NWPSAF 1D-Var User Manual

Peter Weston, Met Office, Exeter, UK

NWPSAF-MO-UD-032

Version 1.0: 9th June 2014

This software and documentation was developed within the context of the EUMETSAT satellite Application Facility (NWP SAF). The partners in the NWP SAF are the Met Office, ECMWF, KNMI, and Météo-France.

Appendix B. Minimization

The `NWPSAF_Minimize.f90` routine takes the observations and background profile and outputs a retrieved profile by minimising the cost function, $J(\mathbf{x})$, where

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x_0})^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x_0}) + (\mathbf{y}-\mathbf{y}(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{y}(\mathbf{x})) \qquad (1)$$

Here the observations, **y**, have error covariances **R**; **B** is the error covariance matrix of the *a priori* (background) profile $\mathbf{x}_0$; and $\mathbf{y}(\mathbf{x})$ is the observed radiance that would result for a given atmospheric state **x**.

Rodgers (1976) gives three variations of iterative solution to the minimisation of $J(\mathbf{x})$, two of which are:

$$\mathbf{x_{n+1}} = \mathbf{x_0} + \mathbf{B}\mathbf{H_n}^T(\mathbf{H_n}\mathbf{B}\mathbf{H_n}^T + \mathbf{R})^{-1}[\mathbf{y}-\mathbf{y}(\mathbf{x_n}) - \mathbf{H_n}(\mathbf{x_0}-\mathbf{x_n})] \qquad (2a)$$

and

$$\mathbf{x_{n+1}} = \mathbf{x_0} + (\mathbf{B}^{-1} + \mathbf{H_n}^T\mathbf{R}^{-1}\mathbf{H_n})^{-1}\mathbf{H_n}^T\mathbf{R}^{-1}[\mathbf{y}-\mathbf{y}(\mathbf{x_n}) - \mathbf{H_n}(\mathbf{x_0}-\mathbf{x_n})] \qquad (2b)$$

where $\mathbf{x}_n$ is the n^{th} estimate of the atmospheric profile (the background profile being the 0^{th} estimate) and $\mathbf{H_n} = \nabla_x \mathbf{y}(\mathbf{x_n})$.

Equation (2a) is the more efficient version when there are more profile elements than channels (i.e., **R** is smaller than **B**). However, when there are more channels than profile elements the form in Equation (2b) should be used.

In Rodgers's paper, Eqn. 2a above is Eqn 101, while Eqn. 2b is Eqn. 100. Therefore in this code, when the minimisation is to be done using the former method `NWPSAF_Minimise_101` is called while `NWPSAF_Minimise_100` is called in the latter case.

Equation (2b) also differs from (2a) in that it can also be extended to allow for extra terms in the cost function (e.g., to penalise solutions with supersaturated levels or with superadiabatic lapse rates) and can be modified to become the Marquardt-Levenberg descent algorithm (Rodgers, 2000). This case is dealt with in the `NWPSAF_Minimise_100ML` subroutine

The Marquardt-Levenberg version of Eqn. (2b) with the additional cost function terms is then:

$$\mathbf{x}_{n+1} = \mathbf{x}_0 + [\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{J}'' + \gamma\mathbf{I}]^{-1}\cdot \qquad (3)$$
$$\{\mathbf{H}^T\mathbf{R}^{-1}[(\mathbf{y}-\mathbf{y}(\mathbf{x}_n) + \mathbf{H}(\mathbf{x}_n-\mathbf{x}_0)] - \mathbf{J}' + \mathbf{J}''(\mathbf{x}_n-\mathbf{x}_0) + \gamma(\mathbf{x}_n-\mathbf{x}_0)\}$$

Here J' is the first differential of the additional cost function with respect to **x**, evaluated at $\mathbf{x}_n$, and J'' is the second differential. Thus, J' is a vector with the same length as $\mathbf{x}_n$ and J'' is a matrix with the same size as **B**.

In the Marquardt-Levenberg minimisation method, the value of γ is varied depending on the non-linearity of the problem and the proximity to the desired solution. When $\gamma = 0$, the equation reduces to the Newtonian inverse Hessian method of Eqn (2b). In non-linear cases, it is possible that the step taken by using the Newtonian method will result in an increased cost function. In this case the value of γ is increased (thereby reducing the step size) until a point is reached where the cost function decreases (as $\gamma \to \infty$, Eqn. (3) becomes the method of steepest descent). After an improved solution (i.e., lower cost function) has been found, the value of γ may once again be increased to allow larger steps in the next iteration.

The `NWPSAF_Minimise` subroutine allows for either the Newtonian or Marquardt-Levenberg algorithms to be used and for the inclusion of additional cost function terms.

In all cases the minimisation is implemented by solving Eqn. 2a, 2b or 3 without explicitly inverting the matrix. This achieved through the use of the Cholesky Decomposition method of solving linear symmetric positive definite systems of equations (e.g., Golub and van Loan, 1996, p.143ff).

Convergence is deemed to have occurred when:

- The cost function changes by no more than 1% (this value may be modified through the `DeltaJ` variable in the ControlData.NL namelist file).
- If using the Marquardt-Levenbverg minimisation, the normalised cost function gradient is less than the square of the cost function (this criterion may be multiplied by the value of `SmallJCost_Gradient` in the ControlData.NL namelist file).
- If using the Marquardt-Levenbverg minimisation, γ decreased in the previous iteration.

The cost function gradient is normalised using the **B**-matrix as a metric:

$$\frac{\partial J}{\partial \mathbf{x}}^T \mathbf{B} \frac{\partial J}{\partial \mathbf{x}}$$

The cost function gradient is used as an additional criterion for convergence in the Marquardt-Levenberg case as if gamma is large it is possible that the cost function will not change between interations simply because of the small step size in the minimisation. The choice of the cost function squared as the convergence criterion in this case is such that if the solution changes by a distance eqivalent to the *a priori* error, the cost value should not change by more than 100% — it is therefore quite a loose additional criterion.

In the case where the cloud top pressure and cloud fraction are being retrieved (with very large assumed bacground error variances), the appropriate elements of the cost function gradient are set to zero. The cost function gradient is not calculated and is set to zero if Eqn. 2a is being used for minimisation.

The manipulation of the observational error covariance matrix in its various forms (see Section 2.2.4) requires some additional code in all four of the routines considered in this section. For the band diagonal case (except for diagonal matrices which are trivial to manipulate), further subroutines are called to invert and multiply the matrix. The inversion of a band diagonal matrix is handled by the `NWPSAF_Band_Inverse` subroutine which uses a Band Cholesky routine in the process (Golub and van Loan, 1996, pp.155-6).

[Return to Top]

References.

Golub, G.H. and C.F. Van Loan (1996). *Matrix Computations.* The Johns Hopkins University Press.

Rodgers, C. D. (1976). Retrieval of atmospheric temperature and composition from remote measurements of thermal radiation. *R. Geophys. Space Phys.*, **14**, 609-624.

Rodgers, C. D. (2000). *Inverse Methods for Atmospheres: Theory and Practice. World Scientific Publishing.*

[Return to Top]

[Return to Main Manual]