Guide and theory for source/sink data assimilation code

Ross Bannister, Data Assimilation Research Centre, University of Reading

January 2008/November 2009

This program needs the associated Legendre polynomials file *AssocLegendrePoly*.

Input state mode (to generate the main input data files for tracer, source and winds)

Source.out	
I	
Number of wind fields	*E.g. 200*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Advection scheme	1: Centred difference 2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Extract which level number?	*E.g. 11 (~500 hPa)* 0 do not do convert wind resolution step
Type of set-up for truth field	1: tracer 1000 $\pm$ 2 to be advected, 1 source, 1 sink 2: as opt. 1 but with 1 source field 3: data from *UnadvectTruth.nc* tracer $\pm$ 2 to be advected
Wind inflation factor	*E.g. 1.0*

<u>Algorithm</u>

- If level > 0: Convert first "Number of wind fields" wind files specified in the file *FullResFileList* (files are in directory *FullResWinds*) to low resolution and at one level. Each low resolution wind is output in *RedResWinds/RedResWindxxxx*. This name is appended to the file *RedResFileList*.
  - Each high resolution wind field (u,v) should be on a C-grid and stored in a netCDF file. The number of longitude and latitude points are specified by *nlongs_in* and *nlats_in*. One level only, "Extract which level number", is extracted per field. The number of longitudes and latitudes is *nlongs*, *nlats*.
- Set-up the truth field, *Truth.nc*.
  - If "Truth set-up type" is 1 then the source is set to one source and one sink region in the domain, the tracer is set to 1000 $\pm$ 2, and is advected by the winds to give the *Truth.nc*.
  - If "Truth set-up type" is 2 then the source and initial tracer is read from the file *UnadvectTruth.nc*, the initial tracer is $\pm$ 2 and is advected by the winds to give the *Truth.nc*.

Standard deviation mode (to generate a file of background standard deviations)

Source.out	
S	
Option for creation of standard deviation	1: constant 2: lat dependent 3: land-sea mask dependent
<u>If choose opt. 1</u>	
Tracer standard deviation	*E.g. 15.0*
Source standard deviation	*E.g. 1.5E-5*
<u>If choose opt. 2</u>	
Tracer standard deviation	*E.g. 15.0*
High source standard deviation	*E.g. 4.0E-6*
Low source standard deviation	*E.g. 2.0E-6*
Lat of peak source	*E.g. 40*
Width of peak source	*E.g. 17*
<u>If choose opt. 3</u>	
Tracer standard deviation	*E.g. 15.0*
Land source standard deviation	*E.g. 2.5E-5*
Sea source standard deviation	*E.g. 2.0E-6*

Algorithm

- Make file BgStd.nc with the background error standard deviations as specified above.
- Note that option 3 needs to input the file *SourceSinkMask*

Background state mode (to generate a valid background state)

Source.out	
B	
Tracer background error peak spectral variance	*E.g. 3.8828*
Source background error peak spectral variance	*E.g. 5.88*
Tracer background error lengthscale	*E.g. 0.03*
Source background error lengthscale	*E.g. 0.01*

Algorithm

- Set-up the background spectra of the tracer and source using the above information.
- Read-in the tracer/source standard deviation fields, *BgStd.nc*.
- Read-in the true tracer and source field, *Truth.nc*.
- Add random noise (consistent with the background error covariances) to the truth.
- Output the background tracer and source field, *Background.nc*.

Observation mode (to generate synthetic observations)

Source.out	
O	
Number of wind fields	*E.g. 200*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Source flag	0 off 1 on
Advection scheme	1: Centred difference 2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Wind inflation factor	*E.g. 1.0*
Preset or arbitrary observation times/locations	0: arbitrary (times/locations) read from file 1: preset positions (regular spaced) 2: preset positions (satellite tracks)
<u>If choose arbitrary observations (0)</u>:	
Use rdm noise for pseudo obs or fixed innov	1: random obs with variance spec in file 2: fixed innov with sqrt of variance spec in file
<u>If choose preset observations (1) or (2)</u>:	
Observation error variance	*E.g. 25.0*
Number of time steps between observations	*E.g. 1*
<u>If choose preset regular spaced observations (1)</u>:	
Spacing between observations in grid points	
<u>If choose preset satellite observations (2)</u>:	
Period of orbit (seconds)	*E.g. 7200.0 (2 hours)*
Longitude of node (degrees)	*E.g. 30.0*
Inclination (degrees)	*E.g. 65.0*
Greenwich offset at t=0 (degrees)	*E.g. 0.0*

<u>Algorithm</u>

- If choose regular or satellite observations
  - Place observations locations in structure (timestep, longitude, latitude, variance).
- If choose arbitrary observations
  - Read-in observations from file *ArbitraryObs* with the structure given below.
- Generate synthetic observations by running the model from the truth (in file *Truth.nc*) and applying the forward mode and adding Gaussian noise (the model run requires the file *ResResFileList* - see above).
- Model observations are output to the file *Observations* with the structure given below.

<u>Structure of file *ArbitraryObs* (second line onwards repeated for each observation)</u>
Number of observations in this file
Blank line
Timestep of this observation
Longitude of this observation
Latitude of this observation
Variance of this observation or square of fixed innovation (depending upon flag above)

<u>Structure of file *Observations* (second line onwards repeated for each observation)</u>
Number of observations in this file
Separator line
Timestep of this observation

Longitude of this observation
Latitude of this observation
Variance of this observation
Value of this observation

Variational mode (to do a weak constraint variational data assimilation run)

Source.out	
V	
Number of wind fields	*E.g. 200*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Source flag	0 off 1 on
Advection scheme	1: Centred difference 2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Tracer error variance	*E.g. 3.8828*
Source error variance	*E.g. 5.88*
Tracer error lengthscale	*E.g. 0.03*
Source error lengthscale	*E.g. 0.01*
Wind inflation factor	*E.g. 1.0*
Size of gradient to satisfy convergence	*E.g. 1.0e-10*
Max number of iterations	*E.g. 50*
Estimate analysis error	0 off 1 on

Algorithm

- Read in background state, file *Background.nc*.
- Read in observations, file *Observations*.
- Set-up the background spectra of the tracer and source using the above information.
- Read-in the tracer/source standard deviation fields, *BgStd.nc*.
- Output verbose description of operations, file *VarStatus*.
- For each iteration, output observation information in file *VarStatus*
  - timestep, obs value, model obs value, variance
- For each iteration, output statistical information in file *VarStats*
  - iteration No., J, JB_tracer, JB_source, JO, sqrt(norm(residual))
- Output analysis increments, *AnalInc.nc*, the analysis, *Anal.nc*, and (if requested) *Anal_err.nc*.

Test mode (to perform test routines)

Source.out	
T	
Test run type	1 random number test 2 Legendre polynomial orthog. test 3 inverse transform tests 4 adjoint tests 5 implied covariance test (t=0)

	6 implied covariance test (t>=0)
	7 gradient test
<u>If choose random number test (1)</u>	
Random number variance	
Random number mean	
<u>If choose Legendre polynomial orthog. test (2)</u>	
Maximum number of wavenumbers in test	
<u>If choose inverse transform test (3)</u>	
Maximum number of wavenumbers in test	
Source flag	0 off
	1 on
<u>If choose adjoint tests (4)</u>	
Tracer error variance	*E.g. 3.8828*
Source error variance	*E.g. 5.88*
Tracer lengthscale	*E.g. 0.03*
Source lengthscale	*E.g. 0.01*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Number of wind fields	*E.g. 200*
Source flag	0 off
	1 on
Advection scheme	1: Centred difference
	2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Wind inflation factor	*E.g. 1.0*
<u>If choose t=0 implied covariance test (5)</u>	
Tracer error variance	*E.g. 3.8828*
Source error variance	*E.g. 5.88*
Tracer lengthscale	*E.g. 0.03*
Source lengthscale	*E.g. 0.01*
x grid position of delta-function (index)	
y grid position of delta-function (index)	
<u>If choose t>=0 implied cov. test (randomizer method) (6)</u>	NOT YET CODED
Tracer error variance	*E.g. 3.8828*
Source error variance	*E.g. 5.88*
Tracer lengthscale	*E.g. 0.03*
Source lengthscale	*E.g. 0.01*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Number of wind fields	*E.g. 200*
Source flag	0 off
	1 on
Advection scheme	1: Centred difference
	2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
x grid position of delta-function (index)	
y grid position of delta-function (index)	
Wind inflation factor	*E.g. 1.0*
<u>If choose gradient test (7)</u>	
Tracer error variance	*E.g. 3.8828*
Source error variance	*E.g. 5.88*

Tracer lengthscale	*E.g. 0.03*
Source lengthscale	*E.g. 0.01*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Number of wind fields	*E.g. 200*
Source flag	0 off 1 on
Advection scheme	1: Centred difference 2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Wind inflation factor	*E.g. 1.0*

Algorithm for random number test

- Choose 1000 top-hat random numbers between 0 and "Random number variance". Output in file, *TopHatTest*.
- Choose 1000 random numbers from Gaussian distribution specified by mean "Random number mean" and variance "Random number variance". Output in file, *GaussianTest*.

Algorithm for Legendre polynomial orthogonality test

$$\text{Check } \frac{\pi}{180} \int_{\theta=-90}^{90} d\theta \hat{P}_l^m (\sin \theta\pi/180) \hat{P}_{l'}^m (\sin \theta\pi/180) \cos \theta\pi/180 \overset{?}{=} \delta_{ll'}$$

Algorithm inverse transform test

- Read in truth state, *Truth.nc*.
- Add Gaussian noise.
- Output noisy field, *TestFieldIn.nc*.
- Spectral transform to spectral space.
- Inverse spectral transform back to real space.
- Output the original field in real space in *TestField.nc*.
- Calculate and output the difference between final and original fields in *Difference.nc*.

Algorithm for adjoint tests

$$(\mathbf{Av})^T \mathbf{Av} \overset{?}{=} \mathbf{v}^T \mathbf{A}^T \mathbf{Av}$$

- Interpolate_1d
- Interpolate_2d_tracer
- Obs_Operator_t
- Spec2Real
- CVT
- SortHalos_field
- Model

Algorithm for t=0 implied covariance test

$$\mathbf{B} = (CVT)(CVT)^T$$

- Set-up the background spectra of the tracer and source using the above information.
- Read-in the tracer/source standard deviation fields, *BgStd.nc*.
- Output in file *StrucFunc.nc*

Algorithm for gradient test

$$J(\mathbf{x} + \delta\mathbf{x}) = J(\mathbf{x}) + \nabla J(\mathbf{x})\, \alpha\delta\mathbf{x} + O(\alpha\delta\mathbf{x})^2$$

$$\frac{J(\mathbf{x} + \delta\mathbf{x}) - J(\mathbf{x})}{\nabla J(\mathbf{x})\, \alpha\delta\mathbf{x}} = 1 + O(\alpha\delta\mathbf{x})^2$$

$$\therefore \lim_{\alpha \to 0} \frac{J(\mathbf{x} + \delta\mathbf{x}) - J(\mathbf{x})}{\nabla J(\mathbf{x})\alpha\delta\mathbf{x}} \to 1$$

- Output verbose description of operations, file *GradTest*
- Read-in background state, file *Background.nc*.
- Read-in observations, file *Observations*.
- Set-up the background spectra of the tracer and source using the above information.
- Read-in the tracer/source standard deviation fields, *BgStd.nc*.
- Calculate J and the gradient at the background and output *Forwardxxxx.nc*, *Adjointxxxx.nc*, *Minus_obs_grad.nc* and *Minus_bg_grad.nc* as part of the gradient calculation.
- Calculate a random state vector.
- Loop around different values for α.
    - Calculate the above quantity to file *GradTest*.

Forecast mode

Source.out	
F	
Number of wind fields	*E.g. 200*
Time difference between wind fields (seconds)	*E.g. 21600.0 (6 hours)*
Time step of integration scheme (seconds)	*E.g. 360.0 (6 minutes)*
Source flag	0 off 1 on
Advection scheme	1: Centred difference 2: Forward-upstream
Diffusion coefficient	*E.g. 120000.0*
Output frequency	Output every n steps to file ForeStepxxx.nc -1 for end only
Wind inflation factor	*E.g. 1.0*

Algorithm for forecast mode

- Read in initial conditions, *InitConds.nc*.
- Advect tracer.
- Output final state with filename, *Forecast.nc*.

The state vector and 4d-Var formulae

The augmented state vector

$$\vec{X}(t) = \begin{pmatrix} \vec{x}(t) \\ \vec{\rho}(t) \end{pmatrix}$$

The time evolution model

$$\vec{x}(t+1) = M_{t+1}\vec{x}(t) + \vec{\rho}(t)$$

$$\vec{\rho}(t+1) = \vec{\rho}(t)$$

The sensitivity of future states to the immediately earlier state

$$\delta\vec{x}(t+1) = \frac{\partial\vec{x}(t+1)}{\partial\vec{x}(t)}\delta\vec{x}(t) + \frac{\partial\vec{x}(t+1)}{\partial\vec{\rho}(t)}\delta\vec{\rho}(t)$$

$$= \mathbf{M}_{t+1}\delta\vec{x}(t) + \mathbf{N}_{t+1}\delta\vec{\rho}(t)$$

$$\delta\vec{\rho}(t+1) = \delta\vec{\rho}(t)$$

$$\begin{pmatrix} \delta\vec{x}(t+1) \\ \delta\vec{\rho}(t+1) \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{t+1} & \mathbf{N}_{t+1} \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \delta\vec{x}(t) \\ \delta\vec{\rho}(t) \end{pmatrix}$$

$$\delta\vec{X}(t+1) = \Omega_{t+1}\delta\vec{X}(t)$$

The adjoint operator

$$\delta\hat{\vec{X}}(t) = \Omega_{t+1}^{T}\delta\hat{\vec{X}}(t+1)$$

$$\begin{pmatrix} \delta\hat{\vec{x}}(t) \\ \delta\hat{\vec{\rho}}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{t+1}^{T} & 0 \\ \mathbf{N}_{t+1}^{T} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \delta\hat{\vec{x}}(t+1) \\ \delta\hat{\vec{\rho}}(t+1) \end{pmatrix}$$

The sensitivity of the state at time T to all earlier times

$$\delta\vec{x}(T)$$

$$= \frac{\partial\vec{x}(T)}{\partial\vec{x}(T-1)}\delta\vec{x}(T-1) + \frac{\partial\vec{x}(T)}{\partial\vec{\rho}(T-1)}\delta\vec{\rho}(T-1)$$

$$= \frac{\partial\vec{x}(T)}{\partial\vec{x}(T-1)}\left(\frac{\partial\vec{x}(T-1)}{\partial\vec{x}(T-2)}\delta\vec{x}(T-2) + \frac{\partial\vec{x}(T-1)}{\partial\vec{\rho}(T-2)}\delta\vec{\rho}(T-2)\right) + \frac{\partial\vec{x}(T)}{\partial\vec{\rho}(T-1)}\delta\vec{\rho}(T-2)$$

$$= \frac{\partial\vec{x}(T)}{\partial\vec{x}(T-1)}\left(\frac{\partial\vec{x}(T-1)}{\partial\vec{x}(T-2)}\left(\frac{\partial\vec{x}(T-2)}{\partial\vec{x}(T-3)}\delta\vec{x}(T-3) + \frac{\partial\vec{x}(T-2)}{\partial\vec{\rho}(T-3)}\delta\vec{\rho}(T-3)\right) + \frac{\partial\vec{x}(T-1)}{\partial\vec{\rho}(T-2)}\delta\vec{\rho}(T-3)\right) + \frac{\partial\vec{x}(T)}{\partial\vec{\rho}(T-1)}\delta\vec{\rho}(T-3)$$

$$\delta\vec{x}(T) = \mathbf{M}_T\mathbf{M}_{T-1}\ldots\mathbf{M}_1\delta\vec{x}(0) + \mathbf{N}_T\delta\vec{\rho}(0) + \sum_{t=1}^{T-1}\mathbf{M}_T\mathbf{M}_{T-1}\ldots\mathbf{M}_{t+1}\mathbf{N}_t\delta\vec{\rho}(0)$$

The 4d-Var gradient formula (without the source term)

$$\nabla_{x(0)}J = -\mathbf{B}^{-1}(\vec{x}_B - \vec{x}) - \sum_{t=0}^{T}\mathbf{M}_1^T\mathbf{M}_2^T\ldots\mathbf{M}_t^T\mathbf{H}_t^T\mathbf{R}_t^{-1}(\vec{y}(t) - H_t[\vec{x}(t)])$$

The 4d-Var gradient formula (with the source term)

$$\nabla_{X(0)}J = -\mathbf{B}^{-1}(\vec{X}_B - \vec{X}) - \sum_{t=0}^{T}\Omega_1^T\Omega_2^T\ldots\Omega_t^T\mathbf{H}_t^T\mathbf{R}_t^{-1}(\vec{y}(t) - H_t[\vec{x}(t)])$$

Simple advection scheme for the sources and sinks

$$\frac{\partial\chi}{\partial t} = -\vec{u}\cdot\nabla\chi + \rho$$

$$\frac{\chi_{ij}^{t+1} - \chi_{ij}^{t}}{\delta t} = -\frac{1}{2}\left(u_{ij}^{t}\frac{\chi_{i+1,j}^{t} - \chi_{ij}^{t}}{\delta x} + u_{i-1,j}^{t}\frac{\chi_{ij}^{t} - \chi_{i-1,j}^{t}}{\delta x} + v_{ij}^{t}\frac{\chi_{i,j+1}^{t} - \chi_{ij}^{t}}{\delta y} + v_{i,j-1}^{t}\frac{\chi_{ij}^{t} - \chi_{i,j-1}^{t}}{\delta y}\right) + \rho_{ij}$$

$$\chi_{ij}^{t+1} = \chi_{ij}^{t} - \frac{\delta t}{2}\left(u_{ij}^{t}\frac{\chi_{i+1,j}^{t} - \chi_{ij}^{t}}{\delta x} + u_{i-1,j}^{t}\frac{\chi_{ij}^{t} - \chi_{i-1,j}^{t}}{\delta x} + v_{ij}^{t}\frac{\chi_{i,j+1}^{t} - \chi_{ij}^{t}}{\delta y} + v_{i,j-1}^{t}\frac{\chi_{ij}^{t} - \chi_{i,j-1}^{t}}{\delta y}\right) + \delta t\rho_{ij}$$

The spherical harmonics

Expansion

$$f(\theta,\phi) = \sum_{l=0}^{L}\sum_{m=-l}^{l} F_l^m \exp im\frac{\theta\pi}{180}\hat{P}_l^m(\sin\frac{\phi\pi}{180})$$

Orthogonality relations for exponentials and associated Legendre polynomials

$$\frac{\pi}{180}\int_{\theta=0}^{360} d\theta \exp im\frac{\theta\pi}{180}\exp -im'\frac{\theta\pi}{180} = 2\pi\delta_{mm'}$$

$$\frac{\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\hat{P}_{l'}^m(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180} = \delta_{ll'}$$

Definition of $\hat{P}_l^m$

$$\hat{P}_l^m = P_l^m\left(\frac{2}{2l+1}\frac{(l+m)!}{(l-m)!}\right)^{-1/2}$$

Identity relating P_l^m and P_l^{-m}

$$P_m^{-m} = (-1)^m\frac{(l-m)!}{(l+m)!}P_l^m$$

Identity relating P_l^m and P_l^{-m} follows

$$\left(\frac{2}{2l+1}\frac{(l-m)!}{(l+m)!}\right)^{1/2}\hat{P}_l^{-m} = (-1)^m\frac{(l-m)!}{(l+m)!}\left(\frac{2}{2l+1}\frac{(l+m)!}{(l-m)!}\right)^{1/2}\hat{P}_l^m$$

$$\frac{(l-m)!}{(l+m)!}\hat{P}_l^{-m} = (-1)^m\frac{(l-m)!}{(l+m)!}\hat{P}_l^m$$

$$\hat{P}_l^{-m} = (-1)^m\hat{P}_l^m$$

Inverse expansion (using orthogonality)

$$f(\theta,\phi) = \sum_{l'=0}^{L}\sum_{m'=-l'}^{l'} F_{l'}^{m'}\exp im'\frac{\theta\pi}{180}\hat{P}_{l'}^{m'}(\sin\frac{\phi\pi}{180})$$

$$\frac{\pi}{180}\int_{\theta=0}^{360} d\theta\exp -im\frac{\theta\pi}{180}\frac{\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}f(\theta,\phi)$$

$$= \sum_{l'=0}^{L}\sum_{m'=-l'}^{l'} F_{l'}^{m'}\frac{\pi}{180}\int_{\theta=0}^{360} d\theta\exp -im\frac{\theta\pi}{180}\exp im'\frac{\theta\pi}{180}\frac{\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\hat{P}_{l'}^{m'}(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}$$

$$= \sum_{l'=0}^{L}\sum_{m'=-l'}^{l'} F_{l'}^{m'}2\pi\delta_{mm'}\frac{\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\hat{P}_{l'}^{m'}(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}$$

$$= 2\pi\sum_{l'=0}^{L} F_{l'}^m\frac{\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\hat{P}_{l'}^m(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}$$

$$= 2\pi\sum_{l'=0}^{L} F_{l'}^m\delta_{ll'} = 2\pi F_l^m$$

$$\therefore\quad F_l^m = \frac{1}{2}\frac{\pi}{180^2}\int_{\theta=0}^{360} d\theta\exp -im\frac{\theta\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}f(\theta,\phi)$$

Expansion for real functions

$$F_l^{m^*} = \frac{1}{2}\frac{\pi}{180^2}\int_{\theta=0}^{360} d\theta\exp im\frac{\theta\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^m(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}f^*(\theta,\phi)\qquad f^*(\theta,\phi) = f(\theta,\phi)$$

$$F_l^{-m^*} = \frac{1}{2}\frac{\pi}{180^2}\int_{\theta=0}^{360} d\theta\exp -im\frac{\theta\pi}{180}\int_{\phi=-90}^{90} d\phi\hat{P}_l^{-m}(\sin\frac{\phi\pi}{180})\cos\frac{\phi\pi}{180}f(\theta,\phi)$$

$$= (-1)^m \frac{1}{2} \frac{\pi}{180^2} \int_{\theta=0}^{360} d\theta \, \exp -im\frac{\theta\pi}{180} \int_{\phi=-90}^{90} d\phi \hat{P}_l^m (\sin \frac{\phi\pi}{180}) \cos \frac{\phi\pi}{180} f(\theta, \phi)$$

$$= (-1)^m F_l^m$$

Recovering the function using only the positive ms

$$f(\theta,\phi) = \sum_{l=0}^{L} \sum_{m=-l}^{l} F_l^m \exp im\frac{\theta\pi}{180} \hat{P}_l^m (\sin \frac{\phi\pi}{180})$$

$$= \sum_{l=0}^{L} \left(\sum_{m=-l}^{-1} F_l^m \exp im\frac{\theta\pi}{180} \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) + \sum_{m=1}^{l} F_l^m \exp im\frac{\theta\pi}{180} \hat{P}_l^m (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{-m=-l}^{-1} F_l^{-m} \exp -im\frac{\theta\pi}{180} \hat{P}_l^{-m} (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) + \sum_{m=1}^{l} F_l^m \exp im\frac{\theta\pi}{180} \hat{P}_l^m (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{m=-l}^{1} (-1)^m F_l^{m^*} \exp -im\frac{\theta\pi}{180} (-1)^m \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) + \sum_{m=1}^{l} F_l^m \exp im\frac{\theta\pi}{180} \hat{P}_l^m (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{m=1}^{l} \left[F_l^{m^*} \exp -im\frac{\theta\pi}{180} + F_l^m \exp im\frac{\theta\pi}{180} \right] \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{m=1}^{l} \left[\left(F_l^m \exp im\frac{\theta\pi}{180} \right)^* + F_l^m \exp im\frac{\theta\pi}{180} \right] \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{m=1}^{l} 2 \, \mathrm{Re} \left(F_l^m \exp im\frac{\theta\pi}{180} \right) \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) \right)$$

$$= \sum_{l=0}^{L} \left(\sum_{m=1}^{l} 2 \left(\mathrm{Re}(F_l^m) \cos m\frac{\theta\pi}{180} - \mathrm{Im}(F_l^m) \sin m\frac{\theta\pi}{180} \right) \hat{P}_l^m (\sin \frac{\phi\pi}{180}) + F_l^0 \hat{P}_l^0 (\sin \frac{\phi\pi}{180}) \right)$$

The conjugate gradient minimization algorithm

See [1] for details of the conjugate gradient algorithm.

1 Set the initial guess, χ_0
2 Calculate the initial residual, $\boldsymbol{r}_0 = -\nabla J(\chi_0)$
3 Set the initial search direction, $\boldsymbol{p}_0 = \boldsymbol{r}_0$
4 Set the iteration number to zero, $i = 0$
5 Do a line minimization along the search direction, $\boldsymbol{p}_i$ to determine χ_{i+1} - see Eqs. (S5.1) and (S5.3)
6 Calculate the new residual, $\boldsymbol{r}_{i+1} = -\nabla J(\chi_{i+1})$
7 Calculate the new search direction, $\boldsymbol{p}_{i+1}$ - see Eqs. (S7.1) and (S7.2)
8 Set $i \rightarrow i + 1$
9 Go to 5 unless converged

Notes about some of these steps are shown below.

Notes about step 5

Equation (1.8) of [1] says that

$$\chi_{i+1} = \chi_i + \beta_i \boldsymbol{p}_i \qquad (S5.1)$$

The scalar a_i is to be determined by line minimization.

$$J = J(\chi_i) \qquad J^+ = J(\chi_i + \delta \cdot \boldsymbol{p}_i) \qquad J^- = J(\chi_i - \delta \cdot \boldsymbol{p}_i) \qquad (S5.2)$$

$$\beta_i = \frac{1}{2}\frac{J^- - J^+}{J^- + J^+ - 2J}\delta \qquad (S5.3)$$

Notes about step 7
Equations (1.26) and (1.28) of [1] say that

$$\boldsymbol{p}_{i+1} = \boldsymbol{r}_{i+1} + \alpha_{ii}\boldsymbol{p}_i \qquad (S7.1)$$

$$\text{where} \qquad \alpha_{ii} = \frac{\boldsymbol{r}_{i+1}^T\boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^T\boldsymbol{r}_i} \qquad (S7.2)$$

Estimating the analysis errors from the conjugate gradient calculation

The conjugate gradient algorithm may be summarised as (S5.1), (S5.3), (S.6), (S7.1) and (S7.2). The equations bear some similarity with the Lanczos method [2][3]. To see this, consider the following.

Taylor series and its derivative (where $\mathbf{A}_\chi$ is the Hessian in the χ-representation).

$$J(\chi_{i+1}) = J(\chi_i) + \nabla_\chi^{\mathrm{T}} J[\chi_i](\chi_{i+1} - \chi_i) + \frac{1}{2}(\chi_{i+1} - \chi_i)^{\mathrm{T}}\mathbf{A}_\chi(\chi_{i+1} - \chi_i) \qquad (1)$$

$$\nabla_\chi J[\chi_{i+1}] = \nabla_\chi J[\chi_i] + \mathbf{A}_\chi(\chi_{i+1} - \chi_i)$$

$$-\boldsymbol{r}_{i+1} = -\boldsymbol{r}_i + \mathbf{A}_\chi(\chi_{i+1} - \chi_i)$$

$$\therefore \qquad \mathbf{A}_\chi(\chi_{i+1} - \chi_i) = -(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i) \qquad (2)$$

Eliminate the search directions (the $\boldsymbol{p}_i$) from (S5.1) and (S7.1)

$$(S5.1) \Rightarrow \chi_{i+1} = \chi_i + \beta_i\boldsymbol{p}_i$$

$$(S5.1) \Rightarrow \chi_i = \chi_{i-1} + \beta_{i-1}\boldsymbol{p}_{i-1}$$

$$(S7.1) \Rightarrow \boldsymbol{p}_i = \boldsymbol{r}_i + \alpha_{i-1,i-1}\boldsymbol{p}_{i-1}$$

$$\frac{\chi_{i+1} - \chi_i}{\beta_i} = \boldsymbol{r}_i + \alpha_{i-1,i-1}\frac{\chi_i - \chi_{i-1}}{\beta_{i-1}} \qquad (3)$$

Act with $\mathbf{A}_\chi$ and use (2)

$$\frac{\mathbf{A}_\chi(\chi_{i+1} - \chi_i)}{\beta_i} = \mathbf{A}_\chi\boldsymbol{r}_i + \alpha_{i-1,i-1}\frac{\mathbf{A}_\chi(\chi_i - \chi_{i-1})}{\beta_{i-1}}$$

$$\frac{-(\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)}{\beta_i} = \mathbf{A}_\chi\boldsymbol{r}_i - \alpha_{i-1,i-1}\frac{\boldsymbol{r}_i - \boldsymbol{r}_{i-1}}{\beta_{i-1}} \qquad (4)$$

Let $\boldsymbol{q}_i = c_i\boldsymbol{r}_i$ and rearrange

$$c_i = (\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i)^{-1/2} \qquad (5)$$

$$\frac{-(\boldsymbol{q}_{i+1}/c_{i+1} - \boldsymbol{q}_i/c_i)}{\beta_i} = \mathbf{A}_\chi\frac{\boldsymbol{q}_i}{c_i} - \alpha_{i-1,i-1}\frac{\boldsymbol{q}_i/c_i - \boldsymbol{q}_{i-1}/c_{i-1}}{\beta_{i-1}}$$

$$-\frac{1}{c_{i+1}\beta_i}\boldsymbol{q}_{i+1} + \frac{1}{c_i\beta_i}\boldsymbol{q}_i = \frac{1}{c_i}\mathbf{A}_\chi\boldsymbol{q}_i - \frac{\alpha_{i-1,i-1}}{\beta_{i-1}}\left(\frac{\boldsymbol{q}_i}{c_i} - \frac{\boldsymbol{q}_{i-1}}{c_{i-1}}\right)$$

$$\boldsymbol{q}_{i+1} - \frac{c_{i+1}}{c_i}\boldsymbol{q}_i = -\frac{c_{i+1}\beta_i}{c_i}\mathbf{A}_\chi\boldsymbol{q}_i + \frac{c_{i+1}\beta_i\alpha_{i-1,i-1}}{\beta_{i-1}}\left(\frac{\boldsymbol{q}_i}{c_i} - \frac{\boldsymbol{q}_{i-1}}{c_{i-1}}\right)$$

$$\boldsymbol{q}_{i+1} = -\frac{c_{i+1}\beta_i}{c_i}\left(\mathbf{A}_\chi\boldsymbol{q}_i - \frac{c_i\alpha_{i-1,i-1}}{\beta_{i-1}}\left(\frac{\boldsymbol{q}_i}{c_i} - \frac{\boldsymbol{q}_{i-1}}{c_{i-1}}\right) - \frac{1}{\beta_i}\boldsymbol{q}_i\right)$$

$$= -\frac{c_{i+1}\beta_i}{c_i}\left(\mathbf{A}_\chi \boldsymbol{q}_i - \frac{c_i \alpha_{i-1,i-1}}{\beta_{i-1}}\frac{\boldsymbol{q}_i}{c_i} + \frac{c_i \alpha_{i-1,i-1}}{\beta_{i-1}}\frac{\boldsymbol{q}_{i-1}}{c_{i-1}} - \frac{1}{\beta_i}\boldsymbol{q}_i\right)$$

$$= -\frac{c_{i+1}\beta_i}{c_i}\left(\mathbf{A}_\chi \boldsymbol{q}_i - \frac{\alpha_{i-1,i-1}}{\beta_{i-1}}\boldsymbol{q}_i + \frac{c_i \alpha_{i-1,i-1}}{c_{i-1}\beta_{i-1}}\boldsymbol{q}_{i-1} - \frac{1}{\beta_i}\boldsymbol{q}_i\right)$$

$$= -\frac{c_{i+1}\beta_i}{c_i}\left(\mathbf{A}_\chi \boldsymbol{q}_i - \left[\frac{\alpha_{i-1,i-1}}{\beta_{i-1}} + \frac{1}{\beta_i}\right]\boldsymbol{q}_i + \frac{c_i \alpha_{i-1,i-1}}{c_{i-1}\beta_{i-1}}\boldsymbol{q}_{i-1}\right) \tag{6}$$

The Lanczos method for operator $\mathbf{A}_\chi$ is [3]

$$\boldsymbol{q}_{i+1} = \frac{1}{\mathbf{A}^q_{i+1,i}}\left(\mathbf{A}_\chi \boldsymbol{q}_i - \mathbf{A}^q_{ii}\boldsymbol{q}_i - \mathbf{A}^q_{i-1,i}\boldsymbol{q}_{i-1}\right) \tag{7}$$

where $\mathbf{A}^q_{i,i}$ etc. are the matrix elements of $\mathbf{A}_\chi$ in the $\boldsymbol{q}$-representation, $\mathbf{A}^q_{ij} = \boldsymbol{q}_i^{\mathrm{T}}\mathbf{A}_\chi \boldsymbol{q}_j$. Comparing (9) and (10)

$$\mathbf{A}^q_{i+1,i} = -\frac{c_i}{c_{i+1}\beta_i} \tag{8}$$

$$\mathbf{A}^q_{ii} = \frac{\alpha_{i-1,i-1}}{\beta_{i-1}} + \frac{1}{\beta_i} \tag{9}$$

$$\mathbf{A}^q_{i-1,i} = -\frac{c_i \alpha_{i-1,i-1}}{c_{i-1}\beta_{i-1}} \tag{10}$$

Lanczos theory states that the $\boldsymbol{q}$-vectors are orthonormal and the Hessian matrix in this representation - as (8)-(10) is tridiagonal.

Are these consistent given that $\mathbf{A}_\chi$ is Hermitian? Is it true that

$$(\mathbf{A}_\chi)_{i,i+1} \overset{?}{=} (\mathbf{A}_\chi)_{i+1,i}$$

where the LHS is found from (10) and the RHS from (8)?

$$-\frac{c_{i+1}\alpha_{i,i}}{c_i \beta_i} \overset{?}{=} -\frac{c_i}{c_{i+1}\beta_i}$$

$$\text{where} \quad \alpha_{ii} = \frac{\boldsymbol{r}_{i+1}^{\mathrm{T}}\boldsymbol{r}_{i+1}}{\boldsymbol{r}_i^{\mathrm{T}}\boldsymbol{r}_i} = \frac{c_i^2}{c_{i+1}^2} \qquad \text{using (5)}$$

$$-\frac{c_{i+1}\alpha_{i,i}}{c_i} \overset{?}{=} -\frac{c_i}{c_{i+1}}$$

$$\frac{c_{i+1}}{c_i}\frac{c_i^2}{c_{i+1}^2} \overset{?}{=} \frac{c_i}{c_{i+1}}$$

$$\frac{c_i}{c_{i+1}} = \frac{c_i}{c_{i+1}} \qquad \text{confirmed.}$$

How can this be used to estimate the analysis error?

Analysis error is the inverse Hessian. This can be calculated from the above procedure and transformed to real-space. Let $\mathbf{L}^q$ be the eigenvectors of the tridiagonal $\mathbf{A}^q$ and Λ be the eigenvalues. Let $\mathbf{Q}$ be the matrix of $\boldsymbol{q}$-vectors. This allows states in the $\boldsymbol{q}$, χ and $\boldsymbol{x}$-representations to be related

$$\boldsymbol{v}_\chi = \mathbf{Q}\boldsymbol{v}_q, \tag{11}$$

$$\boldsymbol{v}_x = \mathbf{U}\boldsymbol{v}_\chi, \tag{12}$$

$$\text{where} \quad \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}. \tag{13}$$

The Hessian in the $\boldsymbol{q}$-representation may be decomposed as

$$\mathbf{L}^{q\mathrm{T}}\mathbf{A}^q\mathbf{L}^q = \Lambda. \tag{14}$$

The Hessians in $\boldsymbol{q}$ and χ-spaces are related via

$$\mathbf{A}^{q} = \mathbf{Q}^{\mathrm{T}}\mathbf{A}_{\chi}\mathbf{Q}, \tag{15}$$

and the Hessians in χ and $\boldsymbol{x}$-spaces are related via

$$\mathbf{A}_{\chi} = \mathbf{U}^{\mathrm{T}}\mathbf{A}_{x}\mathbf{U}. \tag{16}$$

Combining (14), (15) and (16)

$$(\mathbf{UQL}^{q})^{\mathrm{T}}\,\mathbf{A}_{x}\mathbf{UQL}^{q} = \Lambda. \tag{17}$$

The inverse of (17) is

$$(\mathbf{UQL}^{q})^{-1}\,\mathbf{A}_{x}^{-1}\,(\mathbf{UQL}^{q})^{-\mathrm{T}} = \Lambda^{-1}$$

$$\begin{aligned}
\mathbf{A}_{x}^{-1} &= \mathbf{UQL}^{q}\Lambda^{-1}\,(\mathbf{UQL}^{q})^{\mathrm{T}} \\
&= \mathbf{UQL}^{q}(\Lambda^{-1} - \mathbf{I} + \mathbf{I})\,(\mathbf{UQL}^{q})^{\mathrm{T}} \\
&= \mathbf{UQL}^{q}(\Lambda^{-1} - \mathbf{I})\,(\mathbf{UQL}^{q})^{\mathrm{T}} + \mathbf{UQL}^{q}\,(\mathbf{UQL}^{q})^{\mathrm{T}} \\
&\approx \mathbf{UQL}^{q}(\Lambda^{-1} - \mathbf{I})\,(\mathbf{UQL}^{q})^{\mathrm{T}} + \mathbf{B}_{x}.
\end{aligned} \tag{18}$$

How to set-up the system for assimilation (from scratch)

- Generate the truth state:
  - If require system that respects continents, run MakeWorldData.out ("UnadvectTruth.nc").
  - In all cases, run option 'I' of Source.out ("Truth.nc").
- Generate background standard deviations, run option 'S' of Source.out ("BgStd.nc").
- Decide on the peak spectral variance and lengthscales for the tracer and source fields (this will require experimentation).
- Create a background state with the correct characteristics, run option 'B' of Source.out ("Background.nc").
- Run forecasts from truth and background (option 'F' of Source.out) to check that the model behaves reasonably with these states.
- Generate observations, option 'O' of Source.out ("Observations").
- Do a variational assimiltion run, option 'V' of Source.out ("AnalInc.nc", "Anal.nc", "VarStats", "VarStatus", ["Anal_err.nc"]).

Sample of scientific questions

- Is it possible to find out the source and sink field structure with little or no prior knowledge?
- What is the accuracy and resolution of the source/sink data?
- What does the background error coupling of the tracer and source look like?
- Is there a way of modelling the background error coupling of the tracer and source?
- Can we design an optimal observation network for sources and sinks?
- Budget calculations.

References

[1] Bannister R.N., 2003, Iterative solvers and minimization algorithms.

[2] Fisher M. and Courtier P., 1995, Estimating the covariance matrices of analysis and forecast error in variational data assimilation.

[3] Bannister R.N., 2003, Lanczos method for Hermitian and non-Hermitian operators.