Response to reviewer RC1

The following is my reply to RC1, whose comments are reproduced in italics. Anticipated actions for the revision are also highlighted.

The author presents a useful "toy model" data assimilation environment for flux estimation, complementing a long history of such toy models in the dynamical literature. The model description is appropriate and the presented case studies are interesting, and I recommend publication after the following comments are addressed. My comments are from my perspective as a practitioner of emissions flux inversions (i.e., an enthusiastic potential user of EnviFlux), and are primarily about the extent to which the full range of problems arising in my work could be investigated with this tool.

Many thanks to the reviewer for their time and effort in providing valuable feedback. Their insights are very interesting and gratefully received.

What is the computational performance of EnviFlux, in terms that are meaningful for a typical practitioner? For example, what would it take to infer daily fluxes at roughly 2x2.5 degree resolution for a year? How many cores? How much wall time, memory, etc?

The inversions in the paper are for a global grid of longitude $\times$ latitude $\times$ levels $= 65 \times 33 \times 56$, which is about $5.5^\circ \times 5.5^\circ$ in the horizontal. The runs are performed for 100 days of monthly fluxes (although memory is reserved for 16 months). These take around six hours wall time on a single core and use about 64 MB of RAM. Looking at $2.0^\circ \times 2.5^\circ$, this is 180×72 horizontal grid points. Since EnviFlux uses a global spectral representation of the background error covariance matrix, the grid is designed to be consistent with this. This uses a "maximum degree" wavenumber, L, which has an equivalent grid of size $(2L+1) \times (L+1)$. A value $L = 80$ gives roughly the same number of horizontal grid points as 180×72, but the configuration consistent with this is 161×81 or about $2.25^\circ \times 2.25^\circ$. Assuming the same number of vertical levels (56), this grid has about $161 \times 81/(65 \times 33) \sim 6$ times more grid points than the configuration used in the paper. For daily fluxes, this requires $365/16 =\sim 23$ more times in the state vector.

- The RAM required is estimated to be $6 \times 23 \times 64\,\mathrm{MB} \sim 9\,\mathrm{GB}$.
- The wall time is estimated to be $6 \times 365/100 \sim 22\,$hours.

The wall time estimate assumes the same number of variational iterations (20). These are quite high running costs, but the main purpose of EnviFlux is to research new ideas and methods at lower-resolution than the 'real-world' counterparts.

There is no parallelisation implemented in EnviFlux v1.0, but implementing this in the most computationally intensive parts of the system (the forward and adjoint operators) would obviously reduce the wall time. I would implement this with a shared memory parallelisation such as OpenMP.

Action: I will mention the resources used by EnviFlux in the revision.

Clarify the spatial domain(s) supported by EnviFlux (e.g., does it have to be global). Regional simulations with boundary conditions imposed by a global flux inversion are a very important case in, for example, methane, and introduce huge questions in error quantification (e.g., to what extent do we alias boundary condition error onto in-domain emissions estimates); it would be enormously useful to investigate this with a low-cost tool like EnviFlux.

The spatial domains supported by EnviFlux Vn 1.0 are global with number of longitudes $\times$ latitudes $=(2L+1)\times(L+1)$, where L is the spectral maximum degree wavenumber. EnviFlux could be adapted for a local domain by (i) altering the grid data structure, the input/output and inner-product routines; (ii) calling a limited area model (and adjoint) which is driven by boundary conditions; (iii) altering the background error covariance model; and (iv) exploiting the existing descent algorithm to do the minimisation. This is a moderate piece of work, but would be possible.

Action: I will mention this possibility in the revision.

What are the spatial and temporal resolutions of the case studies presented, and what are the range of possible resolutions?

The case studies presented in the paper use $L = 32$, which is a global grid of size $(2L+1) \times (L+1) =$ longitude $\times$ latitude $=65 \times 33$. There are 56 levels. It is possible to set L to any value that the memory and netCDF libraries permit.

Action: I will make this information more prominent in the paper (and correct an error at the top of Table 1, where I have got the dimensions the wrong way round).

How are wind fields specified? Could users prescribe assimilated wind fields?

The driving winds are stored as a set of netCDF files – one set of u, v, w 3D fields per major time step. These are interpolated from ERA-5 reanalyses. In principle, these could be derived from any reasonable source.

Action: I will make this information more prominent in the paper, and include the ERA-5 reference.

The semi-Lagrangian approach makes simulating chemistry tricky, making this tool (at least at global resolutions) most useful for long-lived species. However, for problems like methane and N2O, it is necessary to account for 3D atmospheric sink terms due to tropospheric OH and stratospheric loss processes respectively. I would be very interested in exploring the sensitivity of inversion approaches to characterizing this sort of loss field. Could EnviFlux be adapted to such problems?

Such sinks could be introduced in EnviFlux via prescribed loss rates. It is an interesting question how a semi-Lagrangian method would work with processes like the OH sink. The The semi-Lagrangian method is used in EnviFlux for its stability, but at the cost of lack of strict conservation. One could approximate the methane loss terms in a semi-Lagrangian scheme by treating the advection completely separately from the chemistry (so that the chemical sink of methane happens at the air parcels' destination points and neglecting contributions over the back-trajectories). It would also be possible to add atmospheric loss rates to the state vector so that the sinks (due to OH and methane oxidation, e.g.) could form parts of the state vector, in addition to the surface fluxes. Toy systems like EnviFlux could be useful to do exploratory work in this area, especially to help understand how the observations divide information between the surface fluxes and the atmospheric sinks.

Action: I will discuss this in the paper.

Section 3.2: observational error correlation has proven to be a big problem in my own satellite-based data assimilation. Shared surface characteristics (e.g. mountains) make for systematic errors that are irreducible due to averaging. How hard would it be to specify more complex observational error distributions with off-diagonal terms?

The principle of dealing with correlated $\mathbf{R}$ is relatively straightforward, but the practice is complicated due to determining which observation errors are correlated and by how much, and cost implications for large numbers of observations. If correlated observations can be batched, this would increase the numerical efficiency of accounting for them.

I would adopt the following strategy to calculate the gradient of the cost function. Assuming that observation errors (even at different times) are correlated, the gradient calculation would no longer have the form of Eq. (6) of the manuscript, which assumes uncorrelated observation error across different times:

$$\begin{aligned}\nabla_{\delta\mathbf{v}} J(\delta\mathbf{v}) &= \delta\mathbf{v} - \\ &\mathbf{B}^{\intercal/2} \sum_{k=0}^{N_T} \left(\mathbf{M}_{0\to k}^{\left\{\mathbf{x}^{\mathrm{b}}+\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\rho} \right)^{\intercal} \mathbf{H}_k^{\intercal} \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \mathbf{H}_k \mathbf{M}_{0\to k}^{\left\{\mathbf{x}^{\mathrm{b}}+\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\rho} \left[\boldsymbol{\chi}^{\mathrm{b}}(0) + \left\{\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\chi \right] \right).\end{aligned}$$

Instead, the gradient would have the form:

$$\begin{aligned}\nabla_{\delta\mathbf{v}} J(\delta\mathbf{v}) &= \delta\mathbf{v} - \mathbf{B}^{\intercal/2} \times \\ &\begin{pmatrix} \mathbf{M}_0^{\intercal}\mathbf{H}_0^{\intercal} & \cdots & \mathbf{M}_{N_T}^{\intercal}\mathbf{H}_{N_T}^{\intercal} \end{pmatrix} \underbrace{\begin{pmatrix} \\ \mathbf{R} \\ \\ \end{pmatrix}^{-1} \underbrace{\begin{pmatrix} \mathbf{y}_0 - \mathbf{H}_0\mathbf{M}_0 \left[\boldsymbol{\chi}^{\mathrm{b}}(0) + \left\{\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\chi \right] \\ \vdots \\ \mathbf{y}_{N_T} - \mathbf{H}_{N_T}\mathbf{M}_{N_T} \left[\boldsymbol{\chi}^{\mathrm{b}}(0) + \left\{\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\chi \right] \end{pmatrix}}_{\mathbf{p}}}_{\mathbf{q}},\end{aligned}$$

where $\mathbf{M}_k$ is shorthand here for $\mathbf{M}_{0\to k}^{\left\{\mathbf{x}^{\mathrm{b}}+\mathbf{B}^{1/2}\delta\mathbf{v}\right\}_\rho}$ in the first equation, $\mathbf{R}$ is the observation error covariance for all observations, and the other notation is explained in the paper. The key aspect of the last equation is that all observations are represented together, allowing observation errors to couple different elements in observation space. Either $\mathbf{R}$ can be written in terms of its eigenvalue decomposition, so $\mathbf{R}^{-1} = \mathbf{F}\boldsymbol{\Lambda}^{-1}\mathbf{F}^{\intercal}$ to evaluate the above, or the equation $\mathbf{p} = \mathbf{R}\mathbf{q}$ can be solved for $\mathbf{q}$, where $\mathbf{p}$ and $\mathbf{q}$ are the observation space vector as indicated above. The eigenvalue decomposition contains information on the condition number of $\mathbf{R}$, which may need to be monitored.

Actions: Correct missing $\mathbf{H}_k^{\intercal}$ in Eq. (6) (it is corrected in the first equation above), and discuss the correlated $\mathbf{R}$ problem in manuscript.

Section 3.3: How easy would it be to trade out this background matrix for other approaches that might better approximate the varied operational systems used by potential EnviFlux users?

All what the EnviFlux algorithm needs is a square-root representation of the background error covariance matrix, which could be something very different to that implemented. Examples are where the correlation lengthscales depend on land or sea, or where the land and sea points are uncorrelated. If the resolution is low, then a horizontal correlation matrix could be explicitly constructed and then 'square-rooted' with an eigenvalue or Cholesky decomposition. If one moves away from the spectral scheme that is implemented in v1.0, then there would no longer be any need to have a grid with $(2L+1)\times(L+1)$ dimensions (see above).

Action: Discuss alternative **B**-matrices in the paper.

Lines 411-414: This is a very interesting discrepancy. In my very rough experience, satellite data is more "reliable" because of the representation error of surface sites, and because satellite data is generally less sensitive dilution in a mischaracterized PBL (the convective transport explanation of Basu et al. is also plausible). Or at least, this is the way I have thought about it without rigorously investigating the issue. To what extent could these sorts of errors in convection/PBLH be studied with EnviFlux?

In the work shown in the manuscript there is no unaccounted transport (in experiments when the wind factor $\eta = 1$) and there is no representativity error. This is likely to be the reason for the discrepancy with Basu et al. (2018). These issues could be studied in EnviFlux by running a much higher resolution model to make the synthetic observations (the nature run), but maintain the lower resolution model in the assimilation. The nature run could also include the effects of sub-grid transport.

Actions: Include further discussion of the discrepancy between this study and Basu et al. (2018), and discuss representativity error.

R.N. Bannister, May 2026