Supplement of Atmos. Chem. Phys., 25, 12737–12751, 2025
https://doi.org/10.5194/acp-25-12737-2025-supplement

Atmospheric Chemistry and Physics
Open Access
EGU

Supplement of

Efficient use of a Lagrangian particle dispersion model for atmospheric inversions using satellite observations of column mixing ratios

Rona L. Thompson et al.

Correspondence to: Rona L. Thompson (rlt@nilu.no)

Description of the new and modified routines in FLEXPART v10.4

The following routines were added to, or modified in, FLEXINVERT v10.4 for the calculation of total column SRRs:

readreleases_satellite.f90	Reads netcdf files containing the latitude and longitude coordinates, pressure levels, averaging kernels and pressure weighting information for the retrievals and sets up the coordinate framework for the particle releases
releaseparticles_satellite.f90	Calculates the particle release positions for the FLEXPART coordinate system
concoutput_inversion.f90	Small modification to name the output files according to the number of retrieval for satellite SRRs
concoutput_inversion_nest.f90	Small modification to name the output files according to the number of retrieval for satellite SRRs
initial_cond_output_inversion.f90	Small modification to name the output files according to the number of retrieval for satellite SRRs
timemanager.f90	Small modification to control if particle releases should be made for satellites or not
readcommand.f90	Included reading of a logical variable (introduced into the COMMAND file) indicating whether runs should be made for satellites or not

Usage with FLEXPART-v11

In FLEXPART-v11, the particle initial positions can be specified by reading in a NetCDF file (Bakels et al. 2024) (this was not possible with previous FLEXPART versions). This means that almost no changes to the FLEXPART-v11 code itself are needed to calculate the SRRs for satellite observations if the correct particle initialization as given in the NetCDF file. The only code change needed is to name the output files by retrieval number and not by timestamp as is the default. The developments described in this paper can be used to create such a NetCDF file. This involves i) reading the satellite observations (L2 data), ii) performing any averaging of the retrievals to super-observations (optional step), and then iii) adapting the routine written for FLEXPART-v10.4, “releaseparticles_satellite.f90” (which calculates the particle locations) to save the locations to a NetCDF file.

Description of the affine algorithm for releasing particles for the satellite pixel geometry

In FLEXPART, a release is made either for a point or a volume. If it is a volume release, then the particles are distributed randomly throughout the volume by generating a random number for each particle and multiplying this by the distance between the latitudinal boundaries of the box to get the latitudinal coordinate, and by the distance between the longitudinal boundaries to get the longitudinal coordinate. However, this obviously means that only rectangular volumes that are aligned with the meridians and parallels can be represented. Therefore, to represent different pixel geometries an affine transformation was implemented as follows.

Affine algorithm steps:

1. Calculate the angle of rotation of the satellite pixel relative to the meridians
2. Calculate the equivalent latitude and longitude bounds of the unrotated pixel
3. Calculate the angle of distortion of the satellite pixel
4. Calculate the equivalent latitude and longitude bounds of the undistorted and unrotated pixel
5. Distribute the particles randomly within this rectangular volume
6. For each particle, reapply the distortion and rotation so that the particles all fall within the original satellite pixel.

This is performed in the subroutine releaseparticles_satellite.f90 which is available from the open gitlab repository: https://git.nilu.no/flexpart/flexpart.git. The sections of code that perform the affine transformation for the horizontal locations of the particles for a given particle release is given below.

For a given retrieval the 4 longitudinal coordinates are given by `xpoints` and the 4 latitudinal coordinates are given by `ypoints`.

In loop over retrievals with index `i`:

```
! calculate rotation and distortion of pixels
!*********************************************

! calculate rotation of pixels
delx=xpoints(i,1)-xpoints(i,2)
dely=ypoints(i,2)-ypoints(i,1)
alpha=atan(dely/delx)              ! angle of rotation

! undo rotation
! point of rotation is pixel centre so adjust x,y so centre is at origin
xpoints0(:)=xpoints(i,:)-sum(xpoints(i,:))/4.
ypoints0(:)=ypoints(i,:)-sum(ypoints(i,:))/4.
xp1=xpoints0(1)*cos(-1.*alpha)+ypoints0(1)*sin(-1.*alpha)
yp1=-1.*xpoints0(1)*sin(-1.*alpha)+ypoints0(1)*cos(-1.*alpha)
xp4=xpoints0(4)*cos(-1.*alpha)+ypoints0(4)*sin(-1.*alpha)
xp2=xpoints0(2)*cos(-1.*alpha)+ypoints0(2)*sin(-1.*alpha)
yp4=-1.*xpoints0(4)*sin(-1.*alpha)+ypoints0(4)*cos(-1.*alpha)

! calculate angle of distortion
! from unrotated pixel
theta=atan((xp1-xp4)/(yp4-yp1))

! undo distortion
xp0=xp1-((yp4-yp1)/2.)*tan(theta)
yp0=yp1

! pixel width and height
yaux=yp4-yp1
xaux=xp2-xp1
```

In an inner loop over layers and particles (note `xtra1` and `ytra1` are the particle release positions and `ipart` is the index to the particle):

```
! Particle coordinates are determined by using a random position within the release volume

            ! Determine horizontal particle position
            !****************************************

            ! for satellites need to account for rotation of pixels
            ! transform all xtra1 and ytra1 positions accordingly

            ! initial particle positions
            ! these are the unrotated and undistorted positions with centre at origin
            ! ran1 is random number generator in range (0,1)
            xpini=xp0+ran1(idummy)*xaux
            ypini=yp0+ran1(idummy)*yaux

            ! reapply distortion x direction
            xpini=xpini-ypini*tan(theta)
            ypini=ypini
            ! rotate particle positions and shift so pixel centre is in former position
            xtra1(ipart)=xpini*cos(alpha)+ypini*sin(alpha)+sum(xpoints(i,:))/4.
            ytra1(ipart)=-1.*xpini*sin(alpha)+ypini*cos(alpha)+sum(ypoints(i,:))/4.

            ! limit particle positions to domain
            if (xglobal) then
              if (xtra1(ipart).gt.real(nxmin1)) xtra1(ipart)= &
                   xtra1(ipart)-real(nxmin1)
              if (xtra1(ipart).lt.0.) xtra1(ipart)= &
                   xtra1(ipart)+real(nxmin1)
            endif
```

Supplementary Figures

Figure S1: Maps of the variable grid used for the optimization of the fluxes. a) for the TROPOMI inversion, b) for the ground-based observation inversion. The blue grid cells represent those over ocean and are not optimized in the inversion.

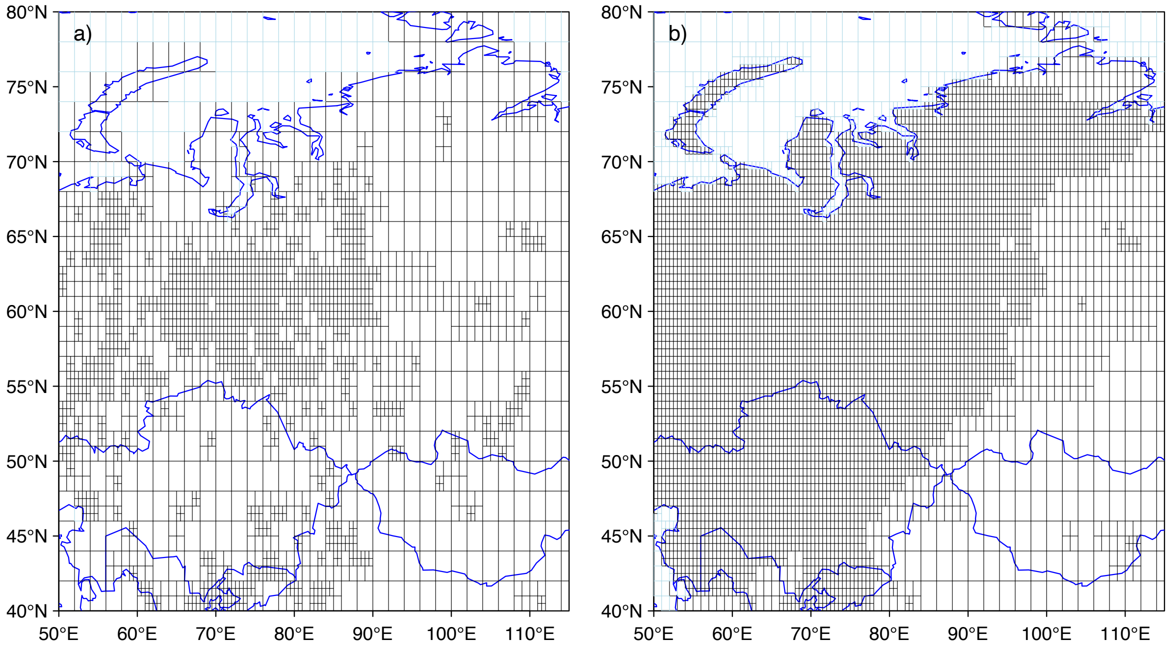

Figure S2: Overview of the number of observations per state vector time step (14 days). a) satellite super-observations in 5° latitude bands for the domain from 40° to 80°N, b) ground-based observations.

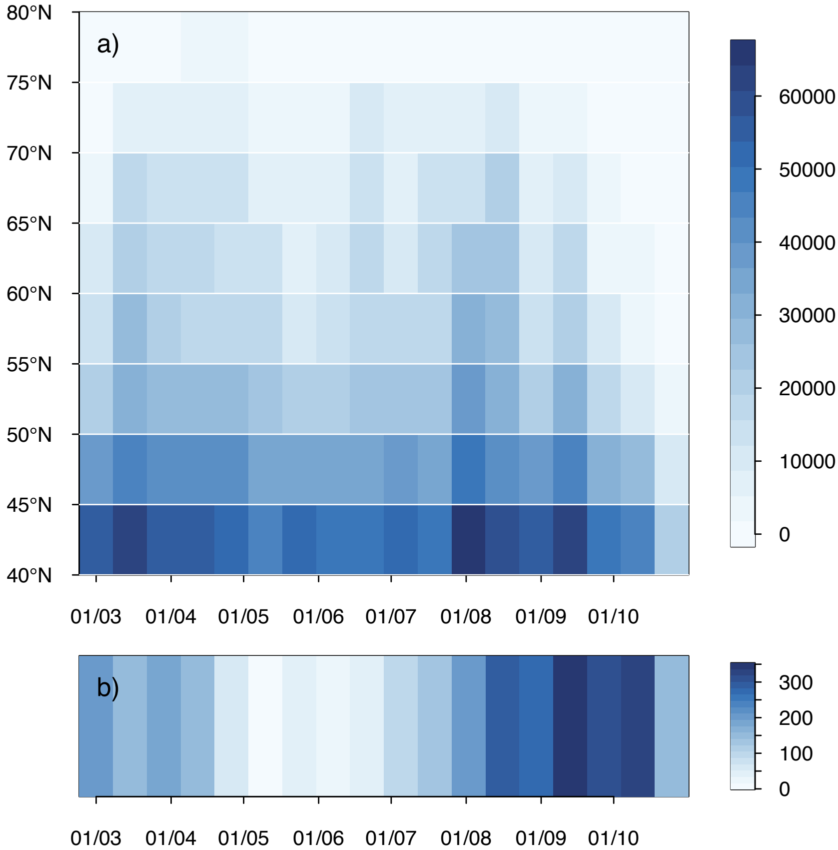

Figure S3: The cost at each iteration for a) the inversion using TROPOMI observations, and b) the inversion using ground-based observations.

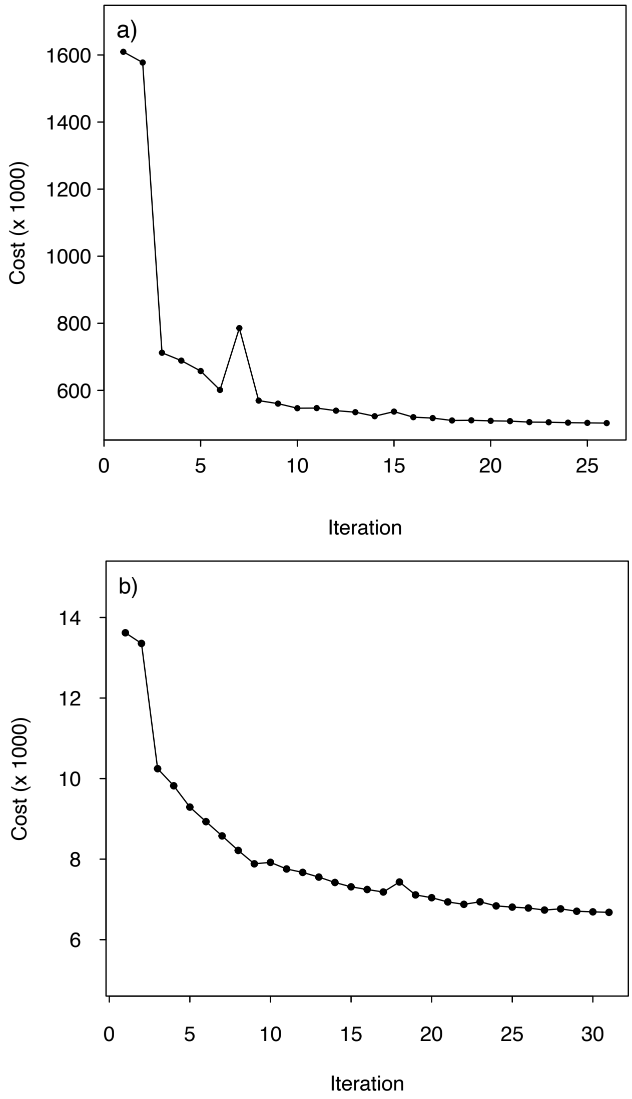

Figure S4: Difference between the posterior and prior modelled XCH_4 (units ppb) using EGG4 initial mixing ratios, a) March and b) July.

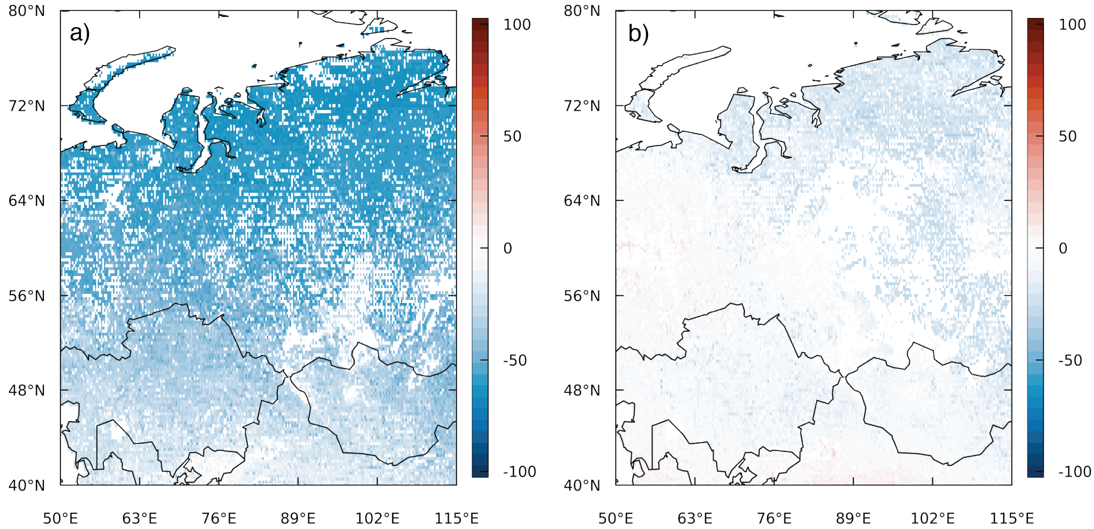

Figure S5: Comparison of the prior modelled XCH_4 (ppb) using different initial mixing ratio fields for the boundary conditions. a) March 2020 using CAMSv20r1, b) March 2020 using EGG4, c) July 2020 using CAMSv20r1, and d) July 2020 using EGG4.

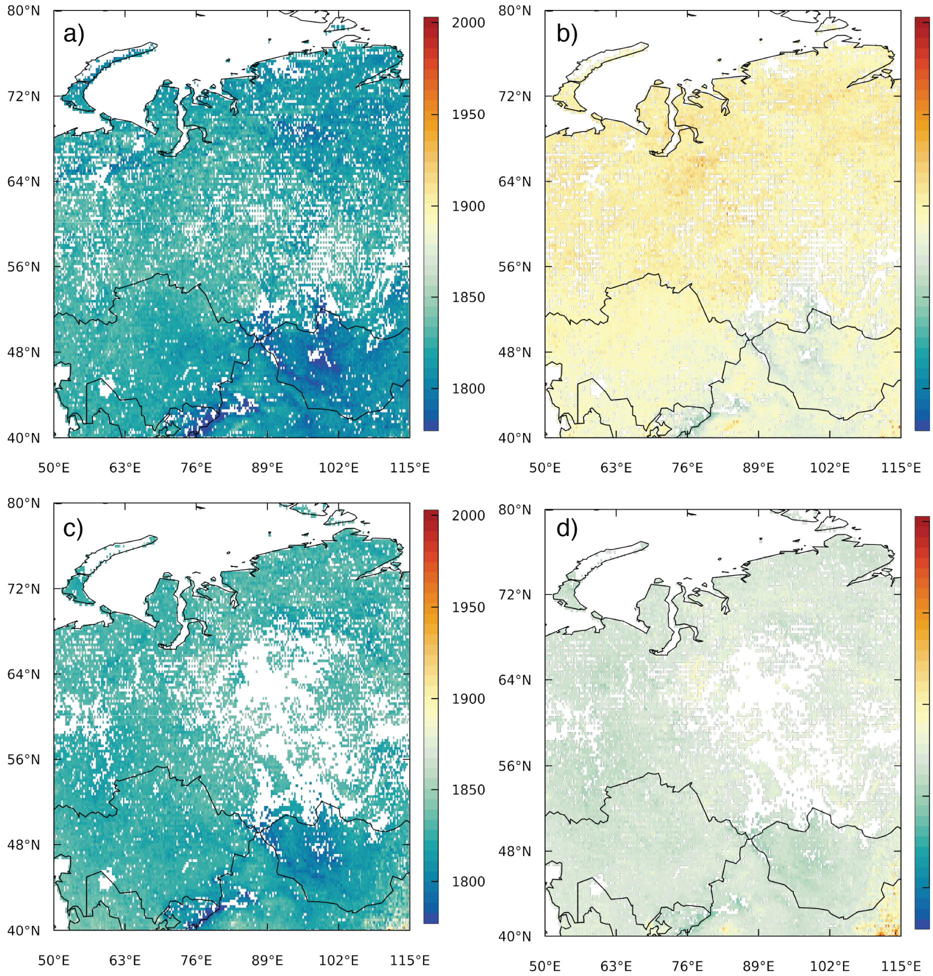

Figure S6: Vertical profile of CH_4 mixing ratio (zonal mean) in units of ppb. a) CAMSv20r1 for March 2020, b) EGG4 for March 2020, c) CAMSv20r1 for July 2020 and d) EGG4 for July 2020

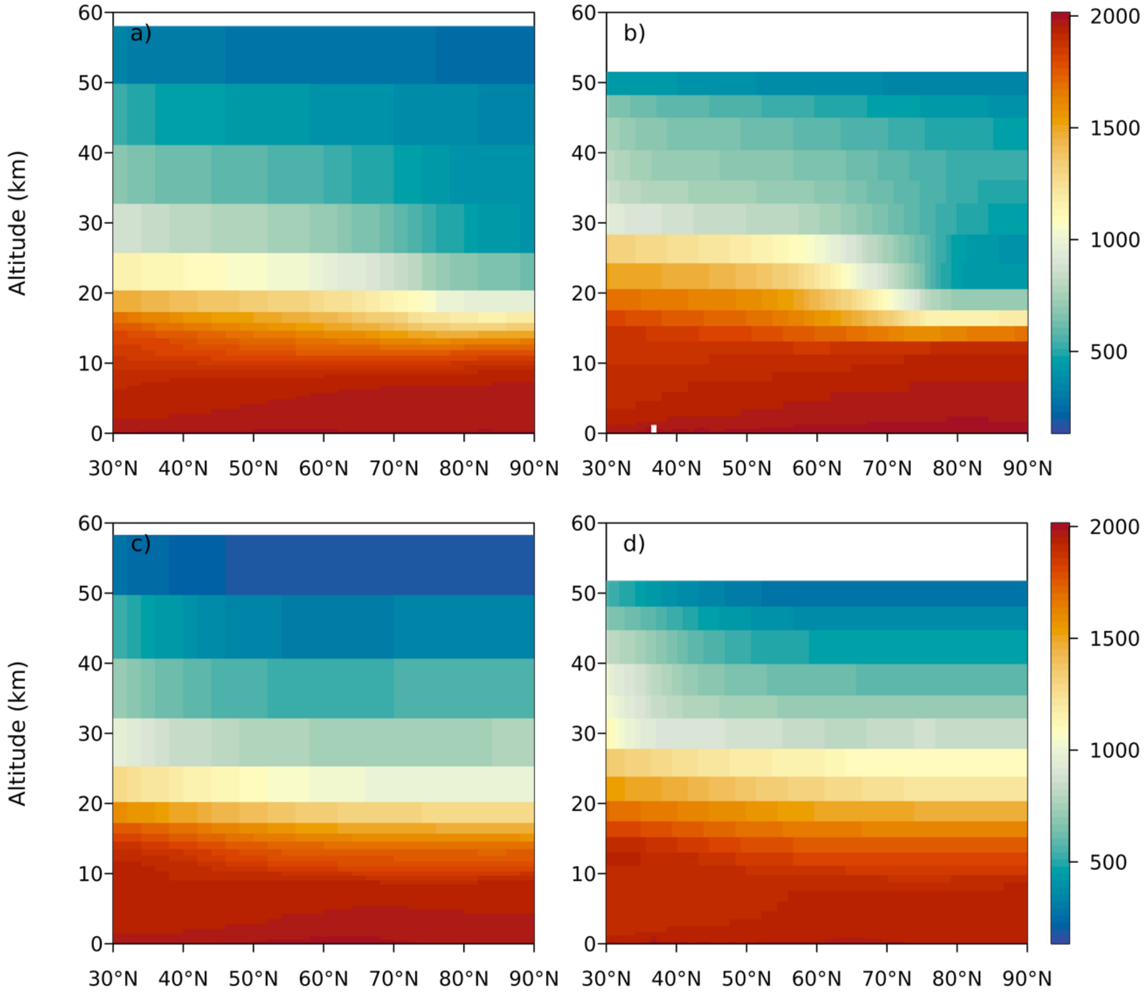

Figure S7: Comparison of XCH_4 derived directly from the initial mixing ratio fields with observations. a) observed for March 2020, b) CAMSv20r1 for March 2020, c) EGG4 for March 2020, d) observed for July 2020, e) CAMSv20r1 for July 2020, f) EGG4 for July 2020.

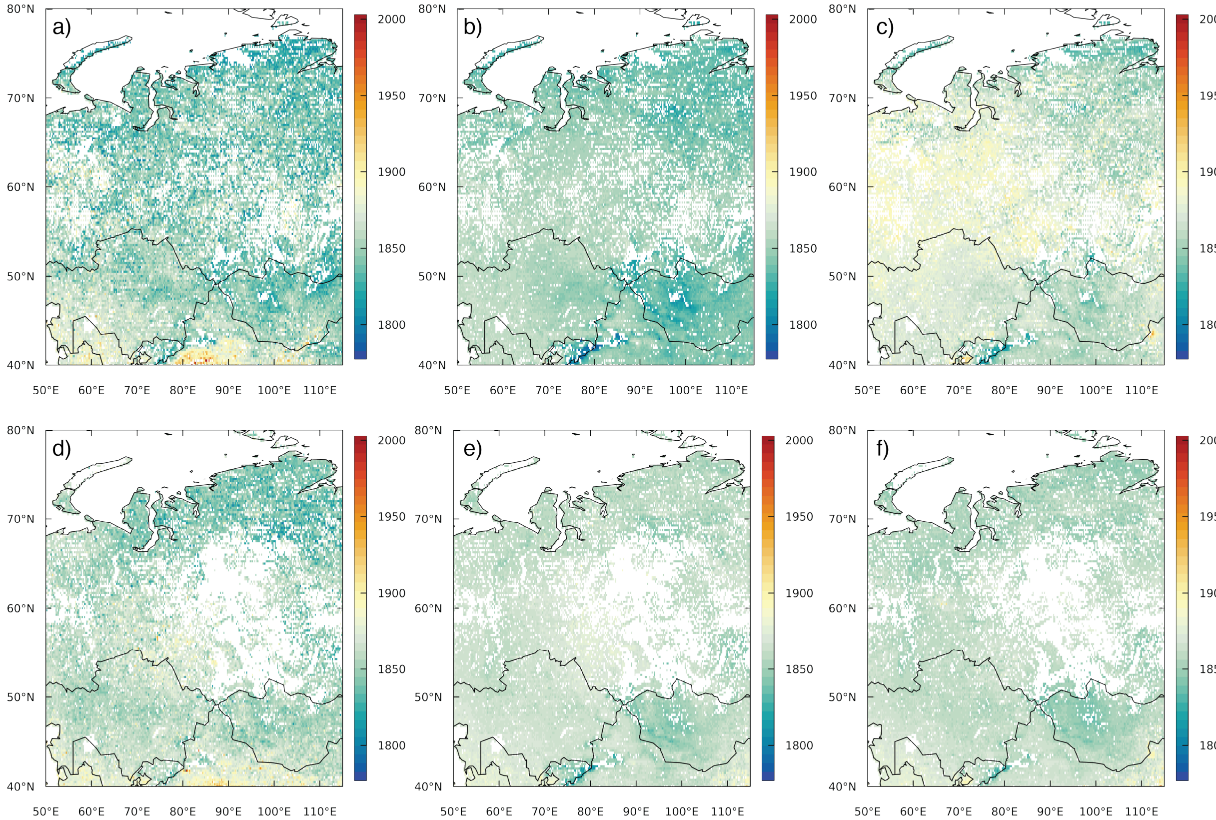

Figure S8: Posterior scalars of the initial mixing ratios from the TROPOMI inversion using 3D initial mixing ratio fields from CAMSv20r1. In each sub-panel, the scalars are shown for each timestep of 28 days and for each of the four latitude bands.

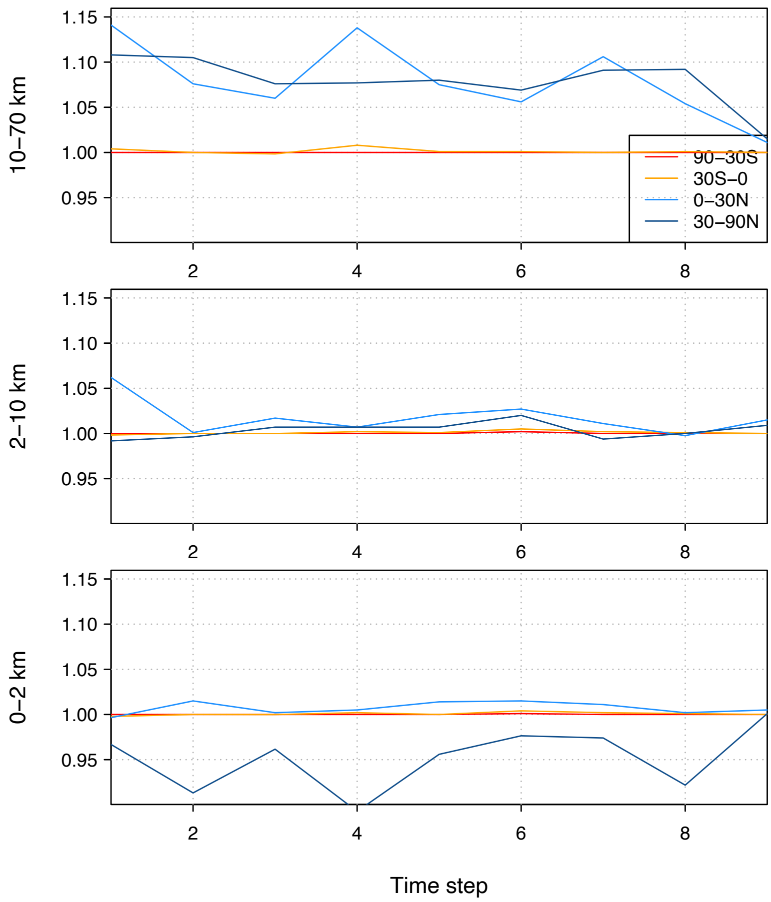

Figure S9: Mean posterior fluxes (units of g/m^2/day) for March to October from a) the inversion using CAMSv20r1 for the boundary conditions, and b) the inversion using EGG4 for the boundary conditions.

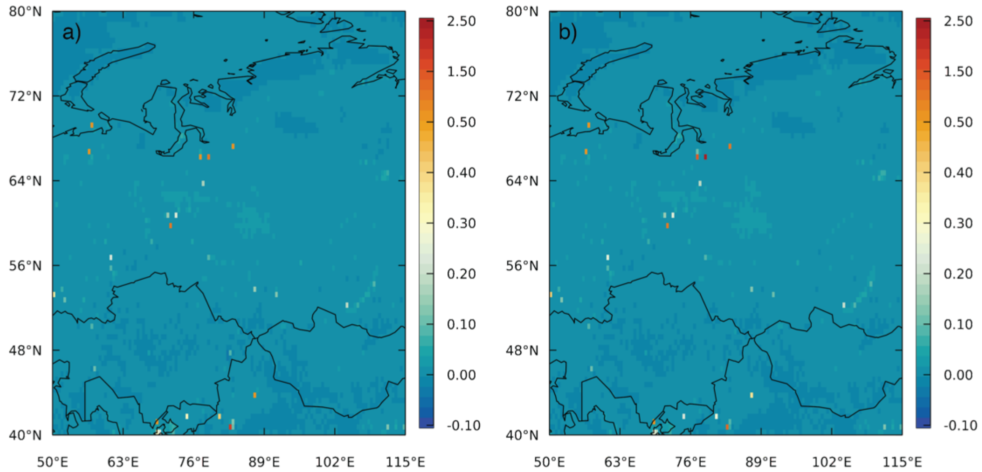

Figure S10: a) Mean prior flux uncertainties (units of g/m^2/day) for March to October and b) mean source-receptor-relationship (units of s kg/m^3)) for March to October.