ABC vn 1.4da: Guide to running the research modelling and data assimilation system

May 26, 2020

Ross Bannister, National Centre for Earth Observation, University of Reading, Reading, UK
r.n.bannister@reading.ac.uk

1 Introduction

1.1 The ABC model

The ABC model and data assimilation system (vn 1.4da) is a combined 2D (longitude/height) convective-scale toy model (currently dry dynamics) and variational data assimilation system. The model equations are as follows:

$$\frac{\partial u}{\partial t} + B\mathbf{u} \cdot \nabla u + C\frac{\partial \tilde{\rho}'}{\partial x} - fv = 0, \tag{1a}$$

$$\frac{\partial v}{\partial t} + B\mathbf{u} \cdot \nabla v + fu = 0, \tag{1b}$$

$$\frac{\partial w}{\partial t} + B\mathbf{u} \cdot \nabla w + C\frac{\partial \tilde{\rho}'}{\partial z} - b' = 0, \tag{1c}$$

$$\frac{\partial \tilde{\rho}'}{\partial t} + B\nabla \cdot (\tilde{\rho}\mathbf{u}) = 0, \tag{1d}$$

$$\frac{\partial b'}{\partial t} + B\mathbf{u} \cdot \nabla b' + A^2 w = 0. \tag{1e}$$

The prognostic variables are as follows: u is the zonal wind, v is the meridional wind, w is the vertical wind, $\mathbf{u} = (u, v, w)$ is the wind vector, $\tilde{\rho}$ is a density-like variable (where $\tilde{\rho}'$ is the perturbation, $\tilde{\rho} = \tilde{\rho}_0 + \tilde{\rho}'$, where in this model, $\tilde{\rho}_0 = 1$), and b' is a buoyancy-like variable (for meteorologists, b' is related to potential temperature, θ', by $b' = g\theta'/\theta_{\mathrm{R}}$, where g is the acceleration due to gravity and θ_{R} is the reference potential temperature of 273K). The dimension variables are as follows: x is longitudinal distance, z is vertical distance, and t is time. Constant parameters to be chosen by the user are as follows: A (units s^{-1}) is the static stability (equivalent to the pure gravity wave frequency), B (dimensionless) multiplies the advection and divergence terms, and C (units $\mathrm{m}^2\mathrm{s}^{-2}$) relates density perturbations to pressure perturbations, $p' = C\rho_0\tilde{\rho}'$, where ρ_0 is a reference density. The value of $\sqrt{BC}$ is the pure acoustic wave speed). These parameters give the model its "ABC" name. The remaining constant is f, which is the Coriolis parameter.

There is also a tracer transport equation, which advects a tracer, q, with the wind vector $\mathbf{u}$, and not by the modified winds, $B\mathbf{u}$:

$$\frac{\partial q}{\partial t} + \mathbf{u} \cdot \nabla q = 0. \tag{2}$$

The model is run in a 2D slice (longitude/height) geometry. All variables are considered constant in the meridional direction. The model grid is an Arakawa-C grid in the horizontal and a Charney-Phillips grid in the vertical (Fig. 1). The horizontal resolution of the model is 1.5km, there are 360

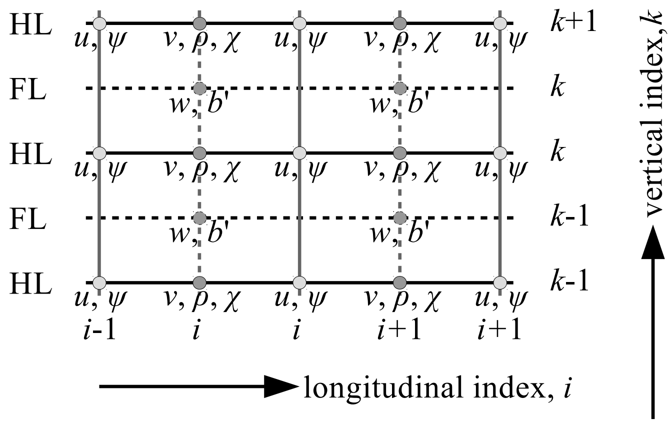

Figure 1: The arrangement of variables on the toy model's grid: an Arakawa-C grid in the horizontal and a Charney-Phillips grid in the vertical. Note the abbreviations: FL=Full Level and HL=Half Level.

grid-points in the horizontal, and 60 vertical levels. The scientific rational for this model is given in **(author?)** [1].

The code is set-up as a number of master subroutines, each designed to do a particular job (such as running the model from specified initial conditions, to performing a data assimilation cycle; there are other routines, e.g. for calibrating the control variable transform, and generating synthetic observation). This guide comprises the following sections, in Sect. 2 we give the full list of master routines; in Sect. 3 we describe how the software is downloaded and installed (also stating which software libraries are required), and in Sect. 4 (which makes up the bulk of this documentation) we describe how each master routine is used.

1.2 Difference between this and the previous software version

Although the code for the ABC model is exactly the same as in the previous released version of this system (vn 1.0), the organisation of the code is different (e.g. the master program Main.f90 in vn 1.0 is only for running the model and doing linear analysis; these have separate master routines in vn 1.4da). Another difference is that, while vn 1.0 used NAG routine libraries for eigen analysis and fast Fourier transforms, vn 1.4da uses free software libraries as documented in Sect. 3.

2 List of master routines

The code is written in Fortran-90, and the master routines are run inside bash script wrappers. The recommeded operating system to run the code is Linux. Scientific documentation is provided on the model [1] and on the data assimilation (in preparation). Each of these master routines is associated with a top-level Fortran 90 subroutine (*.f90*), and for some a top-level python routine (*.py*) for plotting the results. There are also examples (labelled with the respective master routine name) available.

Master routine	**Purpose**
Master_prepareABC_InitState (Sect. 4.1)	Inputs a UM dump and generates a single 2D longitude/height slice set of fields that is suitable as a set of initial initialised initial conditions for the ABC model.
Master_RunNLModel (Sect. 4.2)	Makes a single run of the ABC model from a specified set of initial conditions.
Master_Linear_Analysis (Sect. 4.3)	Analyses the linear modes of the ABC model.
Master_Calibration (Sect. 4.4)	Runs any of the required stages of computation to compute data needed to specify the background error covariance matrix used in the data assimilation (specifies all aspects of the control variable transform according to implemented options). There are nominally five stages, and so this code is run five times in succession.
Master_TestSuite (Sect. 4.5)	Tests various aspects of the components of the data assimilation system (adjoint and inverse tests of the control variable transforms, and linearization tests of the ABC model and observation operators).
Master_ImpliedCov (Sect. 4.6)	Computes a selection of implied covariances (selected columns of $\mathbf{U}\mathbf{U}^{\mathrm{T}}$) between the model variables.
Master_RawCov (Sect. 4.7)	Computes raw covariances from a population of states (can be compared to the implied covariances).
Master_MakeBgObs (Sect. 4.8)	Generates a set of synthetic observations for assimilating (and outputs the associated 'truth' trajectory), and a synthetic background state.
Master_Assimilate (Sect. 4.9)	Performs a variational data assimilation run.

3 Downloading and installing the software

3.1 Required libraries on host system

This sofware requires the following free software and libraries to be installed on the host system.

- Fortran compiler (f95). This is needed to compile the Fortran-90 code into executable files.
  - Net CDF library. This is needed to handle the input and output of fields.
  - FFTpack. This is needed to perform the fast Fourier transforms.
  - LApack.
  - tmglib.
  - refblas.
- Python v2. This is needed to manage the graphical diagnostics.
  - matplotlib.

3.2 Contents of ABC download

The following sets of files are included with this software in the respective directories.

- ABCvn1.4/src
  - Fortran-90 source code (multiple .f90 files).
  - Interface files for subroutines that have optional arguments (multiple .interface files).
  - makefile (to manage the compilation of the software, depending upon the master routine to be run).
- ABCvn1.4/graphics
  - python source code (multiple .py files).
- ABCvn1.4/examples
  - Sample data and namelists, depending upon the master routine to be run.
- ABCvn1.4/docs
  - Documentation of the system.
- ABCvn1.4/scripts
  - Example bash scripts (one to allow assimilation cycles to be run, and another to generate ensembles from an existing calibrated assimilation system).

3.3 Compiling the code

Once the *src* directory is downloaded, go into this directory and type

```
make all
```

to compile all programs. Alternatively to compile just one master program (e.g. Master_Assimilate) and type

```
make Master_Assimilate.out
```

Further guidance is given below.

4 Using the master routines

Each master routine is described in this section. This includes how to compile the master routine and dependent code, and how to run it. The namelist variables are defined, which includes mention of the required input files, the output files, and how the outputs can be inspected.

4.1 Master_PrepareABC_InitState

Inputs a UM dump and generates a single 2D longitude/height slice set of fields that is suitable as a set of initial initialised initial conditions for the ABC model.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_PrepareABC_InitState.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_PrepareABC_InitState.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_PrepareABC_InitialState/UserOptions.nl.

Variable	Type	Description	Default	Notes
Init_ABC_opt	integer	How to create ABC initial conditions.	No default	1=slice of UM; 2=zero apart from pres pert bubble; 3=sum of 1,2; 4=zero apart from buoy pert bubble; 5=sum of 1,4; 6=sum of 2,4; 7=sum of 1,2,4.
datadirUM	string	Directory containing UM data.	No default	Init_ABC_opt=1,3,5
init_um_file	string	UM data filename (expected in above directory).	No default	Init_ABC_opt=1,3,5
latitude	integer	Index of single latitude to extract from UM file.	144	Init_ABC_opt=1,3
Regular_vert_grid	logical	Used to set a regularly-spaced vertical grid.	.TRUE.	
A	double precision	Model parameter (pure gravity wave frequency).	0.02	s^{-1}
B	double precision	Model parameter (modulation of the divergent and advection terms).	0.01	

C	double precision	Model parameter (proportionality constant for the equation of state).	100000.0	m^2s^{-2}
f	double precision	Model parameter (Coriolis parameter).	0.0001	s^{-1}
source_x	integer	To specify horizontal grid box of centre of press/buoy pert.	180	Init_ABC_opt=2,3,4,5
source_z	integer	To specify vertical grid box of centre of press/buoy pert.	30	Init_ABC_opt=2,3,4,5
press_amp	double precision	Amplitude of pressure perturbation	0.01	Init_ABC_opt=2,3
buoy_amp	double precision	Amplitude of buoyancy perturbation	0.1	Init_ABC_opt=4,5
x_scale	integer	No. of horizontal grids to describe size of press/buoy pert.	80	Init_ABC_opt=2,3,4,5
z_scale	integer	No. of vertical grids to describe size of press/buoy pert.	3	Init_ABC_opt=2,3,4,5
Adv_tracer	logical	Used to switch on/off tracer advection.	.FALSE.	Sets up a 4×5 grid of point initial tracer positions.
gravity_wave_switch	logical	Used to switch on/off setting of $u = 0$ to simulate gravity waves.	.FALSE.	
BoundSpread	double precision	No. of horizontal grid points to spread boundary discontinuity for periodic boundary conditions.	50.0	
datadirABC_out	string	Main output directory.		
init_ABC_file	string	Output filename (in above directory).		

Input and output files

- Inputs
  - Suitable Unified Model (UM) dump for the Southern UK region (datadirUM/init_um_file). This is a netcdf file of fields of 360 longitudes, 287/288 latitudes, and 70/71 vertical levels. The file contains the following: u (zonal wind), v (meridional wind), dz_dt (vertical wind, w), $unspecified$ (density, ρr_{E}^2, where r_{E} is the Earth's radius), $theta$ (potential temperature, θ), $field7$ (exner pressure, Π), and ht (2D orographic height field). The dimension names are x (longitude axis for u), x_1 (longitude axis for v, w, ρr_{E}^2, θ, Π, and ht), y (latitude axis for u, θ), y_1 (latitude axis for v, w, ρr_{E}^2, θ, Π, and ht), $hybrid\_ht$ (vertical axis for u, v, and ρr_{E}^2), $hybrid\_ht\_1$ (vertical axis for θ), $hthybrid\_ht\_2$ (vertical axis for w), $hybrid\_ht\_3$ (vertical axis for Π), and $surface$ (vertical axis for ht). The complete filename is datadirUM/init_um_file.

- Outputs
  - Initial dump for the ABC model. This is a netcdf file of fields of 360 longitudes and 60 vertical levels (datadirABC_out/init_ABC_file). The file contains the following: u, v, w, ρ', b', $\rho = 1 + \rho'$, b_{eff}, $tracer$, geo_imbal (geostrophic imbalance), $hydro\_imbal$ (hydrostatic imbalance), $wmom\_source$ (vertical momentum source), $horiz\_div$ (horizontal divergence), $horiz\_vort$ (horizontal vorticity), E_{k} (total kinetic energy), E_{b} (total buoyant energy), E_{e} (elastic energy), and E (total energy, $E = E_{\mathrm{k}} + E_{\mathrm{b}} + E_{\mathrm{e}}$). The dimension names are $longs\_u$ (horizontal axis for u), $longs\_v$ (horizontal axis for other fields), $half\_level$

(vertical axis for u, v, ρ', ρ, *tracer*, *geo_imbal*, *horiz_div*, and *horiz_vort*), *full_level* (vertical axis for w, b', b_{eff}, *hydro_imbal*, and *wmom_source*.

The meanings of the symbols in the output file are described in **(author?)** [1]. In particular, the model equations are given as Eqs. (15) of that reference and the grid positions are shown in Fig. 1 of that reference.

Graphics tools

- The python code *PlotModelFields.py* can be used to plot the initial ABC model state.

4.2 Master_RunNLModel

Makes a single run of the ABC model from a specified set of initial conditions. The model equations are given as Eqs. (15) of **(author?)** [1] (reproduced near the beginning of this document), the boundary conditions are specified in Sect. 3.2, and the numerical integration scheme is described in Sect. 3.3.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_RunNLModel.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_RunNLModel.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_RunNLModel/UserOptions.nl.

Variable	Type	Description	Default	Notes
datadirABC_in	string	Input directory.		
init_ABC_file	string	Input filename (in above directory).		
A	double precision	Model parameter (pure gravity wave frequency).	0.02	s^{-1}
B	double precision	Model parameter (modulation of the divergent and advection terms).	0.005	
C	double precision	Model parameter (proportionality constant for the equation of state).	100000.0	m^2s^{-2}
f	double precision	Model parameter (Coriolis parameter).	0.0001	s^{-1}
dt	double precision	Model time step size.	4.0	s
runlength	double precision	Length of integration.	60.0	s
ndumps	integer	The number of times to dump the model state throughout runlength.	10	
Adv_tracer	logical	Used to switch on/off tracer advection.	.FALSE.	
Lengthscale_diagnostics	logical	Used to switch on/off computation of characteristic lengthscales of variables at the final time (and other diagnostics).	.FALSE.	
datadirABC_out	string	Main output directory.		

output_ABC_file	string	Output file (model trajectory, in above directory).
diagnostics_file	string	Diagnostics file (in above directory).

Note that if *runlength*=0.0 and *ndumps*=0, the model is not run, and the output of the code (datadirABC_out/output_ABC_file) is formed from the initial conditions, i.e. the fields that comprise the last time in the input ABC file (datadirABC_in/init_ABC_file).

Input and output files

- Inputs
  - Suitable ABC dump (datadirABC_in/init_ABC_file, e.g. output by Master_PrepareABC_InitState in Sect. 4.1). The latest time present in this file is used as the initial conditions for the model.
- Outputs
  - Time sequence of ABC model trajectory (datadirABC_out/output_ABC_file). This is a netcdf file of fields as the initial dump, but with multiple times.
  - Diagnostics file (datadirABC_out/diagnostics_file). This contains a model-time-step-by-model-time-step output of each component of energy (time, E_{k}, E_{b}, E_{e}, E, see Sect. 4.1), followed by diagnostics for the last timestep (if Lengthscale_diagnostics is set).

Graphics tools

- The python code *PlotModelFields.py* can be used to plot the ABC model state for a specified output time step.
- The python code *PlotEnergy.py* can be used to plot the total energy of the run, as a function of time.

4.3 Master_Linear_Analysis

Analyses the linear modes of the ABC model. A description of the linear analysis is given in Sect. 4 of **(author?)** [1].

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_Linear_Analysis.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_Linear_Analysis.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_Linear_Analysis/UserOptions.nl.

Variable	Type	Description	Default	Notes
A	double precision	Model parameter (pure gravity wave frequency).	0.02	s^{-1}
B	double precision	Model parameter (modulation of the divergent and advection terms).	0.005	
C	double precision	Model parameter (proportionality constant for the equation of state).	100000.0	m^2s^{-2}
f	double precision	Model parameter (Coriolis parameter).	0.0001	s^{-1}
H	double precision	Model domain height.	14862.01	m
datadirLinearAnal	string	Output directory.		

Input and output files

- Inputs
  - No inputs.
- Outputs
  - Gravity wave frequencies file (datadirLinearAnal/grav_frequency.dat). Given as a function of horizontal and vertical wavenumbers.
  - Acoustic wave frequencies file (datadirLinearAnal/acou_frequency.dat). Given as a function of horizontal and vertical wavenumbers.
  - Gravity wave speed (in the horizontal) file (datadirLinearAnal/hori_grav_speed.dat). Given as a function of horizontal and vertical wavenumbers.

- Acoustic wave speed (in the horizontal) file (datadirLinearAnal/hori_acou_speed.dat). Given as a function of horizontal and vertical wavenumbers.
- Gravity wave speed (in the vertical) file (datadirLinearAnal/vert_grav_speed.dat). Given as a function of horizontal and vertical wavenumbers.
- Acoustic wave speed (in the vertical) file (datadirLinearAnal/vert_acou_speed.dat). Given as a function of horizontal and vertical wavenumbers.

Graphics tools

- The python code *PlotWaveSpeeds.py* can be used to plot the wave frequencies and wave group speeds.

4.4 Master_Calibration

Runs any of the required stages of computation to compute data needed to specify the background error covariance matrix used in the data assimilation (specifies all aspects of the control variable transform according to implemented options). There are nominally five stages, and so this code is run five times in succession.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_Calibration.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_Calibration.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following tables, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_Calibration/x/UserOptions.nl, where x represents one of the calibration run stages.

Variable	Type	Description	Default	Notes
CalibRunStage	Integer	Runs the calibration stage. Run serially 1 to 5.	1	1=convert UM to ABC forecast ensemble, 2=compute forecast perturbations, 3=determine the regression parameters, 4=do parameter transform, 5=calibrate spatial statistics.

The namelist variables are given here separately for each stage.

4.4.1 CalibRunStage = 1: Generating sample forecasts from UM data dumps (in the table below NYI=not yet implemented)

Variable for stage 1	Type	Description	Default	Notes
Nens	integer	Number of ensembles used for calibration.	50	0=Do not use ensembles to calibrate.
NEnsMems	integer	Number of ensemble members in each ensemble.	24	
EnsDirs(:)	string array	Names of directories containing ensembles.		Containing UM files Member001.nc, Member002.nc, ...
NNMC	integer	Number of NMC forecast pairs.	0	0=Do not use NMC method to calibrate. NYI.
NMCDirs(:)	string array	Names of directories containing NMC pairs.		NYI.
datadirABC_in	string	Directory containing sample ABC file		If ens mems already exist, need sample ABC file for longs and levs (see note below).
init_ABC_file	string	Sample ABC file		If ens mems already exist, need sample ABC file for longs and levs (see note below).
Nlats	integer	The number of latitude slices that are to be extracted from each ensemble member file.	1	NEnsMems $\times$ Nlats effective ensemble members per ensemble
latindex(:)	integer array	Specifies the latitude indices of the Nlats latitude slices.		
BoundSpread	double precision	No. of horizontal grid points to spread boundary discontinuity for periodic boundary conditions.	50.0	
A	double precision	Model parameter (pure gravity wave frequency).	0.02	s^{-1}
B	double precision	Model parameter (modulation of the divergent and advection terms).	0.005	
C	double precision	Model parameter (proportionality constant for the equation of state).	100000.0	$\mathrm{m}^2\mathrm{s}^{-2}$
f	double precision	Model parameter (Coriolis parameter).	0.0001	s^{-1}
dt	double precision	Model time step size.	4.0	s
runlength	double precision	Length of integration.	60.0	s

Variable for stage 1	Type	Description	Default	Notes
Adv_tracer	logical	Used to switch on/off tracer advection.	.FALSE.	Current implementation requires this to be .TRUE.
datadirABCfcs	string	Output directory of ABC forecast ensemble members.		Output files FC_Ens001_Item001.nc
datadirCVT	string	Output directory of blank CVT file.		
CVT_file	string	Name of blank CVT file to be output (in above directory).		
CVT% CVT_order	integer	Order of the transforms.	1	1=as original MetO, 2=reversed horiz/vert, 3=normal mode based (NYI).
CVT% CVT_param_opt_gb	integer	Geostrophic balance option for the transform.	1	1=analytical geo balance, 2=statistical balance NYI), 3=no geo balance.
CVT% CVT_param_opt_hb	integer	Hydrostatic balance option for the transform.	1	1=analytical hydro balance, 2=statistical balance (NYI), 3=no hydro balance.
CVT% CVT_param_opt_ab	integer	Anelastic balance option for the transform.	1	1=analytical anel balance, 2=no anel balance.
CVT% CVT_param_opt_reg	integer	Vertical regression option for the geostrophic balance field.	1	1=use vertical regression of the gb r, 2=no vertical regression.
CVT% CVT_vert_opt_sym	integer	Symmetry option for vertical covariances.	1	1=non-symmetric transform, 2=symmetric transform.
CVT% CVT_stddev_opt	integer	Standard deviation option.	2	1=stddev constant for each control variable, 2=level dependent only, 3=Longitude and level dependent.

Input and output files

- Inputs
  - Data from one or more suitable UM data files (EnsDir(i)/Member001.nc, etc.). For the specification of the UM data file, see Sect. 4.1.
  - If the ensemble members already exist (i.e. they have originated from an external source and so there is no need to specify a suitable UM file to generate them from), then this stage still needs to be run (to create the CVT file), but it should be run with $Nens = 0$, $NNMC = 0$, and $Nlats = 1$. Also need to specify a sample ABC file for longitudes and levels (these are output to the skeleton CVT file).
- Outputs
  - Effective ensemble members (datadirABCfcs/FC_Ens001_Item001.nc, etc.; the first 001 is the ensemble number, and the second 001 is the item number – e.g. for NEnsMems=2, and Nlats=3, for the first ensemble member, item=1,2,3, and for the second ensemble member, item=3,4,5). There are a total of Nens × NEnsMems × Nlats output forecast states in total. If $Nens = 0$, $NNMC = 0$, and $Nlats = 1$ (as specified in the previous bullet point), then the effective ensemble members are assumed to already exist and so no effective ensemble members are output.

- A skeleton control variable transform file (datadirCVT/CVT_file) is a netcdf file, which is blank apart from containing the values A, B, C, f, CVT_order, CVT_param_opt_gb, CVT_param_opt_hb, CVT_param_opt_ab, CVT_param_opt_reg, CVT_vert_opt_sym, and CVT_stddev_opt.

Graphics tools

- The python code *PlotEnsemblesABC.py* can be used to plot the full ensemble members.

4.4.2 CalibRunStage = 2: Compute perturbations from the forecast data of stage 1

Variable for stage 2	Type	Description	Default	Notes
Nens	integer	Number of ensembles used for calibration (as stage 1).	50	0=Do not use ensembles to calibrate.
NEnsMems	integer	Number of ensemble members in each ensemble (as stage 1).	24	
datadirABCfcs	string array	Directory containing ensemble or NMC forecasts (as output in stage 1).		Containing FC_Ens001_Item001.nc, ...
NNMC	integer	Number of NMC forecast pairs (as stage 1).	0	0=Do not use NMC method to calibrate. Not yet implemented.
Nlats	integer	The number of latitude slices (as stage 1).	1	NEnsMems × Nlats effective ensemble members for each ensemble
datadirABCperts	string	The name of the directory to output the ensemble means and perturbations.		The ensemble means output are MeanABC001.nc, ..., and the perturbations output are PertABC_Ens001_Item001.nc, ...
datadirCVT CVT_file	string string	Directory containing CVT file. Name of CVT file. Used to extract options (output to this file was done in stage 1).		

Input and output files

- Inputs
  - Forecast data from one or more forecasts from the ABC model (datadirABCfcs/FC_Ens001_Item001.nc, etc.). For the specification of the ABC data file, see Sect. 4.1.
  - Model specification data (A, B, C, f) output to the CVT file during stage 1 (datadirCVT/CVT_file).
- Outputs
  - Ensemble means for each ensemble (datadirABCperts/MeanABC001.nc, etc.).
  - Ensemble perturbations for each ensemble, and each member (datadirABCperts/PertABC_Ens001_Item001 etc.; the first 001 is the ensemble number, and the second 001 is the item number). The output perturbation file labelling corresponds to the input forecast file labelling.

Graphics tools

- The python code *PlotEnsemblesABC.py* can be used to plot the ensemble perturbations.

4.4.3 CalibRunStage = 3: Compute vertical regression matrix for geostrophic balanced mass fields

Variable for stage 3	Type	Description	Default	Notes
Nens	integer	Number of ensembles used for calibration (as stage 1).	50	0=Do not use ensembles to calibrate.
NEnsMems	integer	Number of ensemble members in each ensemble (as stage 1).	24	
datadirABCperts	string array	Directory containing ensemble perturbations (as output in stage 2).		Containing PertABC_Ens_001_Item_001.nc ...
NNMC	integer	Number of NMC forecast pairs (as stage 1).	0	0=Do not use NMC method to calibrate. Not yet implemented.
Nlats	integer	The number of latitude slices (as stage 1).	1	NEnsMems $\times$ Nlats effective ensemble members for each ensemble
datadirCVT	string	Directory containing CVT file.		
CVT_file	string	Name of CVT file. Used to extract options, and to output vertical covariance matrix.		
datadirRegression	string	Directory to containing sample files from this run.		r for the first ensemble/ensemble member, and its balanced version.

Input and output files

- Inputs
  - ABC model perturbations as output from stage 2 (datadirABCperts/PertABC_Ens_001_Item_001.nc, etc.).
  - Model specification data (A, B, C, f), and covariance options output to the CVT file during stage 1 (datadirCVT/CVT_file).
- Outputs (if CVT options allow)
  - Regression matrices to the covariance file (datadirCVT/CVT_file.nc).
  - Sample fields for the first ensemble, and first ensemble member, namely r and the geostrophically balanced version of r (datadirRegression/r_001.nc, datadirRegression/psi_001.nc, and datadirRegression/rbal_001.nc). The output perturbation file labelling corresponds to the input forecast file labelling.

Graphics tools

- The python code *PlotRegressionMatrices.py* can be used to plot the matrices output by this run stage.

4.4.4 CalibRunStage = 4: Perform parameter transform (convert model perturbations to parameters)

Variable for stage 4	Type	Description	Default	Notes
Nens	integer	Number of ensembles used for calibration (as stage 1).	50	0=Do not use ensembles to calibrate.
NEnsMems	integer	Number of ensemble members in each ensemble (as stage 1).	24	
datadirABCperts	string array	Directory containing ensemble perturbations (as output in stage 2).		Containing PertABC_Ens_001_Item_001.nc ...
NNMC	integer	Number of NMC forecast pairs (as stage 1).	0	0=Do not use NMC method to calibrate. Not yet implemented.
Nlats	integer	The number of latitude slices (as stage 1).	1	NEnsMems × Nlats effective ensemble members for each ensemble
datadirCVT	string	Directory containing CVT file.		
CVT_file	string	Name of CVT file. Used to extract options, and to output vertical covariance matrix.		
datadirConParams	string	Directory to contain perturbations of control parameters (converted from perturbations of model variables).		PertParam_001_Item001.nc, etc.

Input and output files

- Inputs
  - ABC model perturbations as output from stage 2 (datadirABCperts/PertABC_Ens_001_Item_001.nc, etc.).
  - ABC mean states as output from stage 2 (datadirABCperts/MeanABC001.nc, etc.).
  - Model specification data (A, B, C, f), covariance options output to the CVT file during stage 1, and regression data computed from stage 3 (datadirCVT/CVT_file).
- Outputs
  - Ensemble of pertubations of control parameters (datadirConParams/PertParam_001_Item001.nc, etc).
  - Sample fields for the first ensemble, and first ensemble member, namely balanced b (datadirConParams/b_b.nc), balanced r pre-regression step (datadirConParams/r_b_preregress.nc), and balanced r post-regression step (datadirConParams/r_b_postregress.nc).

Graphics tools

- The python code *PlotParameterEnsembles.py* can be used to plot the parameter perturbations.

- The python code *Plot_rp_pre+post_regress.py* can be used to plot the example pre- and post-regression balanced r.

4.4.5 CalibRunStage = 5: Perform calibration of each parameter (compute parameter standard deviations, and vertical and horizontal control variable transforms)

Variable for stage 5	Type	Description	Default	Notes
Nens	integer	Number of ensembles used for calibration (as stage 1).	50	0=Do not use ensembles to calibrate.
NEnsMems	integer	Number of ensemble members in each ensemble (as stage 1).	24	
datadirABCperts	string array	Directory containing ensemble perturbations (as output in stage 2).		Read PertABC_Ens_001_Item_001.nc dimension information.
datadirConParams	string	Directory containing perturbations of control parameters.		Containing Pert-Param_001_Item001.nc, etc.
NNMC	integer	Number of NMC forecast pairs (as stage 1).	0	0=Do not use NMC method to calibrate. Not yet implemented.
Nlats	integer	The number of latitude slices (as stage 1).	1	NEnsMems × Nlats effective ensemble members for each ensemble
VertSmoothPoints	integer	Number of points in vertical (in each direction) to average for standard dev.	0	0=No vertical smoothing of standard dev.
HorizSmoothPoints	integer	Number of points in horizontal (in each direction) to average for standard dev.	0	0=No horizontal smoothing of standard dev.
ForceCor	logical	To adjust statistics so that horiz. and vert. transforms imply correlations.	.TRUE.	
datadirCVT	string	Directory containing CVT file.		
CVT_file	string	Name of CVT file. Used to extract options.		

Input and output files

- Inputs
  - Parameter perturbations as output from stage 4 (datadirConParams/PertParam_001_Item_001.nc, etc.).
  - Sample ABC model perturbation for reading in dimension data (datadirABCperts/PertABC_Ens_001_Item
  - Covariance options output to the CVT file during stage 1 (datadirCVT/CVT_file).
- Outputs
  - A complete CVT definition file for the specification made in stage 1 (datadirCVT/CVT_file).

Graphics tools

- The python code *PlotCVT.py* can be used to inspect the contents of the output CVT file.

4.5 Master_TestSuite

Tests various aspects of the components of the data assimilation system (adjoint and inverse tests of the control variable transforms, and linearization tests of the ABC model and observation operators). The linear ABC model is not yet implemented. This section also documents Master_GradTest which performs a gradient test.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_TestSuite.out
make Master_GradTest.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_TestSuite.out
```

or

```
$ABC_SRC/Master_GradTest.out
```

depending on which tests to perform.

Namelist variables for Master_TestSuite

The namelist variables for Master_TestSuite are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_TestSuite/UserOptions.nl.

Variable	Type	Description	Default	Notes
datadirABCfcs	string	Directory containing an ABC model state.		
LS_file	string	Name of a test ABC model state.		
datadirABCperts	string	Directory containing an ABC model perturbation.		RunInvTests=.TRUE.
Pert_file	string	Name of a test ABC perturbation file.		RunInvTests=.TRUE.
datadirCVT	string	Directory containing a CVT file.		RunAdjTests_CVT=.TRUE. or RunInvTests=.TRUE.
CVT_file	string	Name of a CVT file (as found from Master_Calibration).		RunAdjTests_CVT=.TRUE. or RunInvTests=.TRUE.
datadir_Obs	string	Directory containing observations.		RunAdjTests_obs=.TRUE.
Obs_file	string	Observation file (in above directory).		RunAdjTests_obs=.TRUE.
datadirTestDA	string	Directory to output textual and field results of test suite.		
diagnostics_file	string	File to output textual diagnostics (in above directory).		

RunAdjTests_CVT	logical	Set to perform adjoint tests on operators used in CVT	.FALSE.
RunAdjTests_obs	logical	Set to perform adjoint tests on observation operators	.FALSE.
RunInvTests	logical	Set to perform inverse tests	.FALSE.

Input and output files for Master_TestSuite

- Inputs
  - A complete CVT definition file after running the complete calibration suite (stages 1 to 5) (datadirCVT/CVT_file), needed when RunAdjTests_CVT=.TRUE. or RunInvTests=.TRUE.
  - An example ABC forecast file to act as a linearisation state for the parameter transform and observation operator (datadirABCfcs/LS_file).
  - An example ABC perturbations perturbation file (datadirABCperts/Pert_file), needed when RunInvTests=.TRUE.
  - An observations file for doing adjoint test of observation operator (datadir_Obs/Obs_file), needed when RunAdjTests_obs=.TRUE.
- Outputs
  - The text file of diagnostic results datadirTestDA/diagnostics_file shows output of the adjoint and/or inverse tests.
  - Fields associated with the inverse tests (output when RunInvTests=.TRUE.). The files output are datadirABCperts/uv2psi.nc, datadirABCperts/uv2chi.nc, datadirABCperts/psichi2u.nc, datadirABCperts/psichi2v.nc, datadirABCperts/fft_test.nc, and datadirABCperts/UpInvTest.nc. See below for descriptions of these files.
  - The observations file datadirTestDA/Obs_processed.dat (when RunAdjTests_obs=.TRUE.). This is a version of the observations file which has been processed (to contain, e.g. model observations, innovations, etc – see the observation file format in Sect. 4.8.

Graphics tools for Master_TestSuite

There are currently no special utilities to analyse the output from this program. The output file datadirTestDA/diagnostics_file contains some textual diagnostics.

Notes on the tests in Master_TestSuite

Many of the mathematical transforms used in the DA system require an adjoint version, which appear in the expressions computing the gradient of the cost function, needed for the minimization algorithm. Even though it may not be explicitly coded as such, a linear transform is equivalent to the action of a matrix on an input vector to give an output vector. An adjoint version of a transform is essentially the equivalent to the transpose of this matrix (or in the case of complex values the combined adjoint and complex conjugate of this). To distinguish the transform from its adjoint, the transform itself is often called the forward transform. As with the forward transform, the adjoint is not necessarily written in an explicit matrix operation, but instead comprises a set of linear code. Mistakes often creep into the translation of code from the forward version to the adjoint version. An adjoint test is a reliable way of testing that this procedure has been done correctly.

In its basic form the adjoint test takes an arbitrary vector $\mathbf{v}$ (e.g. a vector of random numbers), and checks that the following holds:

$$\begin{aligned} (\mathbf{Av})^{\dagger}\,\mathbf{Av} &\stackrel{?}{=} \left(\mathbf{A}^{\dagger}\mathbf{Av}\right)^{\dagger}\mathbf{v}, \\ \text{or } \langle \mathbf{Av}, \mathbf{Av} \rangle_{\mathbf{I}} &\stackrel{?}{=} \left\langle \mathbf{A}^{\dagger}\mathbf{Av}, \mathbf{v} \right\rangle_{\mathbf{I}}, \end{aligned}$$

where $\mathbf{A}$ is the forward operator and $\mathbf{A}^{\dagger}$ is the adjoint operator. The above must be satisfied to machine precision. The above basic form assumes that the inner product metric is the identity matrix, i.e. that an inner product between two matrices $\mathbf{u}$ and $\mathbf{v}$ is $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{I}} = \mathbf{u}^{\dagger}\mathbf{I}\mathbf{v} = \mathbf{u}^{\dagger}\mathbf{v}$.

The following CVT operators are coded for adjoint tests in the current test suite:

- Boundaries (code to swap halos to satisfy boundary conditions in model fields)
- Boundaries_CV (as above, but for control fields)
- LinearBal_r (computation of the linearly balanced r field from the streamfunction field)
- Anbalw (computation of the anelasticlly balanced w field from the u field)
- Helmholtz (computation of u and v from the streamfunction and velocity potential)
- INT_HF (function to do vertical interpolation from half to full levels)
- INT_FH (as above, but from full to half levels)
- HydroBal_b (computation of the hydrostatically balanced b field from r)
- U_p (the parameter transform)
- U_v (the vertical transform)
- U_stddev (the standard deviation transform)
- fft_real2spec (FFT from real to spectral spaces)
- fft_spec2real (FFT from spectral to real spaces)
- U_h (the horizontal transform)
- U_trans (the complete control variable transform, e.g. if using the traditional formulation this includes the horizontal, vertical, parameter, and standard deviation transforms).

The following observation operators are coded for adjoint tests in the current test suite:

- Interpolate1D
- Interpolate3D
- ModelObservations_linear

In order to do an adjoint test of the ModelObservations_linear operator, an observations file is required. The time dimension test is not currently implemented (only 3DFGAT is implemented). An observations file may be produced by running stages 1 and 2 of 4.8.

The inverse tests $\mathbf{A}^{-1}\mathbf{A}$ is performed on the following operators:

- InverseSymMat (where $\mathbf{A}$ is the auto-covariance balanced r, as found in the CVT file, and the outputs of $\mathbf{A}^{-1}\mathbf{A}$ [and also $\mathbf{A}\mathbf{A}^{-1}$] are sent to the diagnostics file, datadirTestDA/diagnostics_file).
- The u and v fields from datadirTestDA/Pert_file are transformed to ψ and χ (routine Helmholtz_inv) and output as datadirTestDA/uv2psi.nc and datadirTestDA/uv2chi.nc. These fields are then transformed back to u and v (routine Helmholtz) and output as datadirTestDA/psichi2u.nc and datadirTestDA/psichi2v.nc.
- The r field from datadirTestDA/Pert_file is transformed to spectral space (routine fft_real2spec), and then back to real space (routine fft_spec2real). The result is output to datadirTestDA/fft_test.nc.
- The u, v, w, r, b, and tracer fields from datadirTestDA/Pert_file.nc are transformed to control variables (routine U_p_inv) and then transformed back to the original fields (routine U_p). The result is output to datadirTestDA/UpInvTest.nc.

Namelist variables for Master_GradTest

The namelist variables for Master_GradTest are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files and red entries describe the output directories/-files. An example namelist is given in the file ABCvn1.4/examples/Master_GradTest/UserOptions.nl.

Variable	Type	Description	Default	Notes
Vartype	integer	The type of data assimilation	3	3=3DVar, 35=3D-FGAT, 4=4DVar
$t0$	integer	Time of start of this DA cycle	100000.0	s
datadir_Bg	string	Directory containing an ABC model background.		
Bg_file	string	ABC model background filename.		
datadirCVT	string	Directory containing a CVT file.		
CVT_file	string	Name of a CVT file (as found from Master_Calibration).		
datadir_Obs	string	Directory containing observations.		
Obs_file	string	Observation file (in above directory).		
datadirTestDA	string	Directory to output textual and field results of test suite.		
diagnostics_file	string	File to output textual diagnostics (in above directory).		

Input and output files for Master_TestSuite

- Inputs
  - The input file datadir_Bg/Bg_file is a suitable background file. The latest time present in this file is read-in as the background state.
  - The input file datadir_CVT/CVT_file contains information defining the control variable transform. This file contains many of the other things needed to define the way that the data assimilation is done (see Section 4.4).
  - The input file datadir_Obs/Obs_file contains the observations. It also contains information that defines the length of the time window (it contains the number of model timesteps, the number of data assimilation timesteps, and the step lengths dt and dt_da. It has the same format as that specified in Sect. 4.8 (*Generate_mode=2*). Note that not all elements need to be specified to run the assimilation – see notes accompanying the format specification detailing which are needed by the assimilation (unused elements may be set to zero as dummy values). A version of the observations file is output by the assimilation in which these elements gain meaningful values, which are useful for diagnostics and monitoring – see output file descriptions below.
- Outputs
  - The text file of diagnostic results datadirTestDA/diagnostics_file shows output of the gradient test (see below for what the gradient test is, and what quantities are output.

Graphics tools for Master_GradTest

There are currently no special utilities to analyse the output from this program. The output file datadirTestDA/diagnostics_file contains some textual diagnostics.

Notes on the tests in Master_TestSuite

The gradient test checks that the gradient of the cost function has been evaluated consistently. It essentially performs a sequence of progressively more accurate finite difference calculations of the gradient and compares those to the analytical gradient itself.

Consider the following Taylor series expansion of the cost function as a function of the control variable

$$J(\boldsymbol{\chi} + \alpha \mathbf{h}) = J(\boldsymbol{\chi}) + \alpha \mathbf{h}^{\mathrm{T}} \nabla_{\chi} J + \mathcal{O}(\alpha^2).$$

Rearranging this to give a quantity Φ:

$$\Phi = \frac{J(\boldsymbol{\chi} + \alpha \mathbf{h}) - J(\boldsymbol{\chi})}{\alpha \mathbf{h}^{\mathrm{T}} \nabla_{\chi} J} = 1 + \mathcal{O}(\alpha).$$

Thus, by choosing α to have progressively smaller values (by orders of magnitude), and evaluating and plotting Φ, it can be determined if the gradient has been computed successfully. In the code, $\mathbf{h}$ has been chosen to be the unit vector $\mathbf{h} = \nabla_{\chi} J / |\nabla_{\chi} J|$. The output file datadirTestDA/diagnostics_file contains lines of α, Φ, and $\Phi - 1$.

4.6 Master_ImpliedCov

Computes a selection of implied covariances (selected columns of $\mathbf{U}\mathbf{U}^{\mathrm{T}}$*) between the model variables.*

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_ImpliedCov.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_ImpliedCov.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_ImpliedCov/UserOptions.nl.

Variable	Type	Description	Default	Notes
datadirABCfcs	string	Directory containing an ABC model state.		
LS_file	string	Name of a test ABC model state.		
datadirCVT	string	Directory containing a CVT file.		
CVT_file	string	Name of a CVT file (as found from Master_Calibration).		
datadirImpliedCov	string	Directory to output fields that represent implied covariances.		
ImplCov_npoints	Integer	Number of source points to compute implied covariances with respect to.	0	
longindex(:)	integer array	Specifies the longitude indices of the ImplCov_npoints source points		
levindex(:)	integer array	Specifies the level indices of the ImplCov_npoints source points		

Input and output files

- Inputs
  - A complete CVT definition file after running the complete calibration suite (stages 1 to 5) (datadirCVT/CVT_file).
  - An example ABC forecast file to act as a linearisation state for the parameter transform (datadirABCfcs/LS_file).
- Outputs
  - Fields giving the implied covariances, $\mathbf{U}\mathbf{U}^{\mathrm{T}}$, associated with the source points. The files output are datadirImpliedCov/Point_001_deltau.nc, datadirImpliedCov/Point_001_deltav.nc,

datadirImpliedCov/Point_001_deltaw.nc, datadirImpliedCov/Point_001_deltar.nc, datadirImpliedCov/Point_001_deltab.nc, and datadirImpliedCov/Point_001_deltatracer.nc. where 001 is the point number. See below for descriptions of these files.

Graphics tools

- The python code *PlotCovs.py* can be used to inspect the contents of the output files mentioned above. The same code is used to plot raw covariances (i.e. covariances found from the data used to calibrate the control variable transform, computed from Master_RawCov).

Notes on the implied covariances

The operator $\mathbf{U}\mathbf{U}^{\mathrm{T}}$ is the background error covariance matrix ($\mathbf{B}_{\mathrm{imp}}$) that is implied by the control variable transform. When $\mathbf{U}\mathbf{U}^{\mathrm{T}}$ operates on a vector $\mathbf{v}$ of zeros, apart from one particular unit element, the output $\mathbf{u} = \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{v}$ is the column of $\mathbf{B}_{\mathrm{imp}}$ associated with where the unit element is located. For instance, if the unit element is placed in the field r near the centre of the domain, then $\mathbf{u}$ is the set of fields that comprise the column of $\mathbf{B}_{\mathrm{imp}}$ associated with that single r point. If this spatial point's position is prescibed with longindex(1), levindex(1), then the output file is datadirImpliedCov/Point_001_deltar.nc. The code systematically puts the unit point in each of the six fields in turn to give the six output files specified above.

4.7 Master_RawCov

Computes a selection of raw covariances between the model variables.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_RawCov.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_RawCov.out
```

Note that the input data needed to run this program is generated by stage

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_RawCov/UserOptions.nl.

Variable	Type	Description	Default	Notes
datadirABCperts	string	Directory containing ABC model state perturbations.		As output from run stage 2 of the calibration, 4.4.
datadirRawCov	string	Directory to output fields that represent raw covariances.		
Nens	integer	Number of ensembles used.	50	0=Do not use ensembles to calibrate. Use as in run stage 2 of the calibration, 4.4.
NensMems	integer	Number of ensemble members in each ensemble.	24	Use as in run stage 2 of the calibration, 4.4.
NNMC	integer	Number of NMC forecast pairs.	0	0=Do not use NMC method to calibrate. Not yet implemented.
Nlats	integer	The number of latitude slices.	1	Use as in run stage 2 of the calibration, 4.4.
ImplCov_npoints	integer	Number of source points to compute implied covariances with respect to.	0	
longindex(:)	integer array	Specifies the longitude indices of the ImplCov_npoints source points		
levindex(:)	integer array	Specifies the level indices of the ImplCov_npoints source points		

Input and output files

- Inputs
  - ABC model perturbations (as output from stage 2 – datadirABCperts/PertABC_Ens_001_Item_001.nc, etc.).
- Outputs
  - Fields giving the raw covariances, associated with the source points. The files output are datadirRawCov/Point_001_deltau.nc, datadirRawCov/Point_001_deltav.nc, datadirRawCov/Point_001_deltaw.nc, datadirRawCov/Point_001_deltar.nc, datadirRawCov/Point_001_deltab.nc, and datadirRawCov/Point_001_deltatracer.nc. where 001 is the point number.

Graphics tools

- The python code *PlotCovs.py* can be used to inspect the contents of the output files mentioned above. The same code is used to plot implied covariances (i.e. covariances found from $\mathbf{U}\mathbf{U}^{\mathrm{T}}$, computed from Master_ImpliedCov).

4.8 Master_MakeBgObs

Generates a set of synthetic observations for assimilating, and a synthetic background state.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_MakeBgObs.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_MakeBgObs.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_MakeBgObs/x/UserOptions.nl, where x represents one of the run stages for this routine.

Variable	Type	Description	Default	Notes
Generate_mode	Integer	Specifies what data this routine should produce (1-3).	1	1=generate a file that is used to specify obs times, positions, and types, etc., 2=generate obs consistent with some truth, 3=Generate a background state consistent with some truth.

The namelist variables are given here separately for each stage.

4.8.1 Generate_mode=1: Generate a file that is used to specify observation times, positions, and types, etc.

Variable	Type	Description	Default	Notes
datadir_ObsSpec	string	Directory to output observation specification file.		
ObsSpec_file	string	Filename of observation specification file (output to the above directory).		Format see below.
ObsSpec%year0	integer	Specification of $t = 0$ (year).	2000	
ObsSpec%month0	integer	(month, 1-12).	1	
ObsSpec%day0	integer	(day, 1-31).	1	
ObsSpec%hour0	integer	(hour, 0-23).	0	
ObsSpec%min0	integer	(minute, 0-59).	0	
ObsSpec%sec0	integer	(second, 0-59).	0	
ObsSpec%NumBatches	integer	Number of observation batches.	0	Each batch represents a particular obs type and time. The obs batch is distributed over space as specified below.
ObsSpec%batch(:)	integer array	Batch number.	0	Not currently used, but could be used to group batches together (with a common batch number) for possible later developments with correlated obs.
ObsSpec%seconds(1)	integer array	Abolsute time of this observation batch.	0	seconds since $t = 0$
ObsSpec%ob_of_what(:)	integer array	What is to be observed.	0	1=u, 2=v, 3=w, 4=r, 5=b, 6=tracer, 7=horizontal wind speed, 8=total wind speed.
ObsSpec%NumObs_long(:)	integer array	Number of observations in the longitude direction for this batch	0	
ObsSpec%NumObs_height(:)	integer array	Number of observations in the height direction for this batch	0	
ObsSpec%long_min(:)	double precision array	West-most extent of observation grid for this batch.	0.0	
ObsSpec%long_max(:)	double precision array	East-most extent of observation grid for this batch.	0.0	
ObsSpec%height_min(:)	double precision array	Lowest position of observation grid for this batch.	0.0	

ObsSpec%height_max(:)	double precision array	Highest position of observation grid for this batch.	0.0
ObsSpec%stddev(:)	double precision array	Error standard deviation of this batch.	0.0

Input and output files

- Outputs
  - The observations to be made are specified in the file datadir_ObsSpec/ObsSpec_file, which has the format mentioned below. It is read by the code running with Generate_mode=2 to generate the actual observations. This file can be created separately, but the Generate_mode=1 mode has been provided as a convenient means of created this file. The file can also be editied if required before being read by the Generate_mode=2 mode (e.g. to modify an observation's error standard deviations, etc., etc.

Graphics tools

There are currently no special utitities to analyse the output from this program.

Notes on the output file

The format of the output file datadir_ObsSpec/ObsSpec_file is illustrated with the following example.

```
Observation specification file for ABC model
Format version   :    1
-----------------------------------------------
Ref year         :  2010
Ref month        :    1
Ref day          :    1
Ref hour         :    0
Ref minute       :    0
Ref second       :    0
-----------------------------------------------
Observation No   :            1
Batch ID         :        1
Time of obs (s)  :        300
Longitude (deg)  :     10000.000
Height (m)       :      1000.000
Observation of   :    5
Err stddev       :          0.001
-----------------------------------------------
Observation No   :            2
Batch ID         :        1
Time of obs (s)  :        300
Longitude (deg)  :     10000.000
Height (m)       :      4666.667
Observation of   :    5
Err stddev       :          0.001
-----------------------------------------------
```

...

The "observation of" refers to the quantity observed (see the key for ObsSpec%ob_of_what(:) in the table above).

4.8.2 Generate_mode=2: Generate observations consistent with some truth

Variable	Type	Description	Default	Notes
datadir_ObsSpec	string	Directory containing observation specification file.		
ObsSpec_file	string	Observation specification file (in above directory).		Format see above.
datadirABC_in	string	Directory containing the initial truth state.		End time present is used as init conds.
init_ABC_file	string	ABC model dump containing the truth (in above directory).		
dt	double precision	Time step of the model		
dt_da	double precision	Time step of the data assimliation system.	60.0	Constraint: $dt\_da = ndt$, $n = 1, 2, \ldots$
$t0$	integer	Time of start dump.	0	seconds
Runlength	double precision	Length of DA cycle	60.0	Carried through to the DA via the Obs_file (below)
datadir_Obs	string	Directory containing observational data that can be later assimilated, and truth run.		
Obs_file	string	Observation file (in above directory).		Format see below.
output_ABC_file	string	Output truth trajectory file (in above directory).		
A	double precision	Model parameter (pure gravity wave frequency).	0.02	s^{-1}
B	double precision	Model parameter (modulation of the divergent and advection terms).	0.005	
C	double precision	Model parameter (proportionality constant for the equation of state).	100000.0	m^2s^{-2}
f	double precision	Model parameter (Coriolis parameter).	0.0001	s^{-1}
random_seed	integer	To seed the random number generator	0	

Input and output files

- Inputs
  - Suitable truth input file (datadirABC_in/init_ABC_file). The latest time present in this file is used as the initial conditions for the model truth run.
  - The observations to be made are specified in the file datadir_ObsSpec/ObsSpec_file, which has the format mentioned above.
- Outputs
  - The synthetic observations themselves are output to the file datadir_Obs/Obs_file, which has the format mentioned below.
  - The truth trajectory, output at every data assimilation time step, datadir_Obs/output_ABC_file.

Graphics tools

There are currently no special utitities to analyse the output from this program.

Notes on the output file

The format of the output file datadir_Obs/Obs_file is illustrated with the following example (comprising two observations).

```
Observation file for ABC model
Format version   :   1
maxtime (s)      :     2400
Model ts (s)     :   0.400E+01
No model ts      :     600
DA ts (s)        :   0.600E+02
No DA ts         :      40
--------------------------------------------------
--------------------------------------------------
Observation No   :           1
Batch ID         :       1
Time of obs (s)  :     300
Longitude (deg)  :   10000.000
Height (m)       :    1000.000
xbox_lower       :           6
xbox_lower_ws    :           0
zbox_lower       :           4
zbox_lower_ws    :           0
tstep_lower      :           5
Observation of   :   1
                 : u
y_true_known     : T
y_true           :  -0.124E+00
y                :   0.152E+01
stddev           :   0.500E+00
y_ref            :   0.000E+00
d                :   0.000E+00
deltay_m         :   0.000E+00
ymhx             :   0.000E+00
deltay_m_hat     :   0.000E+00
--------------------------------------------------
Observation No   :          2
Batch ID         :       1
Time of obs (s)  :    1500
Longitude (deg)  :    1000.000
Height (m)       :     500.000
xbox_lower       :           0
xbox_lower_ws    :           0
zbox_lower       :           1
zbox_lower_ws    :           0
tstep_lower      :          25
Observation of   :   5
                 : b
y_true_known     : T
y_true           :  -0.578E-03
y                :  -0.578E-03
stddev           :   0.000E+00
```

```
y_ref              :    0.000E+00
d                  :    0.000E+00
deltay_m           :    0.000E+00
ymhx               :    0.000E+00
deltay_m_hat       :    0.000E+00
-------------------------------------------
...
```

The "observation of" refers to the quantity observed (see the key for ObsSpec%ob_of_what(:) in the table for the Generate_mode=1 mode). The quantity that this corresponds to is also given in the line below this ("observation of" 1 corresponds to quantity u, and 5 corresponds to b in the example), although this textual description is not used by the software, only the "observation of" code.

The xbox_lower and zbox_lower are the grid box indices of the first point on the ABC model grid that is immediately below the observation position. (For example if the height of the observation is 55m, and the vertical grid has level heights 0, 10, 20, 30, 40, 50, 60, ..., then zbox_lower is 6 (the 6th level height is immediately below the 55m height). These indices are stored along with the observation for efficiency (since they are already computed when producing the observation file). If the observation file was produced by some other means, or if the user wishes to manually edit the observation positions, then these indixes may be set to zero – this will force the software to (re)compute these values when the observations are assimilated. Note that some observations need to have indices for more than one quantitiy (e.g. vertical wind speed, which is a function of a combination of u, v, and w values; since u and v are on different horizontal grid points, and u/v, and w are on different vertical levels (see Fig. 1), then two versions of the lower indices are required ([xbox_lower, zbox_lower] for u, [xbox_lower_ws, zbox_lower] for v, and [xbox_lower_ws, zbos_lower_ws] for w. The extra indices xbox_lower_ws and zbox_lower_ws are zero if they are not needed.

There are some elements in the observation file that are not needed by the assimilation, and may be set to arbitrary values. These are y_true_known (T if the 'true' observation is known), y_true (the 'true' value of the observation), y_ref (the (potentially non-linear) model observation computed at the reference state), d (the difference between the observation, y, and y_ref), deltay_m (the perturbation to the modelled observation, computed using the linear operator on a perturbation state), ymhx (the difference between the observation, y, and the modelled observation, $y\_ref + deltay\_m$), and deltay_m_hat (the quantity $\partial J_{\mathrm{O}}/\partial deltay\_m$). These are included in the above file format as a version of the observation file may be output during or post assimilation for diagnostic purposes, where this information is known. This means that a single observation file format is used whether input or output to the data assimilation software.

Variable	Type	Description	Default	Notes
datadirABC_in	string	Directory containing the 'truth'.		$\mathbf{x}^{\mathrm{t}}$
init_ABC_file	string	ABC model dump containing the 'truth' (in above directory).		End time present is used as init conds.
datadirCVT	string	Directory containing a CVT file.		
CVT_file	string	Name of a CVT file (as found from Master_Calibration).		
Bg_inflation	double precision	Inflation factor for background generation	1.0	
datadir_Bg	string	Directory containing the background data.		
Pert_file	string	Background error selected (in above directory).		$\delta\mathbf{x}^{\mathrm{b}}$
Bg_file	string	Background state (in above directory).		$\mathbf{x}^{\mathrm{b}} = \mathbf{x}^{\mathrm{t}} + \delta\mathbf{x}^{\mathrm{b}}$
random_seed	integer	To seed the random number generator	0	

Generate_mode=3: Generate a background state consistent with some truth

Input and output files

- Inputs
  - Suitable truth input file (datadirABC_in/init_ABC_file). The latest time present in this file is used as the state that is perturbed by background error.
  - A complete CVT definition file after running the complete calibration suite (stages 1 to 5) (datadirCVT/CVT_file).
- Outputs
  - A randomly chosen background error file (ABC model format, in file datadir_Bg/Pert_file) statistically consistent with the background error covariance as specified in the CVT file.
  - A background state (ABC model format, in file datadir_Bg/Bg_file) equal to the 'truth' read-in by this routine plus the above error.

Graphics tools

There are currently no special utitities to analyse the output from this program.

4.9 Master_Assimilate

Inputs a background state, observations, and a CVT to give an analysis. Future options to include ensemble and hybrid methods.

To compile

Go to directory containing source code ($ABC_SRC), and issue:

```
make Master_Assimilate.out
```

To run

Prepare namelist file UserOptions.nl, go to directory containing this file, and issue:

```
$ABC_SRC/Master_Assimilate.out
```

Namelist variables

The namelist variables are described in UserOptions.nl, and should be placed in the directory where the above run command is issued. The variables are described in the following table, where blue entries describe the input directories/files, red entries describe the output directories/files, and purple entries describe the input/output directories/files. An example namelist is given in the file ABCvn1.4/examples/Master_Assimilate/UserOptions.nl.

Variable	Type	Description	Default	Notes
Vartype	integer	The type of data assimilation	3	3=3DVar, 35=3D-FGAT, 4=4DVar
Hybrid_opt	integer	Type of hybrid (or if pure Var)	1	1 = standard B, 2 = pure EnVar, 3 = hybrid EnVar , 4 = reduced rank KF-type hybrid (2-4 currently not implemented)
datadir_Bg	string	Directory containing background state.	No default	
Bg_file	string	Background state (in above directory).	No default	Init_ABC_opt=1
datadirCVT	string	Directory containing a CVT file.	No default	
CVT_file	string	Name of a CVT file (in above directory).	No default	
datadir_Obs	string	Directory containing observational data	No default	
Obs_file	string	Observation file (in above directory).	No default	
$t0$	integer	Time of start of this DA cycle	100000.0	s
N_outerloops	integer	Number of outer loops	1	
N_innerloops_max	integer	Maximum number of inner loops	10	
crit_inner	double precision	Stopping criterion for inner loop	0.01	$[\nabla J]_i/[\nabla J]_0 <$ *crit_inner* for iteration i
datadirAnal	string	Data to contain the analysis		

anal_file	string	Analysis file (put in above directory)
analinc_file	string	Analysis increment file (put in above directory)
diagnostics_file	string	Diagnostics file (put in above directory)

Input and output files

- Inputs
  - The intput file datadir_Bg/Bg_file is a suitable background file. The latest time present in this file is read-in as the background state.
  - The input file datadir_CVT/CVT_file contains information defining the control variable transform. This file contains many of the other things needed to define the way that the data assimliation is done (see Section 4.4).
  - The input file datadir_Obs/Obs_file contains the observations. It also contains information that defines the length of the time window (it contains the number of model timesteps, the number of data assimilation timesteps, and the step lengths dt and dt_da. It has the same format as that specified in Sect. 4.8 (*Generate_mode=2*). Note that not all elements need to be specified to run the assimilation – see notes accompanying the format specification detailing which are needed by the assimilation (unused elements may be set to zero as dummy values). A version of the observations file is output by the assimilation in which these elements gain meaningful values, which are useful for diagnostics and monitoring – see output file descriptions below.
- Outputs
  - The analysis is output to datadirAnal/anal_file. This is the background plus the analysis increment.
  - The analysis increment is output to datadirAnal/analinc_file.
  - Diagnostics are output to datadirAnal/diagnostics_file. This contains values of the residuals, values of the cost function, and how different variables, like state energy, change with iteration.
  - Linearisation state trajectories output to datadirAnal/LS_Oloop001_Iloop000.nc (the first number, 001 here, is the outer loop number, and the second number, 000 here, is the inner loop number). The first file, LS_Oloop001_Iloop000.nc, corresponds to the background trajectory, and the last file, e.g. LS_Oloop00n_Iloop000.nc, corresponds to the analysis trajectory, where n is $N\_outerloops + 1$. Put in another way, LS_Oloop00i_Iloop000.nc is the LS trajectory at the start of the ith outer loop.
  - datadirAnal/Delta_Oloop001_Iloop001.nc contains the values of $\mathbf{H}_t^{\mathrm{T}}\mathbf{R}_t^{-1}\left[\mathbf{H}_t\mathbf{M}_{0\rightarrow t}\delta\mathbf{x} - \mathbf{d}(t)\right]$. This is output only for the first inner loop of the first outer loop.
  - datadirAnal/GradJo_Oloop001_Iloop001.nc contains the fields describing the gradient of J_{O}.
  - Versions of the observations file are output to datadirAnal/Obs_001_Iloop000.dat (the first number, 001 here, is the outer loop number, and the second number, 000 here, is the inner loop number). Each observation's time, location, value, and error standard deviation is identical to the respective values in the input file of observations, but other information is included. The extra information includes the model observation value at the particular reference value, the innovation (found with respect to the model reference), the innovation (found with respect to the perturbed model refefence), the linear perturbation to the model observation, and the gradient of the observation term in the cost function with respect to the model observation.

The meanings of the symbols in the output file are described in **(author?)** [1]. In particular, the model equations are given as Eqs. (15) of that reference and the grid positions are shown in Fig. 1 of that reference.

Graphics tools

- The python code *PlotAssimDiags.py* (and the required subroutine contained in *Routines4PlotAssimDiags.py*) can be used to show the following diagnostics from the assimilation run.
  - Background trajectory.
  - Analysis trajectory.
  - Analysis increment at $t = 0$.
  - $\nabla_x J_{\mathrm{O}}$ trajectory.
  - Truth trajectory.
  - Background error trajectory.
  - Analysis error trajectory.
  - Cost function with iteration.
  - Energy with iteration.
  - $|\nabla_\chi J|$ with iteration.
  - Imbalance with iteration.
  - Histograms of O-B, O-A, O-T, B-T, A-T (O=observations, B=background, A=analysis, T=truth).

---

5 Data assimilation/forecast cycling

It is possible to cycle the data assimilation automatically with a suitable script. The commented bash script *DAcycle001.sh* is available (inside the script directory) for this purpose. The script assumes that the following have been prepared beforehand:

- CVT file (see Sect. (4.4) to generate this).
- Truth file (a valid ABC model state).
- An observation network specification file (see Sect. (4.8) to generate this). The way that the above script is written, this observation network is repeated every cycle, although the user may modify the script to allow the observation networks to change.
- Other variables required are documented in the script.

Here are some notes

- The true state is propagated from one cycle to the next and observations are simulated from it.
- The background state is a forecast from the previous analysis (except at the start of the cycling where it is a random perturbation from the truth).
- There is an option in the script to turn-on automatic plotting of the data assimilation results for each cycle using the python codes mentioned beforehand.
- When the cycling is finished the python code *PlotMultiCycleErrors.py* can be run, which shows assimilation diagnostics for the multiple cycles together.
- After they have run, two cycling experiments can be compared by running the python code *Plot_MultiCycle_DiffExps.py*. This will allow the errors of the two cycles to be compared. There are two sets of differences for each field:
  - The first set is the difference between the *mean errors* in the two experiments.
  - The second set is the difference between the *root-mean-squared-errors* in the two experiments.

6 Ensemble generation

It is possible to use the data assimilation cycling to generate an ensemble of ABC model states from a single dump. The commented bash script *MakeEns001.sh* is available (inside the script directory) for this purpose. The script assumes that the following have been prepared beforehand:

- CVT file (see Sect. (4.4) to generate this).
- Truth file (a valid ABC model state).
- An observation network specification file (see Sect. (4.8) to generate this). The way that the above script is written, this observation network is repeated every cycle, although the user may modify the script to allow the observation networks to change.
- Other variables required are documented in the script.

To generate an ensemble member, the script adds random background error to the truth file, and passes it through a specified number of assimilation cycles (with random observations). The forecast from the last cycle is taken as a forecast ensemble member.

References

[1] Ruth Elizabeth Petrie, Ross Noel Bannister, and Michael John Priestley Cullen. The ABC model: a non-hydrostatic toy model for use in convective-scale data assimilation investigations. *Geoscientific Model Development*, 10(12):4419, 2017.