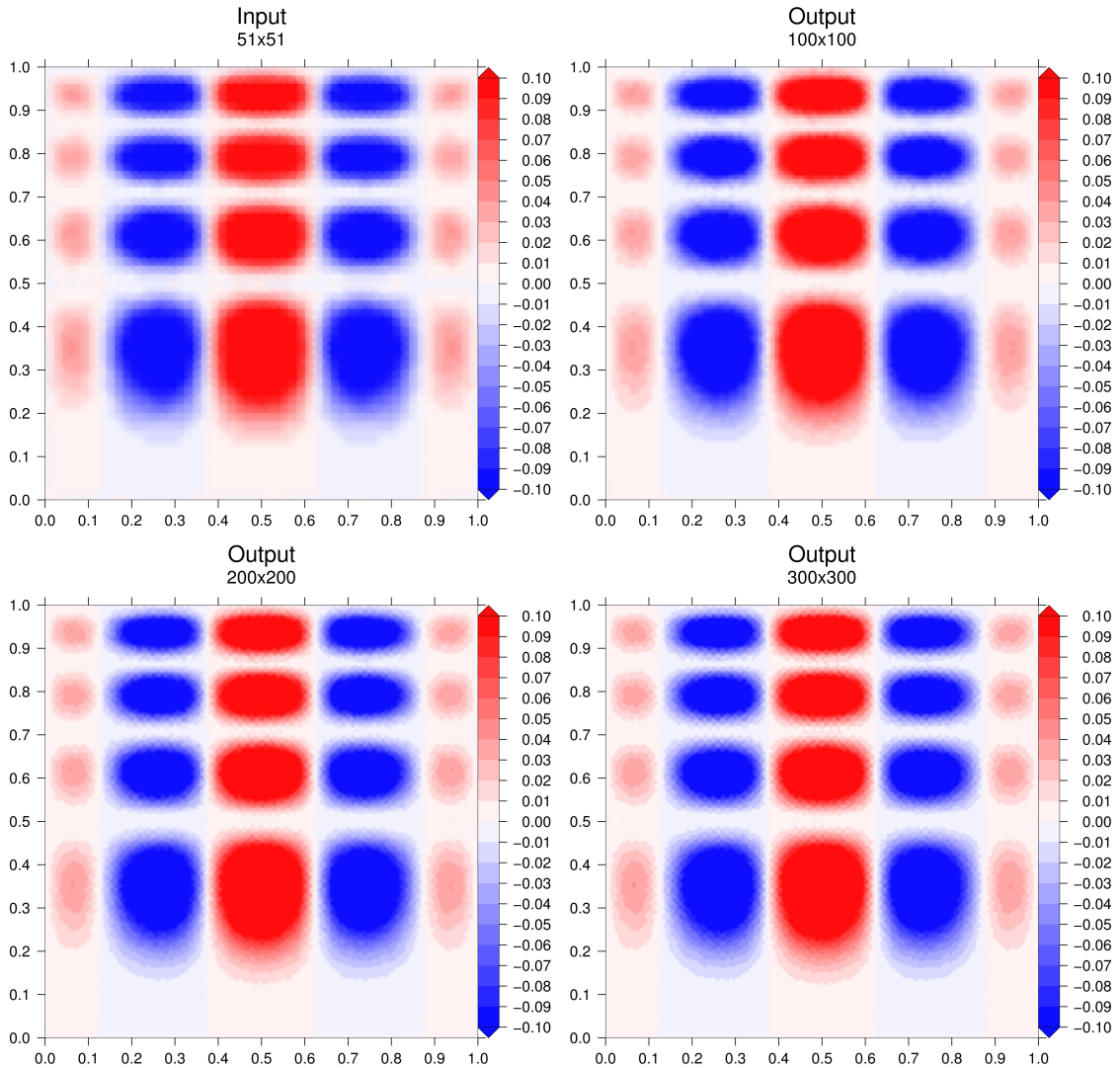


Figure 3. Interpolation results for the three cases. Figures generated with the Generic Mapping Tools (?)

Table 1. Performance for each case with $p = 1$

Case	100×100	200×200	300×300
NMSE (%)	0.324	0.319	0.319
CPU time (s)	0.45	1.84	4.1

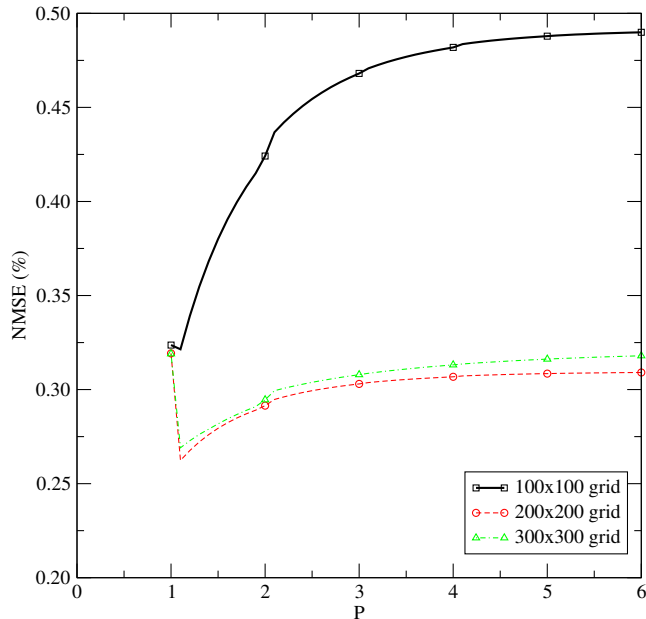


Figure 4. Evolution of performances based on the NMSE for the three cases as a function of the parameter p of the p -distance computation

5 Example in 5D for a regular grid

Still using the *general* program, an example in 5D ($N = 5$) is proposed using the function :

$$Y = f(X) = x_1 \times (1 - x_1) \times \cos(4\pi x_1) \times \sin(4\pi x_2) \times \cos(4\pi x_3) \times \sin(4\pi x_4) \times \cos(4\pi x_5) \quad (12)$$

160 with $x_1, x_2, x_3, x_4, x_5 \in [0, 1]$. The input data grid is a regular grid of $I_i = 35$ interval edges on each dimension $i \in [1, 5]$ then $35^5 = 52521875$ grid points. The goal is to find the results on a coarse grid of 9 elements on each dimension then $9^5 = 59049$ grid points. This case is an opportunity to test the influence of the number of neighbours to calculate the result. In our case, the parameter p of the p -distance is set to $p = 2$. The interpolation seems to provide a better performance on the NMSE for our function with less neighbours (case $N + 1$) and obviously with a lower CPU time. This could certainly ~~depends~~ depend on the
165 type of function to interpolate.

Table 2. Performance for the 5D (N=5) case with $p = 2$

Number of neighbours	2^N	$N + 1$
NMSE (%)	1.570	0.870
CPU time (s)	17.32	6.00

Another test with the 5D case is performed to test the influence of the normalisation as defined in Equation 9 (flag *norm*) by defining an irregular grid with still $35^5 = 52521875$ input data points but with (i) random intervals values, and (ii) one dimension depending on another. The definition of the input grid is defined in Appendix D and provided in the code package. With a similar order of magnitude of consumed CPU time the normalisation *norm = .True.* produce a $NMSE = 0.499\%$ compared to $NMSE = 0.822\%$ for *norm = .False.* There is then an added value of using such a normalisation with comparable CPU time consuming (rising from 2.68 to 3.44 s for our case).

6 Comparison with Python for a regular grid

The code has been tested against the *Python* procedure *scipy.interpolate.griddata* freely available by (?), for the following function:

$$Y = f(X) = x_1 \times (1 - x_1) \times \cos(4\pi x_1) \times \sin(4\pi x_2) \times \cos(4\pi x_3) \quad (13)$$

with $x_1, x_2, x_3 \in [0, 1]$. The input data grid is a regular grid of $I_i = 35$ interval edges on each dimension $i \in [1, 5]$ then $35^3 = 42875$ grid points. The goal is to find the results on a coarse grid of 9 elements on each dimension then $9^3 = 729$ grid points. A case in 3D has been used for this test because the *Python* library was not able to work with very large datasets (*overflow error*) while our program could make ~~it. *scipy.interpolate.griddata* perfectly work.~~ *scipy.interpolate.griddata* is used with the bilinear interpolation option while our method is configured with $p = 2$.

Table 3 clearly shows how the Fortran code is faster compared to the Python library. However, the bilinear interpolation method seems to provide a higher accuracy than the inverse distance ~~one method~~ embedded in our ~~method program~~. Nevertheless, the error produced by our method looks acceptable.

Table 3. Comparison of performances between our ~~method code~~ for a 3D case with the *griddata* Python library. The Machine 2 is used (Appendix F).

	Our code with $N + 1$ neighbours	Our code with 2^N neighbours	Python
NMSE (%)	0.627	1.03	0.326
CPU time (s)	0.04	0.04	19.49

7 Geophysics application

185 7.1 **Methodology**The Backplumes model

~~The back-trajectory module called BACKPLUMES detailed in this section can use output files from CHIMERE (?) or WRF (?) . This kind of model has mainly one calculation to do several times i. e. interpolate the position of a point in a gridded 3D-times domain.~~
In order to test this new interpolation program, it is implemented in a back trajectories model called "Backplumes". This model was already used in some studies such as (?) and (?) for example. Backplumes is open source and is available on the CHIMERE web site. Backplumes calculates back trajectories from a starting point and a starting date. It is ~~why the implementation of a robust and precise interpolation scheme as~~ different from other back trajectories models, such as HYSPLIT (?), STILT (?), (?) and Flexpart (?), because it launches hundreds of particles and plot all trajectories as outputs. Thus, the answer is complementary compared to the other models: the output results is all possible trajectories, and not only the most likely.

195 An advantage of Backplumes for the WRF and CHIMERE users is that the code is dedicated to directly read output results of these models. Being developed by the CHIMERE developers teams, the code is completely homogeneous with CHIMERE in term of numerical libraries. Another advantage is that the code is very fast and calculates hundreds of trajectories in a few minutes. Using the wind fields of WRF or CHIMERE, and running on the ~~one presented in this study is a key point.~~ The first advantage of BACKPLUMES is to use the results of a simulation already performed. The second advantage is to be
200 homogeneous with the "direct" simulation by using the same wind field and the same grid. BACKPLUMES is different than other back-trajectories models, such as Hysplit or Flexpart. Its goal is not to estimate the most likely trajectory as an envelope of numerous possible trajectories . Since it is difficult to calculate correct probabilities back in time, the choice was made to randomly launch numerous trajectories and try to cover all possible origins., the results of back trajectories are fully consistent with the simulations done by the models.

205 Backplumes is dedicated to calculate transport but not chemistry: only passive air particles (or tracers) are released. But a distinction could be made between gaseous or particulate tracers: for the latter one, a settling velocity is calculated to have a more realistic trajectory. The model is easy to use and light because a small set of meteorological parameters is required. These meteorological parameters are described in Table G1 for WRF and CHIMERE. ~~The BACKPLUMES model is an open-source code and is available on the CHIMERE model web site.~~

210 ~~Parameter Model variable name Longitude, latitude XLONG, XLAT Parameters for altitude P_TOP, ZNU, ZNW, P, PB-PH, PHB, PSFC Wind components U, V, W Q₀ HFX \bar{h} PBLH Longitude, latitude lon, lat Altitude hlay Wind winz, winm, winw Q₀ sshf \bar{h} hght List of parameters read by the BACKPLUMES program to calculate trajectories.~~

~~Parameters are the three-dimensional wind components, the boundary layer height \bar{h} , the surface sensible heat flux Q_0 and the altitude of each model layer. The wind components are used for the horizontal and vertical transport. The boundary layer height is used to define the vertical extent of the possible mixing and the surface sensible heat flux is used to know if the current~~
215 ~~modelled hour corresponds to a stable or unstable surface layer (for when the particle is close to the surface).~~

The first step of the calculation is to choose a target location as a starting point. The user has-to-must select a date, longitude, latitude and altitude, obviously included in the modelled domain and during the modelled period. From this starting point, the model will calculate trajectories back in time. The number of trajectories is a-up to the user and may be from one to several hundred of tracers.

~~In general, regional atmospheric models produce hourly outputs. But for some applications, an hourly time-step is not adapted. When the model mesh is very fine, the back-trajectories have no sense if they are calculated on an hourly basis. The air masses may cross several grid cells in 1 hour and a smaller integration time-step is required to preserve the trajectory continuity. This is the problem to respect the Courant Freidrich Levy (CFL) number as in forward transport mode. In this program, the user can specify a sub-hourly step: the meteorological variables are then linearly interpolated between the two consecutive hours using a classical 1D linear interpolation. The altitude is not provided in the WRF output files. It is thus necessary to diagnose it. For all back-trajectories, note that all altitudes are Above Ground Level (AGL). The altitude is computed as follows:-~~

$$p^* = p_{surf} - p_{top}$$

where p_{surf} (PSFC) is the surface pressure and p_{top} is the top pressure of the model domain. If p_{top} is constant over the whole domain, p_{surf} and thus p^* are dependent on the horizontal grid cell.

$$z_0 = \frac{\Phi(1) + \Phi'(1)}{g}$$

where Φ is the geopotential (PHB) and Φ' (PH) its perturbation at vertical level k . g is the acceleration of gravity, $g=9.81 \text{ m s}^{-2}$. For each vertical level k , the layer thickness Δz and the cell top altitude z_k is estimated as:-

$$d_m = \log \left(\frac{p^* \eta_M - p_{top}}{p^* \eta_M + p_{top}} \right)$$

$$d_u = \log \left(\frac{p^* \eta_M - p_{top}}{p^* \eta_F + p_{top}} \right)$$

$$z_1 = \frac{\Phi(k) + \Phi'(k)}{g}$$

$$z_2 = \frac{\Phi(k+1) + \Phi'(k+1)}{g}$$

$$\Delta z = (z_2 - z_1) \frac{d_u}{d_m}$$

$$z(k) = z_1 + \Delta z - z_0$$

where η_M is eta values on full (w) levels (ZNW) and η_F is eta values on half (mass) levels (ZNU). The layer thicknesses varying in space but also in time, this calculation is done for all trajectories and all time-step. At each time-step and for each

trajectory, the position of the air mass is estimated by subtracting its pathway travelled as longitude $\Delta\lambda$ and, latitude $\Delta\phi$ and altitude Δz to the current position in longitude (λ) and latitude (ϕ). To do so, all necessary variables are interpolated in 3D or 2D with our general interpolation program with the interpolation program NterGeo.v2020a described in the previous section.

240 The calculation is described in Appendix G.

$$\begin{aligned}\phi_{rad} &= \phi \frac{\pi}{180} \\ \Delta x &= u \frac{3600}{\Delta t} \\ \Delta y &= v \frac{3600}{\Delta t} \\ \Delta \lambda &= \frac{\Delta x}{R \cos(\phi_{rad})} \frac{180}{\pi} \\ \Delta \phi &= \frac{\Delta y}{R} \frac{180}{\pi}\end{aligned}$$

with the wind speed is provided in m s^{-1} on an hourly basis, R is the Earth radius as $R=6371$ km. The new position for one tracer is thus:-

$$\begin{aligned}\lambda_{t-1} &= \lambda_t - \Delta \lambda \\ \phi_{t-1} &= \phi_t - \Delta \phi\end{aligned}$$

245 The key point of this program is In order to respect the Courant Friedrich Levy (CFL) number, a sub-time step may be calculated. If the input data are hourly provided (as in many regional models), the choice made for the vertical mixing (Figure ??) meteorological variables are interpolated between the two consecutive hours to obtain refined input data.

The goal of backplumes is to estimate all possible back trajectories. Then, starting from one unique point, it is necessary to add a pseudo-turbulence in the calculation of the altitude. Depending on the vertical position of the tracer, several hypotheses
250 are made. Two parameters are checked for each tracer and each time-step: (i) the boundary layer height enables to know if the partiele-tracer is in the boundary layer or above in the free troposphere, (ii) the surface sensible heat fluxes enables to know if the atmosphere is stable or unstable.

Methodology for the vertical distribution as a function of the diurnal cycle of the boundary layer height.

When the tracer is diagnosed ~~to be~~ in the boundary layer, there are two cases: the boundary layer is stable or unstable. If the
255 boundary layer is stable, $Q_0 < 0$, the partiele-tracer stays in the boundary layer at the same altitude. The new vertical position of the tracer is:

$$z_{t-1} = z_t \tag{14}$$

If the boundary layer is unstable, $Q_0 > 0$, the ~~particle-tracer~~ is considered in the convective boundary layer and may be located at every level in this boundary layer the time before. Therefore, a random function is applied to reproduce a potential
260 vertical mixing.

$$z_{t-1} = Rand \times \bar{h} \quad (15)$$

The random function 'Rand' calculates a coefficient, between 0 and 1 to represent a stochastic vertical transport of the tracer.

It is considered that 15 mn is representative of a well-mixed convective layer. ~~?~~ If the time step is larger than 15 mn, the
265 random function is applied. But if the time step is ~~lesser-less~~ than 15 mn, the vertical mixing is reduced to the vicinity of the current position of the tracer. In this case, we have:

$$z_{t-1} = Rand \times \Delta z \times [z_t] \quad (16)$$

where $\Delta z = \frac{1}{2}(z_t^{k-1} + z_t^{k+1})$ and k is the vertical model level corresponding to z_t .

In the free troposphere, the evolution of the ~~particle-tracer~~ is considered to be influenced by the vertical wind component.
270 A random function is applied to estimate its possible vertical motion with values between 0 and $w/2$ m s⁻¹, representative of all possible values of vertical wind speed in the troposphere, ?. The vertical variability of the tracer's position in the free troposphere is calculated by diagnosing the vertical velocity as:

$$z_{t-1} = z_t - (0.5 + Rand)w \frac{3600}{\Delta t} \quad (17)$$

~~where Rand is a random value, between 0 and 1, and different for each tracer and each time.~~

275 7.2 Examples of ~~back-trajectories~~ back trajectories computations

An example is presented for the same case and the WRF and CHIMERE models. The difference between the two models is the number of vertical levels (35 for WRF and 20 for CHIMERE, for the surface to 200 hPa). The online modelling system WRF-CHIMERE is used, meaning that the horizontal grid is the same (a large domain including Europe and Africa and with $\Delta x = \Delta y = 60$ km). The wind field is ~~also the same, WRF sending this information to CHIMERE the same for both models:~~
280 CHIMERE using directly the wind field calculated by WRF. The boundary layer height is different between the two models, WRF using the ~~(?) ?~~ schemes and CHIMERE using the ~~(?) ?~~ scheme. The surface sensible heat flux is the same between the two models, CHIMERE using the flux calculated by WRF. WRF has more vertical model levels than CHIMERE, thus meteorological fields are interpolated from WRF to CHIMERE. It impacts the horizontal and vertical wind fields.

Figure 5 presents the results of ~~back-trajectories~~ back trajectories launched the 10 August 2013 at 12:00 UTC. The location
285 is at longitude +10°E and latitude +25°N, altitude=0 m AGL. This location has no scientific interest but is in the middle of

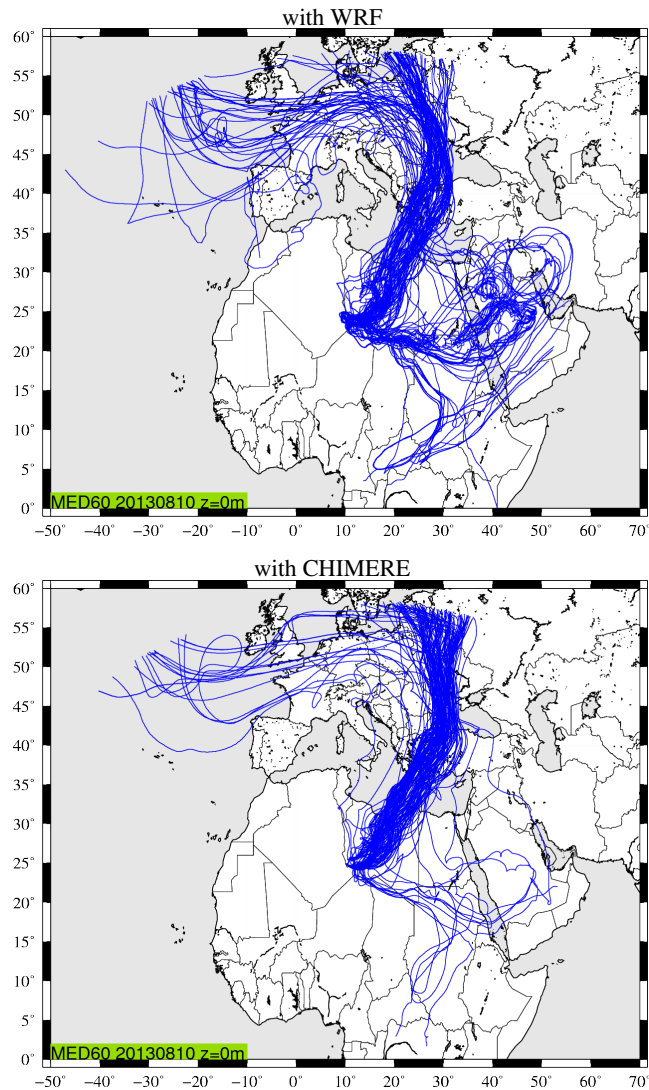


Figure 5. *Backplumes-Back trajectories* calculated using CHIMERE modelled meteorological fields. The starting point is at longitude $+10^{\circ}E$ and latitude $+25^{\circ}N$, altitude=0 m AGL and for the day 10 August 2013 at 12:00 UTC. It corresponds to a case studied during the CHARMEX campaign (?).

the domain, to have the longer *as possible trajectories*trajectories as possible. The complete duration of trajectories represent 10 days back in time. A total of 120 trajectories are launched at the same position and time. They are randomly mixed when they are in the boundary layer to represent the mixing and the diffusion.

The most important part of the plume comes from the North of the starting point. For this main plumes, the calculation is similar between the two models. Another large part of backplumes is modelled at the East of the starting point. However,

290

this fraction is mainly modelled with WRF but not with CHIMERE where only a few trajectories are diagnosed. One possible explanation may be found by analyzing the vertical transport of the trajectories.

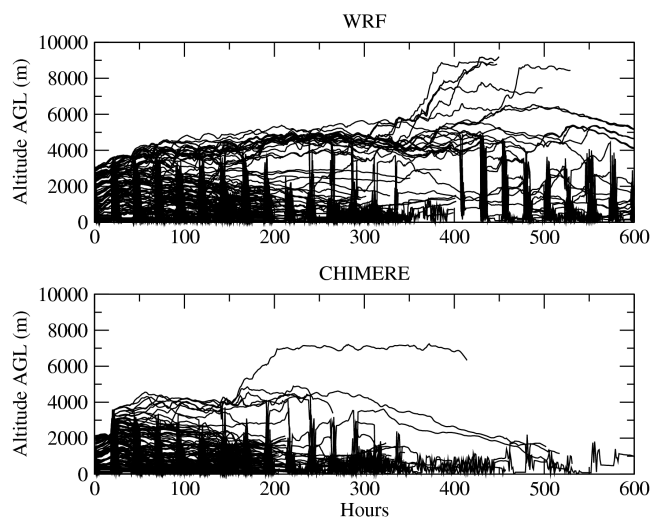


Figure 6. Projection of all ~~back-trajectories~~ back trajectories on a single time-altitude axis.

Figure 6 presents all plumes displayed in the previous figure but projected along the same time-altitude axis. The differences between the two backplumes results are mainly due at-to the calculation of the boundary layer height. When WRF diagnosed an altitude of $\approx 3000\text{m}$, CHIMERE diagnoses $\approx 2000\text{m}$, leading to different direction and wind speed, ~~then-to-~~ Then, this implies a split of the plumes with WRF but not with CHIMERE. This illustrates the sensitivity of the result to the forcing driver model. But, in both cases, the answer in our case is clearly that the main contribution of the air masses located at the starting point are mainly coming from the North-North-East, crossing Tunisia, then the Mediterranean sea and Europe. The main difference between the two calculations is the eastern part of the plume, more intense with WRF than CHIMERE.

300 8 Conclusions

A new interpolation program written in Fortran has been developed to interpolate on N-dimensional matrices. It has been evaluated for several dimension cases up to $N=5$. The code is fast compared to similar Python routines and highly portable in existing geophysical codes. The interpolation program works for any dimension N above 2 and designed to work with irregular but structured grids (characterized by a size for each dimension) or look-up tables. Already used in its "regular"-regular' version in CHIMERE, the "general"-general' program has been tested on a new real application which calculates air mass back-trajectories back trajectories from two widely used atmospheric models: CHIMERE and WRF. This interpolation program can be used for any application in Geophysics and Engineering Sciences but also to explore large structured matrices for Machine Learning applications.

Code availability. The current version of the models are freely available. The exact version of the model used to produce the results used in this paper is archived (i) on Zenodo for NterGeo at <https://doi.org/10.5281/zenodo.3733278> under the GNU General Public License v3.0 or later, as are input data and scripts to run the model and produce the plots for all the simulations presented in this paper. The Backplumes model is an open-source code and is available on the CHIMERE model web site <https://www.lmd.polytechnique.fr/~menut/backplumes.php>.

Appendix A: List of frequently used abbreviations

AGL	Above Ground Level
CFL	Courant Freidrich Levy
CHIMERE	National French CTM
CTM	Chemistry-transport model
CPU	Central Processing Unit
NMSE	Normalized Root Mean Square Error
PBL	Planetary Boundary Layer
PSFC	Surface Pressure
WRF	Weather Research and Forecasting model

315 Appendix B: Binary strategy

This piece of code shows the strategy to optimize the computation of weights for the "regular case". The idea is to minimize the number of operation to benefit from the calculation at each dimension. A non-optimized loop would require $2^N - 1$ multiplications while the optimized loop requires only $2^{N+1} - 4$ multiplications for the weight calculations. Then, for large values of $N \gg 2$, the ratio of required operations between the non-optimized and the optimized loop is $\approx (N - 1)/2$.

```

320 ! ...
      nn=2**ndim           ! ndim is the dimension of the case study
      pn=2**(ndim-1)      !
      ! Loop to convert k in binary
      do k=0,nn-1
325         do j=0,ndim-1
              if (btest(k,j)) then
                  ibin(j,k)=1
              else
                  ibin(j,k)=0
330             endif
          enddo
      enddo
      ...
      ! Main optimized loop (2 loops sequence) to calculate the weighting factors benefiting from
335 ! the previous iteration on the main dimension i

```

```

do i=1,ndim                                ! Loop 1
  ni=2**i
  pi=ni/2
do k=0,ni-1                                ! Loop 2
340  delta=ibin(i-1,k)
      if(i.ne.1) then
        if(delta.eq.0) then
          ws(k)=weight(k)
          weight(k)=weight(k)*w(i,delta) ! where w is the weight on each dimension
345      else
          weight(k)=ws(k-pi)*w(i,delta)
        endif
      else
        weight(k)=w(i,delta)
350    endif
      enddo
    enddo

```

Appendix C: Code design

Note that *avedelta* and *maxdelta* arrays have been externalized to optimize the calculations. In the code package an independent program is available to calculate these arrays to be implemented in your main program. The program is written in Fortran *double precision* ingesting the following arguments:

```

subroutine interpolation_general(&
ndim,      &! Input : Int
maxdim,    &! Input : Int
360 kdim,    &! Input : Array 1D, Int
vect,      &! Input : Array 2D, Real
vtable,    &! Input : Array 1D, Real
table,     &! Input : Array 1D, Real
avedelta,  &! Input : Array 1D, Real
365 maxdelta, &! Input : Array 1D, Real
table,     &! Input : Array 1D, Real
resu,      &! Output : Real
inei,      &! Output : Integer
neighbours,&! Output/Input : Array 2D, Int
370 weights, &! Output : Array 1D, Real
found      &! Output : Logical
)

```

Some hard coded variables can be tested by the user to improve the results. They have been tested and some results are described in this paper. A recompilation is necessary if you change these values.

Table C1. Description of subroutine arguments

Variable	Type	Description	Array dimension
<i>ndim</i>	Integer	Dimension $N > 1$	nd
<i>maxdim</i>	Integer	Total number of elements of the input table $n = \prod_{i=1}^N I_i$ with I_i the number of elements in each dimension i	nd
<i>kdim</i>	Integer 1D Array	Array of number of elements I_i on each dimension i	(0 : N)
<i>vect</i>	Real 2D Array	Array storing in a 1 dimensional array the list of edges on each dimension	(1 : N , 1 : n)
<i>vtable</i>	Real 1D Array	Coordinate values of the point at which to interpolate data	(1 : N)
<i>table</i>	Real 1D Array	Values for the list of known points <i>vect</i> (input grid data)	(1 : n)
<i>avedelta</i>	Real 1D Array	Inverse of average intervals on each dimension N	(N)
<i>maxdelta</i>	Real 1D Array	Maximum intervals on each dimension N	(N)
<i>resu</i>	Real	Result of interpolation for <i>vtable</i>	nd
<i>inei</i>	Integer	Number of neighbours	nd
<i>neighbours</i>	Real 2D Array	Array of neighbours coordinates	(1 : 2^N , 1 : n)
<i>weights</i>	Real 1D Array	Weight for each neighbour	(1 : 2^N)
<i>found</i>	Logical	Returns <i>true</i> or <i>false</i> if respectively the result is found or not found if the point is outside the bounds	nd

```

375 logical , parameter           :: norm=.false.      ! Normalize or not by the average delta on each
    dimension
logical , parameter           :: verbose=.false.    ! Level of message writing (.true. for debug)
logical , parameter           :: neighb=.true.      ! .true. only find up to ndim+1 closest neighbours to
    be faster
380
integer , parameter           :: iconf=1           ! Number of cell to account for before and after
    ! the closest point, iconf=2 can be tested not more
real(kind=iprec) , parameter  :: pnum=2.0d+00     ! p-distance parameter

```


Appendix D: Irregular structured grid example in 5D

385 Herebelow is an example of a 5D array input griddata with irregular intervals with the last dimension (5) depending on dimension (1).

```
! Definition of main dimensions
isize=9           !Outputgrid size
npoints=35       !Inputgrid size
390 ndim=5        !Dimension of the example
allocate(kdim(0:ndim)) !Number of element per dimension array
kdim(0)=1        !Fake dimension for computation issues
kdim(1)=npoints
kdim(2)=npoints
395 kdim(3)=npoints
kdim(4)=npoints
kdim(5)=npoints
! Main array allocation
allocate(delta(ndim))
400 allocate(dstart(ndim))
allocate(vect(ndim,kdim(1),kdim(2),kdim(3),kdim(4),kdim(5)))
allocate(rando(ndim,1:npoints-1)) !Random variable
maxdim=kdim(1)*kdim(2)*kdim(3)*kdim(4)*kdim(5)
!
405 ! Definition of the input grid "vect" with delta, dstart as:
do j=1,ndim
    delta(j)=1.d+00/dfloat(npoints-1)/(dfloat(j)**3)
    dstart(j)=1.d+00/dfloat(npoints-1)/2.d+00/(dfloat(j)**3)
enddo
410 do j=1,ndim
    do i=1,npoints-1
        rando(j,i)=dfloat(int(dlog(rand()*10.d+00+1.d+00))+2) !Random interval on each dimension
    enddo
    rando(j,:)=dfloat(npoints-1)*rando(j,:)/sum(rando(j,:)) !Normalisation of random intervals
415 enddo
vect=0.0d+00 ! Initialisation
! "vect" computation
do i=2,kdim(1)
    vect(1,i,::,::,::)=vect(1,i-1,::,::,::)+rando(1,i-1)*delta(1)
420 enddo
do j=2,kdim(2)
    vect(2,::,j,::,::)=vect(2,::,j-1,::,::)+rando(2,j-1)*delta(2)
enddo
do k=2,kdim(3)
425 vect(3,::,k,::,::)=vect(3,::,k-1,::,::)+rando(3,k-1)*delta(3)
```

```

enddo
do l=2,kdim(4)
  vect(4, :, :, :, l, :) = vect(4, :, :, :, l-1, :) + rando(4, l-1) * delta(4)
enddo
430 do m=2,kdim(5)
  do i=1,kdim(1)
    vect(5, i, :, :, m) = vect(5, i, :, :, m-1) + rando(5, m-1) * delta(5) + &
      &delta(5) * dfloat(i-1) / dfloat(kdim(1)-1) !Dim 5 depends on Dim. 1
  enddo
435 enddo
! ....

```

Appendix E: Characteristics of Machine 1

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- 440 – Byte Order: Little Endian
- CPU(s): 64
- On-line CPU(s) list: 0-63
- Thread(s) per core: 2
- Core(s) per socket: 8
- 445 – Socket(s): 4
- NUMA node(s): 8
- Vendor ID: AuthenticAMD
- CPU family: 21
- Model: 1
- 450 – Model name: AMD Opteron(TM) Processor 6276
- Stepping: 2
- CPU MHz: 2300.000
- CPU max MHz: 2300.0000
- CPU min MHz: 1400.0000
- 455 – BogoMIPS: 4599.83
- Virtualization: AMD-V
- L1d cache: 16K
- L1i cache: 64K
- L2 cache: 2048K
- 460 – L3 cache: 6144K

- Memory block size: 128M
- Total online memory: 128G
- Total offline memory: 0B
- Linux version 3.10.0-1062.12.1.el7.x86_64 (mockbuild@kbuilder.bsys.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-39))

Appendix F: Characteristics of Machine 2

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 96
- On-line CPU(s) list: 0-47
- Off-line CPU(s) list: 48-95
- Thread(s) per core: 1
- Core(s) per socket: 24
- Socket(s): 2
- NUMA node(s): 2
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 85
- Model name: Intel(R) Xeon(R) Platinum 8168 CPU @ 2.70GHz
- Stepping: 4
- CPU MHz: 2701.000
- CPU max MHz: 2701.0000
- CPU min MHz: 1200.0000

- 485 – BogoMIPS: 5400.00
- Virtualization: VT-x
- L1d cache: 32K
- L1i cache: 32K
- L2 cache: 1024K
- 490 – L3 cache: 33792K
- NUMA node0 CPU(s): 0-23
- NUMA node1 CPU(s): 24-47
- Memory block size: 128M
- Total online memory: 190.8G
- 495 – Linux version 3.10.0-957.41.1.el7.x86_64 (mockbuild@x86-vm-26.build.eng.bos.redhat.com) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-36))

Appendix G: The WRF and CHIMERE model parameters used

Parameters are the three-dimensional wind components, the boundary layer height \bar{h} , the surface sensible heat flux Q_0 and the altitude of each model layer. The wind components are used for the horizontal and vertical transport. The boundary layer height is used to define the vertical extent of the possible mixing and the surface sensible heat flux is used to know if the current modelled hour corresponds to a stable or unstable surface layer (for when the tracer is close to the surface).

Backplumes calculates the back trajectories using longitude, latitude and altitude in meters. In case of input data with vertical levels in pressure coordinates, the altitude is calculated from pressure levels. It is the case of the WRF model and the calculation is done as:

The altitude is computed as follows:

$$p^* = p_{surf} - p_{top} \quad (G1)$$

where p_{surf} (PSFC) is the surface pressure and p_{top} is the top pressure of the model domain. If p_{top} is constant over the whole domain, p_{surf} and thus p^* are dependent on the first level grid.

$$z_0 = \frac{\Phi(1) + \Phi'(1)}{g} \quad (G2)$$

<u>Parameter</u>	<u>Model variable name</u>
WRF model	
<u>Longitude, latitude</u>	<u>XLONG, XLAT</u>
<u>Parameters for altitude</u>	<u>P, TOP, ZNU, ZNW, P, PB</u> <u>PH, PHB, PSFC</u>
<u>Wind components</u>	<u>U, V, W</u>
<u>Q₀</u>	<u>HFX</u>
<u>\bar{h}</u>	<u>PBLH</u>
CHIMERE model	
<u>Longitude, latitude</u>	<u>lon, lat</u>
<u>Altitude</u>	<u>hlay</u>
<u>Wind</u>	<u>winz, winm, winw</u>
<u>Q₀</u>	<u>sshf</u>
<u>\bar{h}</u>	<u>hght</u>

Table G1. List of parameters read by the Backplumes program to calculate trajectories.

510 where Φ is the geopotential (PHB) and Φ' (PH) its perturbation at vertical level k . g is the acceleration of gravity, $g=9.81$ m s^{-2} . For each vertical level k , the layer thickness Δz and the cell top altitude z_k is estimated as:

$$\begin{aligned}
 d_m &= \log \left(\frac{p^* \eta_M - p_{top}}{p^* \eta_M + p_{top}} \right) \\
 d_u &= \log \left(\frac{p^* \eta_M - p_{top}}{p^* \eta_F + p_{top}} \right) \\
 z_1 &= \frac{\Phi(k) + \Phi'(k)}{g} \\
 z_2 &= \frac{\Phi(k+1) + \Phi'(k+1)}{g} \\
 \Delta z &= (z_2 - z_1) \frac{d_u}{d_m} \\
 z(k) &= z_1 + \Delta z - z_0
 \end{aligned} \tag{G3}$$

where η_M is its value on full (w) levels (ZNW) and η_F is the eta value on half (mass) levels (ZNU). The layer thicknesses is space and time dependent, this calculation is performed for all trajectories and all time-step.

515 The new position of a tracer back in time is calculated as follow:

$$\begin{aligned}
\phi_{rad} &= \phi \frac{\pi}{180} \\
\Delta x &= u \frac{3600}{\Delta t} \\
\Delta y &= v \frac{3600}{\Delta t} \\
\Delta \lambda &= \frac{\Delta x}{R \cos(\phi_{rad})} \frac{180}{\pi} \\
\Delta \phi &= \frac{\Delta y}{R} \frac{180}{\pi}
\end{aligned}
\tag{G4}$$

with the wind speed is provided in m s^{-1} on an hourly basis, R is the Earth radius as $R=6371$ km. The new position for one tracer is thus:

$$\begin{aligned}
\lambda_{t-1} &= \lambda_t - \Delta \lambda \\
\phi_{t-1} &= \phi_t - \Delta \phi
\end{aligned}
\tag{G5}$$

520 *Author contributions.* Bertrand Bessagnet has developed the code. Laurent Menut and Bertrand Bessagnet has co-developed and implemented the code in the Backplumes.v2020r1 model. Maxime Beauchamp has supported Bertrand Bessagnet for the developments.

Competing interests. The author declares that there is no conflict of interest.

Acknowledgements. This research was funded by the DGA (French Directorate General of Armaments) grant number 2018 60 0074 in the frame of the project NETDESA.

525 **References**

- Donner, L. J., Wyman, B. L., Hemler, R. S., Horowitz, L. W., Ming, Y., Zhao, M., Golaz, J.-C., Ginoux, P., Lin, S.-J., Schwarzkopf, M. D., Austin, J., Alaka, G., Cooke, W. F., Delworth, T. L., Freidenreich, S. M., Gordon, C. T., Griffies, S. M., Held, I. M., Hurlin, W. J., Klein, S. A., Knutson, T. R., Langenhorst, A. R., Lee, H.-C., Lin, Y., Magi, B. I., Malyshev, S. L., Milly, P. C. D., Naik, V., Nath, M. J., Pincus, R., Ploshay, J. J., Ramaswamy, V., Seman, C. J., Shevliakova, E., Sirutis, J. J., Stern, W. F., Stouffer, R. J., Wilson, R. J., Winton, M., Wittenberg, A. T., and Zeng, F.: The Dynamical Core, Physical Parameterizations, and Basic Simulation Characteristics of the Atmospheric Component AM3 of the GFDL Global Coupled Model CM3, *Journal of Climate*, 24, 3484–3519, <https://doi.org/10.1175/2011JCLI3955.1>, <https://doi.org/10.1175/2011JCLI3955.1>, 2011.
- Flamant, C., Deroubaix, A., Chazette, P., Brito, J., Gaetani, M., Knippertz, P., Fink, A. H., de Coetlogon, G., Menut, L., Colomb, A., Denjean, C., Meynadier, R., Rosenberg, P., Dupuy, R., Dominutti, P., Duplissy, J., Bourriane, T., Schwarzenboeck, A., Ramonet, M., and Totems, J.: Aerosol distribution in the northern Gulf of Guinea: local anthropogenic sources, long-range transport, and the role of coastal shallow circulations, *Atmospheric Chemistry and Physics*, 18, 12 363–12 389, <https://doi.org/10.5194/acp-18-12363-2018>, 2018.
- Hardy, R.: Multivariate equations of topography and other irregular surfaces, *Journal of Geophysical Research*, 71, 1905–1915, 1971.
- Hardy, R.: Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988, *Computers and Mathematics with Applications*, 19, 163 – 208, [https://doi.org/https://doi.org/10.1016/0898-1221\(90\)90272-L](https://doi.org/https://doi.org/10.1016/0898-1221(90)90272-L), <http://www.sciencedirect.com/science/article/pii/089812219090272L>, 1990.
- Hofstra, N., Haylock, M., New, M., Jones, P., and Frei, C.: Comparison of six methods for the interpolation of daily, European climate data, *Journal of Geophysical Research: Atmospheres*, 113, <https://doi.org/10.1029/2008JD010100>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD010100>, 2008.
- Hong, S. Y., Noh, Y., and Dudhia, J.: A new vertical diffusion package with an explicit treatment of entrainment processes, *Mon Weather Rev*, 134, 2318–2341, <https://doi.org/10.1175/MWR3199.1>, 2006.
- Kouatchou, J.: NASA Modeling Guru: Basic Comparison of Python, Julia, Matlab, IDL and Java (2018 Edition), available at: <https://modelingguru.nasa.gov/docs/DOC-2676> (last access: 11 March 2020), 2018.
- Lin, J. C., Gerbig, C., Wofsy, S. C., Andrews, A. E., Daube, B. C., Davis, K. J., and Grainger, C. A.: A near-field tool for simulating the upstream influence of atmospheric observations: The Stochastic Time-Inverted Lagrangian Transport (STILT) model, *Journal of Geophysical Research: Atmospheres*, 108, n/a–n/a, <https://doi.org/10.1029/2002JD003161>, <http://dx.doi.org/10.1029/2002JD003161>, 4493, 2003.
- Mailler, S., Menut, L., di Sarra, A. G., Becagli, S., Di Iorio, T., Bessagnet, B., Briant, R., Formenti, P., Doussin, J.-F., Gómez-Amo, J. L., Mallet, M., Rea, G., Siour, G., Sferlazzo, D. M., Traversi, R., Udisti, R., and Turquety, S.: On the radiative impact of aerosols on photolysis rates: comparison of simulations and observations in the Lampedusa island during the ChArMEx/ADRIMED campaign, *Atmospheric Chemistry and Physics*, 16, 1219–1244, <https://doi.org/10.5194/acp-16-1219-2016>, 2016.
- Mailler, S., Menut, L., Khvorostyanov, D., Valari, M., Couvidat, F., Siour, G., Turquety, S., Briant, R., Tuccella, P., Bessagnet, B., Colette, A., Létinois, L., Markakis, K., and Meleux, F.: CHIMERE-2017: from urban to hemispheric chemistry-transport modeling, *Geoscientific Model Development*, 10, 2397–2423, <https://doi.org/10.5194/gmd-10-2397-2017>, 2017.
- Menut, L., Rea, G., Mailler, S., Khvorostyanov, D., and Turquety, S.: Aerosol forecast over the Mediterranean area during July 2013 (ADRIMED/CHARMEX), *Atmospheric Chemistry and Physics*, 15, 7897–7911, <https://doi.org/10.5194/acp-15-7897-2015>, 2015.

- 560 Nehr Korn, T., Eluszkiewicz, J., Wofsy, S., Lin, J., Gerbig, C., Longo, M., and Freitas, S.: Coupled weather research and forecasting-stochastic time-inverted lagrangian transport (WRF-STILT) model, *Meteorology and Atmospheric Physics*, 107, 51–64, <https://doi.org/10.1007/s00703-010-0068-x>, 2010.
- Nenes, A., Pilinis, C., and Pandis, S.: ISORROPIA: A new thermodynamic model for inorganic multicomponent atmospheric aerosols, *Aquatic Geochem.*, 4, 123–152, 1998.
- 565 Nenes, A., Pandis, S. N., and Pilinis, C.: Continued development and testing of a new thermodynamic aerosol module for urban and regional air quality models, *Atmos Environ*, 33, 1553–1560, 1999.
- Pisso, I., Sollum, E., Grythe, H., Kristiansen, N. I., Cassiani, M., Eckhardt, S., Arnold, D., Morton, D., Thompson, R. L., Groot Zwaafink, C. D., Evangelio, N., Sodemann, H., Haimberger, L., Henne, S., Brunner, D., Burkhardt, J. F., Fouilloux, A., Brioude, J., Philipp, A., Seibert, P., and Stohl, A.: The Lagrangian particle dispersion model FLEXPART version 10.4, *Geoscientific Model Development*, 12, 4955–4997, <https://doi.org/10.5194/gmd-12-4955-2019>, <https://www.geosci-model-dev.net/12/4955/2019/>, 2019.
- 570 R.A., P.: *Mesoscale Meteorological Modeling*, Academic Press, 1984.
- Rap, A., Ghosh, S., and Smith, M. H.: Shepard and Hardy Multiquadric Interpolation Methods for Multicomponent Aerosol–Cloud Parameterization, *Journal of the Atmospheric Sciences*, 66, 105–115, <https://doi.org/10.1175/2008JAS2626.1>, <https://doi.org/10.1175/2008JAS2626.1>, 2009.
- 575 Scipy, C.: Interpolate unstructured D-dimensional data: <https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.interpolate.griddata.html>, 2014.
- Shepard, D.: A Two-Dimensional Interpolation Function for Irregularly-Spaced Data, in: *Proceedings of the 1968 23rd ACM National Conference, ACM '68*, p. 517–524, Association for Computing Machinery, New York, NY, USA, <https://doi.org/10.1145/800186.810616>, <https://doi.org/10.1145/800186.810616>, 1968.
- 580 Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Duda, M., Huang, X., Wang, W., and Powers, J. G.: A description of the Advanced Research WRF Version 3, NCAR Tech. Note, pp. 1 – 125, 2008.
- Stein, A. F., Draxler, R. R., Rolph, G. D., Stunder, B. J. B., Cohen, M. D., and Ngan, F.: NOAA’s HYSPLIT Atmospheric Transport and Dispersion Modeling System, *Bulletin of the American Meteorological Society*, 96, 2059–2077, <https://doi.org/10.1175/BAMS-D-14-00110.1>, 2015.
- 585 Stull, R.: *An Introduction to Boundary Layer Meteorology*, Kluwer Academic Publishers Group, 1988.
- Sun, T. and Grimmond, S.: A Python-enhanced urban land surface model SuPy (SUEWS in Python, v2019.2): development, deployment and demonstration, *Geoscientific Model Development*, 12, 2781–2795, <https://doi.org/10.5194/gmd-12-2781-2019>, <https://www.geosci-model-dev.net/12/2781/2019/>, 2019.
- Troen, I. and Mahrt, L.: A simple model of the atmospheric boundary layer: Sensitivity to surface evaporation, *Boundary-Layer Meteorology*, 37, 129–148, 1986.
- 590 Wessel, P., Luis, J. F., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H. F., and Tian, D.: The Generic Mapping Tools Version 6, *Geochemistry, Geophysics, Geosystems*, 20, 5556–5564, <https://doi.org/10.1029/2019GC008515>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GC008515>, 2019.
- Zender, C. S.: Analysis of self-describing gridded geoscience data with netCDF Operators (NCO), *Environmental Modelling & Software*, 23, 1338 – 1342, <https://doi.org/https://doi.org/10.1016/j.envsoft.2008.03.004>, <http://www.sciencedirect.com/science/article/pii/S1364815208000431>, 2008.