The 'potential vorticity'-based control vector project
User's guide for the new code
Ross Bannister, Feb 2007/Apr 2007/Oct 2007/Nov 2008

r.n.bannister@reading.ac.uk

http://www.met.rdg.ac.uk/~ross/DARC/PVcv/UserGuide.pdf

1. Project summary

The meaning of two control parameters ψ and ${}^{A}p$ (members of the V_p structure) have been changed. The standard and new meanings are summarised in table I.

	Standard meaning of control parameter	New meaning of control parameter
ψ	Total streamfunction	Balanced streamfunction
${}^{A}p$	Residual pressure	Unbalanced pressure

Table I: The standard and new (PV-based) control variables

The changes involve new control variable parameter transforms, $\mathbf{U}_p$, $\mathbf{U}_p^T$ and $\mathbf{T}_p$, which are based in the properties of potential vorticity (PV). Unlike the existing transform, the new $\mathbf{T}_p$ transform makes use of an iterative generalised conjugate residual (GCR) solver. As the transforms have been altered significantly, an associated set of covariance statistics have to be computed. The covariance statistics, together with the transforms define the **B**-matrix.

There are other sources of scientific and technical information on this project. The project has a web page [1], containing a brief introduction to the problem [2], scientific information about the main problem [3], the preconditioning aspects [4,5,6] and how to calculate the parameters to determine cut-off wavenumbers for the unbalanced pressure (needed for the transform namelist) [7]. A preliminary publication on the work also exists [8,9]. The work is an extension of [10] at the ECMWF.

2. Modified code

The modified code (at the time of writing yet to be merged and lodged) is found at the following location (on the linux system)

```
    /home/mm0200/frxb/MyVarMods/PV/r12816_PV_Oct08
```

(see trac ticket number 381). Details of all of the changes made to the standard code are found on the Met Office VAR system under this ticket number.

3. Compiling the code

An example compilation job is *vaqv.d* and the location of the executable is

```
    /home/nwp/fr/frxb/var/build/Ticket381/PVcompile_Sep08/bin
```

4. Computing the covariance statistics

The suite jobs that compute the covariance statistics for the new control variables run *VarProg_CovPFStats*, which performs a $\mathbf{T}_p$ transform, and *VarProg_CovAccStats* which collects error statistics over the set of forecast differences. The main output files are shown in table II.

File	Description
COVACC.1	Accumulated forecast error statistics
COVACC.2	Modified from *COVACC.1* (rotation of vertical modes?)
COVACC.3	Modified from *COVACC.2* (scaling?)

Table II: Output files from the suite

COVACC.3 is the covariance file that is input into *Var_AnalysePF*, which runs data assimilation, and *COVACC.1* is the covariance file that is input into the program that plots covariance diagnostics (see below).

There are a number of suite jobs that perform the covariance statistics as follows.

- *sdit.f* is for the standard (non-PV-based) transforms using the NMC method.
- *sdit.e* is for the PV-based transforms using the NMC method.
- selz.a and selz.b are for the standard transforms using the ensemble method (there are two jobs as the *VarProg_CovPFStats* and *VarProg_CovAccStats* parts are performed separately).
- selz.c and selz.d are for the PV-based transforms using the ensemble method (there are two jobs for the same reason as above).

These jobs have been set-up to be given a set of forecast differences which have to be copied to the appropriate directory by hand (as they stand reading and writing to mass has been switched off).

At present the preferred way of plotting diagnostics contained in these files is to use PV-Wave routines. VAR plotting jobs must first be run to generate appropriate input files for these routines, which are contained in *stats.html* files (example VAR plotting jobs are specified below). The PV-Wave routines have various plotting options

- Opt. 1: Observed stats for individual months or seasons - standard deviations and correlations, and plots of mean LS.
- Opt. 2: As option 1 but for standard deviations and correlations only.
- Opt. 5: Observed stats for combined seasons - spectra and lengthscales, etc.
- Opt. 6: Processed stats (presumably characteristics of the transforms themselves) - vertical modes, spectra, lengthscales, etc.
- Opt. 11: Implied stats for individual months or seasons - standard deviations and correlations.
- Opt. 15: More implied stats - lengthscales and spectra of vertical correlations.

5. Plotting diagnostics of the observed cov. statistics (options 1, 2 and 5 of PV-Wave code)

Plotting diagnostics for the observed covariance statistics is a two-stage process.

- Run a *VarProg_CovAccStats* print job with the *COVACC.1* as the input file. This produces a stats file (*stats.html*). Example jobs are *vbwi.c* and *vawi.p*.
- Statistical diagnostics derived from *COVACC.1* are now in a text-file format inside the *stats.html* file and are plotted using the PV-Wave plotting routines detailed below.

For ease of organising information, it is recommended that the output directory (containing the *stdout.html*, *stderr.html* and *stats.html* files) is called, e.g., *Something.Ob*.

For the PV-Wave plotting routines

If not done already, make a new directory and copy the **.pro* files from the following directory into it

```
/home/mm0200/frxb/MareksPVwavecode
```

For this section, options 1, 2 and 5 of the PV-Wave code are relevant (see above). Edit the file *var_cov_plots.pro* to set the input directory *inputdir* for options 1 and 2, and *inputdirC1* for option 5. Each directory should be that containing the *stats.html* file, e.g. *Something.Ob*. Set the output directory *outputdir* which will contain *.ps* files after the code has run.

On the linux box inside the directory containing the PV-Wave code, run PV-Wave (type `wave`). Compile the necessary routines (you need to run the following commands)

```
.run var_cov_plots
.run var_cov_run
```

Run one or more of the routines that generate postscript files, e.g.

Opt. 1: `var_cov_plots,1,'Feb',2,3,0,0`

Opt. 2: `var_cov_plots,2,'Feb',2,3,0,0`

Opt. 5: `var_cov_plots,5,'',2,3,0,0`

More information about the run modes, including the meaning of the parameters, is available in the *var_cov_plots.pro* file.

6. Plotting processed statistics - vertical modes, etc. (option 6 of PV-Wave code)

Plotting diagnostics for the processed statistics is a two-stage process.

- Run a *VarProg_CovAccStats* plotting job (stage 10 flag set) with the *COVACC.1* as the input file. This produces a stats file (*stats.html*), which is automatically copied to the linux system. An example job is *vbwi.o*.
- Statistical diagnostics derived from *COVACC.1* are now in a text-file format inside the *stats.html* file and can be plotted by using PV-Wave plotting routines (detailed below).

For ease of organising information, it is recommended that the output directory (containing the *stdout.html*, *stderr.html* and *stats.html* files) is called, e.g., *Something.Mo*.

For the PV-Wave plotting routines

If not done already, make a new directory and copy the **.pro* files from the following directory into it

```
/home/mm0200/frxb/MareksPVwavecode
```

For this section, option 6 of the PV-Wave code is relevant (see above). Edit the file *var_cov_plots.pro* to set the input directory *inputdirC2*. This directory should be that containing the *stats.html* file, e.g.

Something.Mo. Set the output directory *outputdir* which will contain *.ps* files after the code has run.

On the linux box inside the directory containing the PV-Wave code, run PV-Wave (type `wave`). Compile the necessary routines (you need to run the following commands)

```
.run var_cov_plots
.run var_cov_run
```

Run the routine that generate postscript files, e.g.

Opt. 6: `var_cov_plots,6,'',2,3,0,0`

More information about the run modes, including the meaning of the parameters, is available in the *var_cov_plots.pro* file.

7. Plotting diagnostics of the implied cov. statistics (option 11 of PV-Wave code)

Plotting diagnostics for the implied covariance statistics is a three-stage process.

- Follow the instructions to generate an observed statistics (see section 5) so that the *stats.html* file for that is produced in a directory *Something.Ob*.
- Run a *VarAnalysePF* job (but with a *TestCov.exe* executable) with the *COVACC.3* as the input file. This is run on a supercomputer and produces a stats file (*stats.html*). An example job is *vbwi.n* for implied temperature statistics (using the *VAR_TESTCOV_NL* variable *TestCovNL.T3*). It is possible to use other namelists for other variables.
- Statistical diagnostics derived from *COVACC.3* are now in a text-file format inside the *stats.html* file and can be plotted with PV-Wave plotting routines (detailed below).

It is important to the PV-Wave plotting routines that the output directory (containing the *stdout.html*, *stderr.html* and *stats.html* files) is called, e.g., *Something.T3* for the temperature statistics, etc.

For the PV-Wave plotting routines

If not done already, make a new directory and copy the **.pro* files from the following directory into it

```
/home/mm0200/frxb/MareksPVwavecode
```

For this section, option 11 of the PV-Wave code is relevant (see above). Edit the file *var_cov_plots.pro* to set various directories. Set the input directory *inputdir* to *Something.Ob* (see above), another input directory *inputdirTC* to, e.g., *Something.T3* and the output directory *outputdir* which will contain *.ps* files after the code has run. Within the output directory that has been set, create a directory called *Implied*.

On the linux box inside the directory containing the PV-Wave code, run PV-Wave (type `wave`). Compile the necessary routines (you need to run the following commands)

```
.run var_cov_plots
.run var_cov_run
```

Run the routine that generate postscript files, e.g.

Opt. 11: `var_cov_plots,11,'Feb',2,3,0,0`

More information about the run modes, including the meaning of the parameters, is available in the *var_cov_plots.pro* file.

8. Plotting diagnostics of the implied spectral statistics (option 15 of PV-Wave code)

Plotting diagnostics for the implied spectral covariance statistics is a three-stage process.

- Follow the instructions to generate an observed statistics (see section 5) so that the *stats.html* file for that is produced in a directory *Something.Ob*.
- Run a *VarAnalysePF* job (but with a *TestCov.exe* executable) with the *COVACC.3* as the input file. This is run on a supercomputer and produces a stats file (*stats.html*). Example jobs are *vbwi.m* for implied spectral statistics (using the *VAR_TESTCOV_NL* variable *TestCovNL.IS*).
- Statistical diagnostics derived from *COVACC.3* are now in a text-file format inside the *stats.html* file and can be plotted with PV-Wave plotting routines (detailed below).

It is important to the PV-Wave plotting routines that the output directory (containing the *stdout.html*, *stderr.html* and *stats.html* files) is called, e.g., *Something.IS* for the spectral statistics.

For the PV-Wave plotting routines

If not done already, make a new directory and copy the **.pro* files from the following directory into it

```
    /home/mm0200/frxb/MareksPVwavecode
```

For this section, option 15 of the PV-Wave code is relevant (see above). Edit the file *var_cov_plots.pro* to set various directories. Set the input directory *inputdir* to *Something.Ob* (see above), another input directory *inputdirIS* to, e.g., *Something.IS* and the output directory *outputdir* which will contain *.ps* files after the code has run. Within the output directory that has been set, create a directory called *Implied*.

On the linux box inside the directory containing the PV-Wave code, run PV-Wave (type `wave`). Compile the necessary routines (you need to run the following commands)

```
    .run var_cov_plots
    .run var_cov_run
```

Run the routine that generate postscript files, e.g.

Opt. 15: `var_cov_plots,15,'',2,3,0,0`

More information about the run modes, including the meaning of the parameters, is available in the *var_cov_plots.pro* file.

9. Running the new PV code

There are many new namelist variables associated with the new PV code. The list in table III shows their names and default values (they are for use in the *Transforms namelist part 1* of the varui). The most important is the first one shown in the table which switches the PV-based code on or off. If should be set to .true. to invoke the PV-based transforms.

Variable	Default	Description
`PV_Trans`	.false.	Set to .true. to use the new PV-based transforms, set to .false. for standard code.
`PV_UseAntiPV`	.false.	Set to .true. to fit unbalanced pressures to anti-PV, set to .false. to fit to unbalanced streamfunction.

`LSzmMeanOption`	0	0: zonal mean, 1: global mean
`LSzmIsoThermLevs`	0	Sets No. of levels from ground to have equal T in new LSzm structure (inside LSzm). 0: no isothermal levels, n: n isothermal levels, -1: all isothermal levels
`PV_GCR_restart_bal`	3	The No. of GCR inner iterations used to find balanced ψ.
`PV_GCR_restart_unbal`	5	As above, but for unbalanced ^{A}p.
`PV_GCR_maxiter_bal`	180	The number of GCR outer iterations used to find balanced ψ.
`PV_GCR_maxiter_unbal`	250	As above, but for unbalanced ^{A}p.
`PV_GCR_diagnostics`	2	Diagnostic level. 2 outputs norm of GCR residual.
`PV_GCR_precon_option_bal`	3	Preconditioning type in GCR for balanced ψ. 0: no precon, 1: diag precon, 2: vertical precon, 3: ADI precon.
`PV_GCR_precon_option_unbal`	0	As above, but for unbalanced ^{A}p.
`PV_GCR_ADI_kappa_bal`	-1E-8	Parameter used in ADI preconditioner for balanced ψ (used only if PV_GCR_precon_option_bal=3).
`PV_GCR_ADI_kappa_unbal`	-1E-8?	As above, but for unbalanced ^{A}p.
`PV_GCR_ADI_loops_bal`	4	Number of ADI preconditioner loops for balanced ψ (used only if PV_GCR_precon_option_bal=3).
`PV_GCR_ADI_loops_unbal`	4?	As above, but for unbalanced ^{A}p.
`PV_GCR_MassWt_bal`	.false.	Use mass weighted norms in GCR solver for balanced ψ.
`PV_GCR_MassWt_unbal`	.true.	As above, but for unbalanced ^{A}p.
`PVcutoffApmodes`	.true.	Set Ap modes with large horizontal and small vertical scales to zero [7].
`PVcutoff_fac1`	4.59	See [7].
`PVcutoff_fac2`	0.75	See [7].
`PV_Inv_Test_do`	.false.	Must be .true. to allow inverse test to be performed for PV transforms. Used in `Var_Transform_Test`.

Table III: Namelist variables needed to run the new PV-based transforms

Other useful namelist variables for the transforms namelist are the following.

- `OptionGpScale=1` turns off vertical regression of ${}^{G}p'$.
- `hShapePsi(1:100) = 10,10,10,10, ... 10` allows the horizontal spectra (mode-by-mode) to be taken explicitly from the cov file. (default is 13 which uses a geometric average between the cov file spectra and a prescibed function).

The following is useful for the transforms options namelist.

- `ApTruncation = n` allows the Ap spectrum to be truncated to the first *n* modes.

The following controls the level of the transform tests (to switch transform testing on, see Sec. 10).

- `TestDepth = 1` for parameter transform test only, `TestDepth = 2` for parameter and vertical transform tests and `TestDepth = 3` for parameter, vertical and horizontal tests only.

10. Testing the transforms

The code allows several test modes.

- Adjoint tests are run every time *Var_AnalysePF* is run. These have been performed successfully.
- A special inverse test for the PV-based transforms has been incorporated into the new code. It is invoked by running *Var_AnalysePF* in transform test mode. Choose

  ```
  Start point → AnalysePF → AnalysePF program
  Set "Run mode" to "Test operation".
  Set "Test variant" to "Transforms".
  ```

  A non-zero PF guess is required. Firstly this will perform a (new) $\mathbf{T}_p$-transform and output the new V_p and PF fields together with residual fields of PV and anti-PV associated with the GCR solver (see table IV for a full list). The residuals would be zero if the $\mathbf{T}_p$-transform had worked perfectly. This is not the case in practice and the residuals are sometimes significant. The new $\mathbf{U}_p$ transform is then performed on the output of $\mathbf{T}_p$. The balanced and unbalanced pressure and wind fields are output (see table IV). The Field Output namelist needs to be set-up properly. See *vbwi.e* for an example job that uses the PV-based control variables.
- A contribution test can be performed at the end of *Var_AnalysePF* when it is run in its normal operation with the PV-based variables. The contribution test outputs the contributions from each control variable to the *u*, *v* and *p* analysis increments (see table IV for a full list). A way of studying the analysis of the PV-based transforms is to perform pseudo observation tests with a small number of synthetic observations. Some example jobs that perform contribution tests with the PV-based transforms are the following: *vbwi.g* is a pseudo obs test with 5 pressure observations and *vbwi.i* is a pseudo obs test with 5 zonal wind and 5 meridional wind observations. *vbwi.h* and *vbwi.j* are the analogous jobs set up for the standard transforms for comparison.

Field	Stash	File name	Used with	Called from	Notes
V_p % ψ	43500	*CntrlVars.ff*	IT, CT	Var_TpTransform	
V_p % ^{A}p	43502	*CntrlVars.ff*	IT, CT	Var_TpTransform	
V_p % χ	43501	*CntrlVars.ff*	IT, CT	Var_TpTransform	
Balanced *u*	40002	*Balanced_uv.ff*	IT, CT	Var_UpTransform	(1)
Unbalanced *u*	40002	*Unbalanced_uv.ff*	IT, CT	Var_UpTransform	(1)
Divergent *u*	40002	*Divergent_uv.ff*	IT, CT	Var_UpTransform	(1)
Balanced *v*	40003	*Balanced_uv.ff*	IT, CT	Var_UpTransform	(1)
Unbalanced *v*	40003	*Unbalanced_uv.ff*	IT, CT	Var_UpTransform	(1)
Divergent *v*	40003	*Divergent_uv.ff*	IT, CT	Var_UpTransform	(1)
Balanced *p*	40450	*Pressures.ff*	IT, CT	Var_UpPFHp	(1)(2)
Unbalanced *p*	40502	*Pressures.ff*	IT, CT	Var_UpPFHp	(1)(2)
PF % *u*	41002	*CntrlVars.ff*	IT, CT	Var_TpTransform	
PF % *v*	41003	*CntrlVars.ff*	IT, CT	Var_TpTransform	
PF % *p*	41151	*CntrlVars.ff*	IT, CT	Var_TpTransform	
PF % ^{H}p	41474	*CntrlVars.ff*	IT, CT	Var_TpTransform	
PF % ^{G}p	41450	*CntrlVars.ff*	IT, CT	Var_TpTransform	
PV	40500	*PV_Residuals.ff*	IT	Var_GCR_k_PV	(3)
PV residual	40501	*PV_Residuals.ff*	IT	Var_GCR_k_PV	(3)
$\overline{PV}$	40502	*aPV_Residuals.ff*	IT	Var_GCR_k_PV	(3)
$\overline{PV}$ residual	40503	*aPV_Residuals.ff*	IT	Var_GCR_k_PV	(3)
ψ first guess	40500	*PsiGuess.ff*	IT	Var_GCR_k_PV	
ψ^u	40500	*UnbalPsi.ff*	IT	Var_UpTransform	
p^u first guess	40502	*UnbalPGuess.ff*	IT	Var_GCR_k_PV	

Table IV: Fields output from "inverse tests" (IT) and "contribution tests" (CT). See notes below.

Notes about non-standard use of stash codes:

(1) These have been put under section 40 (LS) as it is the most convenient thing to do, even though they are PF fields (section 41).

(2) Balanced pressure has been stored with the same code as ^{G}p (450) and unbalanced pressure has been stored with the same code as ^{A}p (502).

(3) Special stash codes are being sought for PV, PV residual, anti-PV and anti-PV residual.

For a list of current stash codes, see [11].

11. Fields needed for GCR test IDL plotting code

Provisional STASH	Field	Filename	IDL ref	FieldOutput tag
40500	PV	PV_Residuals.ff		PotV_Residuals
40501	PV residual	PV_Residuals.ff		PotV_Residuals
43502 or 40502	anti-PV or $\psi_u = \psi - \psi_b$	aPV_Residuals.ff		aPV_Residuals
40503 or 43503	anti-PV or ψ_u residual	aPV_Residuals.ff		aPV_Residuals
40500	ψ_b first guess	PsiGuess.ff	psitotal	PsiGuess
43500	ψ_b	CntrlVars.ff	psibal	Var_TpTransform
40500	ψ_u = antibalance (p_u)	UnbalPsi.ff	psiunbal	UnbalPsi
	ψ residual $= \psi - (\psi_b + \psi_u)$		data	
41474	p_H	CntrlVars.ff	ptotal	Var_TpTransform
40450	p_b	Pressures	pbal	BalUnbalP
43502	p_u	CntrlVars.ff	punbal	Var_TpTransform
	p residual $= p_H - (p_b + p_u)$			

For IDL code, see link from, e.g., `http://www.met.rdg.ac.uk/~ross/DARC/PVcv/Tests4/PVPlots.pro`

12. Fields needed for pseudo observation IDL plotting code

Provisional STASH	Field	Filename	FieldOutput tag
41002	u (standard and PV codes)	PFAnal	n/a
41003	v (standard and PV codes)	PFAnal	n/a
41151	p (standard and PV codes)	PFAnal	n/a
40002	u (balanced from PV code)	Balanced_uv.ff	Balanced_uv
40003	v (balanced from PV code)	Balanced_uv.ff	Balanced_uv
40450	p (balanced from PV code)	Pressures.ff	BalUnbalP
40450	p (balanced from standard code)	BalP.ff	BalP
40002	u (unbalanced from PV code)	Unbalanced_uv.ff	Unbalanced_uv
40003	v (unbalanced from PV code)	Unbalanced_uv.ff	Unbalanced_uv
40502	p (unbalanced from PV code)	Pressures.ff	BalUnbalP
43500	ψ (standard code)	ControlVars_min.ff	Var_UpTransform
43502	^{A}p (standard code)	ControlVars_min.ff	Var_UpTransform
43501	χ (standard code)	ControlVars_min.ff	Var_UpTransform
43500	ψ_b (PV code)	ControlVars_min.ff	Var_UpTransform
43502	p_u (PV code)	ControlVars_min.ff	Var_UpTransform
43501	χ (PV code)	ControlVars_min.ff	Var_UpTransform

For IDL code, see link from, e.g., `http://www.met.rdg.ac.uk/~ross/DARC/PVcv/Tests4/PseudoObs.pro`

13. References

[1] Bannister R.N., http://www.met.rdg.ac.uk/~ross/DARC/PVcv/PVcv.html

[2] Bannister R.N., http://www.met.rdg.ac.uk/~ross/DARC/PVcv/Intro.html

[3] Bannister R.N., Cullen M.J.P., 2008, http://www.met.rdg.ac.uk/~ross/DARC/PVcv/PVcv.pdf

[4] Bannister R.N., 2008, Diagonal preconditioning of the PV equations, http://www.met.rdg.ac.uk/~ross/DARC/PVcv/Diag.pdf

[5] Bannister R.N., 2008, Approximate 'vertical only' preconditioning of the PV equations, http://www.met.rdg.ac.uk/~ross/DARC/PVcv/TriDiag.pdf

[6] Bannister R.N., 2008, ADI preconditioning for the PV equations, http://www.met.rdg.ac.uk/~ross/DARC/PVcv/ADI.pdf

[7] Bannister R.N., 2008, What is the cut-off total wavenumber for each vertical mode of unbalanced pressure?, http://www.met.rdg.ac.uk/~ross/DARC/PVcv/CutOff.pdf

[8] Bannister R.N., Cullen M.J.P., 2007, A regime-dependent balanced control variable based on potential vorticity, ECWMF Workshop Proceedings on Flow-dependent aspects of Data Assimilation, 11-13 June 2007, ECMWF, Reading, UK, 101-112.

[9] Bannister R.N., Katz D., Cullen M.J.P., Lawless A.S., Nichols N.K., 2008, Modelling of forecast errors in geophysical fluid flows, Int. J. Numer. Meth. Fluids 56, 1147-1153.

[10] Cullen M.J.P., 2003, Four-dimensional variational data assimilation: A new formulation of the background-error covariance matrix based on a potential-vorticity representation, Q.J.R. Meteorol. Soc. 129, 2777-2796.

[11] VAR Technical Documentation Paper 30, UM STASH details for VAR variables.