

GRAS SAF Report 04
Ref: SAF/GRAS/METO/REP/GSR/004
Web: www.grassaf.org
Date: 23 November 2007

The EUMETSAT
Network of
Satellite Application
Facilities



GRAS SAF Report 04

Error function calculation in ROPP

Huw Lewis

Met Office, UK

Document Author Table

	Name	Function	Date	Comments
Prepared by:	H. Lewis	GRAS SAF Project Team	23 November 2007	
Reviewed by:	D. Offiler	GRAS SAF Project Team	23 November 2007	
Approved by:	K.B. Lauritsen	GRAS SAF Project Manager	17 October 2008	

Document Change Record

Issue/Revision	Date	By	Description
Version 1.0	23 Nov 2007	HL	First version

GRAS SAF Project

The GRAS SAF is a EUMETSAT-funded project responsible for operational processing of GRAS radio occultation data from the Metop satellites. The GRAS SAF delivers bending angle, refractivity, temperature, pressure, and humidity profiles in near-real time and offline for NWP and climate users. The offline profiles are further processed into climate products consisting of gridded monthly zonal means of bending angle, refractivity, temperature, humidity, and geopotential heights together with error descriptions.

The GRAS SAF also maintains the Radio Occultation Processing Package (ROPP) which contains software modules that will aid users wishing to process, quality-control and assimilate radio occultation data from any radio occultation mission into NWP and other models.

The GRAS SAF Leading Entity is the Danish Meteorological Institute (DMI), with Cooperating Entities: i) European Centre for Medium-Range Weather Forecasts (ECMWF) in Reading, United Kingdom, ii) Institut D'Estudis Espacials de Catalunya (IEEC) in Barcelona, Spain, and iii) Met Office in Exeter, United Kingdom. To get access to our products or to read more about the project please go to <http://www.grassaf.org>.

1 Background

The bending angle forward model implemented in ROPP requires the computation of the Abel transform,

$$\alpha(a) = 2 \int_{r_t}^{\infty} d\alpha = -2a \int_{r_t}^{\infty} \frac{1}{\sqrt{r^2 n^2 - a^2}} \frac{d \ln(n)}{dr} dr = -2a \int_a^{\infty} \frac{1}{\sqrt{x^2 - a^2}} \frac{d \ln(n)}{dx} dx \quad (1.1)$$

where $\alpha(a)$ is the bending angle at impact parameter a , $n = (1 + 10^{-6}N)$ is the refractive index of the atmosphere along the ray path and parameter $x = nr = (1 + 10^{-6}N)r$ for a location at range r from the centre of curvature.

This is performed (in routine `ropp_1dvar_abel`) by re-arranging the integral term to give the bending between the j and $j + 1$ levels as

$$\Delta\alpha_j = 10^{-6} \sqrt{2\pi a k_j N_j} \exp(k_j(x_j - a)) [\text{erf}(k_j(x_{j+1} - a)) - \text{erf}(k_j(x_j - a))] \quad (1.2)$$

The variable k_j is defined as

$$k_j = \frac{\ln(N_j/N_{j+1})}{(x_{j+1} - x_j)} \quad (1.3)$$

and `erf` is the error function defined as

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt \quad (1.4)$$

Expressing the Abel transform in terms of a standard integral considerably simplifies its computation.

1.1 Computing the error function

In ROPP v1.0, the error function was evaluated using a third-party library routine provided as part of the public domain DCDFLIB collection from the University of Texas.

Feedback from users at ECMWF (S Healy, pers. comm), where the GRAS-SAF algorithms are implemented in operational processing, highlighted that computation of the error function using a call to a different external function was very costly. This is because the error function will be called millions of times in solving the 1dVar problem for all observations within a given assimilation window.

A more efficient method of computing the error function is therefore required. This is to be achieved using a standard polynomial approximation within the Abel transform subroutine, avoiding the need to call an external function. The expression used is taken from (1) and is of the form

$$\text{erf}(x) \approx 1.0 - (0.3480242z - 0.0958798z^2 - 0.7478556z^3) \exp(-x^2) \quad (1.5)$$

where

$$z = \frac{1.0}{(1.0 + 0.47047x)} \quad (1.6)$$

This expression is only a valid approximation for $x > 0$, but the values of $k_j(x_{j+1} - a)$ and $k_j(x_j - a)$ are required to be positive values within the code. The numerical results of Equation 1.5 equal those of the DCDFLIB library routine to within 10^{-5} . This change is to be implemented in future versions of ROPP.

2 Tangent linear and adjoint code

The 1dVar processing requires computation of the tangent linear and adjoint of the forward model. In ROPP v1.0, the derivative of `erf` was taken to be

$$\text{erf}_{t_l} = \frac{2}{\sqrt{\pi}} \exp(-t^2) t_{t_l} \quad (2.1)$$

where $t = (k_j(x_j - a))^{1/2}$ and t_{t_l} is the tangent linear of that variable. To be strictly correct, the tangent linear should be given by the derivative of the algorithm used to compute the error function rather than the mathematical expression.

For the case of the DCDFLIB library routine, computing the tangent linear and adjoint would be rather complicated and require several additional computations to be made. In contrast, it is straightforward to compute the tangent linear of the polynomial approximation for `erf`. Its implementation in ROPP therefore allows for a consistent treatment of `erf` in the forward model and its tangent linear and adjoint codes.

The forward model computes the difference $\text{erf}(k_j(x_{j+1} - a)) - \text{erf}(k_j(x_j - a))$ as

```
zt_low = 1.0d0/(1.0d0+0.47047d0*t_lower)
erf_low = 1.0d0-(a-(b-c*zt_low)*zt_low)*zt_low*EXP(-(t_lower*t_lower))

zt_up = 1.0d0/(1.0d0+0.47047d0*t_upper)
erf_up = 1.0d0-(a-(b-c*zt_up)*zt_up)*zt_up*EXP(-(t_upper*t_upper))

integral_erf = erf_up - erf_low
```

where variables `t_upper` and `t_lower` correspond to $k_j(x_{j+1}-a)$ and $k_j(x_j-a)$ respectively and variables `a`, `b` and `c` define the constants written in Equation 1.5.

The tangent linear is then written as

```
zt_low = 1.0d0/(1.0d0+0.47047d0*t_lower)
zt_low_t1 = -(0.47047d0*t_lower_t1)*(zt_low*zt_low)

erf_low = 1.0d0-(a-(b-c*zt_low)*zt_low)*zt_low*EXP(-(t_lower*t_lower))
erf_low_t1 = ((a-(b-c*zt_low)*zt_low)*zt_low*2.0d0*t_lower*t_lower_t1 &
              -(a-(2.0d0*b-3.0d0*c*zt_low)*zt_low)*zt_low_t1) &
              *EXP(-(t_lower*t_lower))

zt_up = 1.0d0/(1.0d0+0.47047d0*t_upper)
zt_up_t1 = -(0.47047d0*t_upper_t1)*(zt_up*zt_up)

erf_up = 1.0d0-(a-(b-c*zt_up)*zt_up)*zt_up*EXP(-(t_upper*t_upper))
erf_up_t1 = ((a-(b-c*zt_up)*zt_up)*zt_up*2.0d0*t_upper*t_upper_t1 &
              -(a-(2.0d0*b-3.0d0*c*zt_up)*zt_up)*zt_up_t1) &
```

```
*EXP(-(t_upper*t_upper))
```

```
integral_erf = erf_up - erf_low
```

```
integral_erf_tl = erf_up_tl - erf_low_tl
```

The tangent linear expressions are then used to construct the adjoint code.

```
erf_low_ad = erf_low_ad - integral_erf_ad
```

```
erf_up_ad = erf_up_ad + integral_erf_ad
```

```
integral_erf_ad = 0.0_wp
```

```
t_upper_ad = t_upper_ad + (a-(b-c*zt_up)*zt_up)*zt_up      &  
                *EXP(-(t_upper*t_upper))*2.0d0*t_upper*erf_up_ad
```

```
zt_up_ad = zt_up_ad - (a-(2.0d0*b-3.0d0*c*zt_up)*zt_up)    &  
                *EXP(-(t_upper*t_upper))*erf_up_ad
```

```
erf_up_ad = 0.0_wp
```

```
t_lower_ad = t_lower_ad + (a-(b-c*zt_low)*zt_low)*zt_low  &  
                *EXP(-(t_lower*t_lower))*2.0d0*t_lower*erf_low_ad
```

```
zt_low_ad = zt_low_ad - (a-(2.0d0*b-3.0d0*c*zt_low)*zt_low) &  
                *EXP(-(t_lower*t_lower))*erf_low_ad
```

```
erf_low_ad = 0.0_wp
```

```
t_lower_ad = t_lower_ad - 0.47047d0*zt_low*zt_low*zt_low_ad
```

```
zt_low_ad = 0.0_wp
```

```
t_upper_ad = t_upper_ad - 0.47047d0*zt_up*zt_up*zt_up_ad
```

```
zt_up_ad = 0.0_wp
```

3 ROPP 1dVar results

The performance of the ROPP 1dVar retrieval using the DCDFLIB routine and the polynomial approximation for *erf* has been compared. The stand-alone tool *ropp_1dvar_bangle* has been applied to retrieve temperature, humidity and pressure profiles from the bending angle observations included in the ROPP test procedure IT-1DVAR-03. This procedure includes data from 8 different radio occultation scenarios for testing.

Tables 3.1-3.5 list the cost function values at each iteration for the different observed profiles processed using the *erf* calculation provided as part of the DCDFLIB library (as used in ROPP v1.0) and the polynomial approximation in Equation 1.5. The maximum relative change in the state vector is also listed for each iteration. This change is computed as a fraction of the background error covariance. Convergence is assumed to occur when the maximum fractional change detected between iterations is less than 0.1 for two successive iterations. The processing time for each 1dVar retrieval is listed for each case.

Figure 3.1 shows plots of the retrieved temperature, humidity and pressure profiles for one of the IT-1DVAR-03 occultations processed. The difference between results obtained using DCDFLIB and the polynomial approximation are plotted in green. These plots quantify the impact of using the polynomial approximation to compute *erf* on the 1dVar retrievals compared with the DCDFLIB library routine provided with ROPP v1.0.

3.1 Summary

These tests indicate that the performance of the polynomial approximation is very similar to that of the *erf* computation in the DCDFLIB library. Small differences in the cost function values in Tables 3.1-3.5 result from changes to the forward model used in computing the cost function, and the resulting modifications to the adjoint code used in the minimisation of the cost function. Of the eight profiles tested here, all reached convergence more quickly (by typically 0.015 s) using the method computing *erf* using the polynomial approximation than by using the DCDFLIB routine.

The retrieved profiles plotted in Figures 3.1 demonstrate the impact of the numerical differences between the DCDFLIB and polynomial approximation methods on the 1dVar output. Results for all eight test profiles have maximum differences in temperature of 0.005 K, in specific humidity of 0.0005 g/kg and in pressure of 0.002 hPa. These are negligible and well within the quality tolerances required.

The method used to compute *erf* in the Abel transform algorithm used as part of the bending angle operator in ROPP therefore has minimal impact on the retrieved atmospheric profiles. Given the cost incurred by calling the external library routine DCDFLIB it is recommended that users implement the polynomial approximation provided in *ropp_fm_abel*, *ropp_fm_abel_t1* and *ropp_fm_abel_ad* in future releases of ROPP, with no impact on data quality expected. Further, this change has allowed consistency between the Abel

transform in the forward model and its tangent linear and adjoint codes.

DCDFLIB			polynomial approximation		
n_iter	J	max. relative change state	n_iter	J	max. relative change state
1	206.48	-	1	206.56	-
2	120.35	.34257	2	120.32	.34271
3	63.985	.54345	3	63.927	.54354
4	52.375	.17836	4	52.301	.17842
5	44.413	.11091	5	44.328	.11136
6	31.633	.25184	6	31.559	.25190
7	25.640	.12977	7	25.564	.12980
8	19.131	.21257	8	19.065	.21085
9	15.758	.17157	9	15.727	.17200
10	13.882	.60927E-01	10	13.840	.60724E-01
11	12.402	.89095E-01	11	12.376	.88194E-01
CPU time taken: 0.285 s			CPU time taken: 0.266 s		

Table 3.1: Comparison of cost function values at each iteration applying the ROPP 1dVar to bending angle observations and Profile 1 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to *erf* in the Abel transform.

DCDFLIB			polynomial approximation		
n_iter	J	max. relative change state	n_iter	J	max. relative change state
1	143.77	-	1	143.76	-
2	118.05	.12560	2	118.04	.12551
3	52.345	.66877	3	52.293	.66871
4	41.185	.20322	4	41.129	.20314
5	33.272	.16718	5	33.210	.16752
6	25.145	.19800	6	25.101	.19752
7	18.217	.19741	7	18.173	.19758
8	14.157	.20570	8	14.129	.20461
9	12.107	.12276	9	12.089	.12226
10	10.506	.11588	10	10.487	.11609
11	9.2992	.10583	11	9.2803	.10608
12	8.6456	.11681	12	8.6296	.11666
13	8.4972	.21000E-01	13	8.4819	.20696E-01
14	8.4248	.15080E-01	14	8.4093	.15139E-01
CPU time taken: 0.374 s			CPU time taken: 0.360 s		

Table 3.2: Comparison of cost function values at each iteration applying the ROPP 1dVar to bending angle observations and Profile 2 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to *erf* in the Abel transform.

DCDFLIB			polynomial approximation		
n_iter	J	max. relative change state	n_iter	J	max. relative change state
1	83.261	-	1	83.249	-
2	68.185	.12084	2	68.175	.12075
3	28.811	.56841	3	28.796	.56826
4	23.043	.21358	4	23.026	.21341
5	16.775	.37981	5	16.758	.37985
6	13.539	.27744	6	13.528	.27697
7	11.908	.82347E-01	7	11.902	.81989E-01
8	11.050	.12259	8	11.042	.12245
9	10.809	.72700E-01	9	10.801	.72952E-01
10	10.628	.46099E-01	10	10.618	.46698E-01
CPU time taken: 0.255 s			CPU time taken: 0.254 s		

Table 3.3: Comparison of cost function values at each iteration applying the ROPP 1dVar to bending angle observations and Profile 3 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to erf in the Abel transform.

DCDFLIB			polynomial approximation		
n_iter	J	max. relative change state	n_iter	J	max. relative change state
1	108.34	-	1	108.29	-
2	88.868	.12705	2	88.825	.12698
3	38.660	.71466	3	38.622	.71433
4	30.947	.17916	4	30.910	.17893
5	25.172	.12161	5	25.129	.12194
6	16.424	.28121	6	16.387	.28111
7	13.225	.17619	7	13.203	.17706
8	10.747	.18151	8	10.737	.17813
9	9.5489	.95164E-01	9	9.5373	.95866E-01
10	9.2172	.10902	10	9.2140	.11053
11	8.9385	.30270E-01	11	8.9267	.30981E-01
12	8.8851	.11664E-01	12	8.8739	.11435E-01
CPU time taken: 0.320 s			CPU time taken: 0.304 s		

Table 3.4: Comparison of cost function values at each iteration applying the ROPP 1dVar to bending angle observations and Profile 4 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to erf in the Abel transform.

DCDFLIB			polynomial approximation		
n_iter	J	max. relative change state	n_iter	J	max. relative change state
1	77.874	-	1	77.852	-
2	63.821	.14791	2	63.802	.14785
3	28.042	.59780	3	28.028	.59754
4	22.870	.24684	4	22.855	.24665
5	16.235	.54133	5	16.220	.54088
6	16.056	.70021	6	16.046	.69874
7	12.707	.17175	7	12.698	.17166
8	12.252	.43038E-01	8	12.244	.43051E-01
9	11.791	.15954	9	11.782	.15974
10	11.501	.14469	10	11.492	.14475
11	11.396	.15750	11	11.387	.15741
12	11.344	.61642E-01	12	11.334	.61306E-01
13	11.339	.11664E-02	13	11.330	.90697E-02
CPU time taken: 0.345 s			CPU time taken: 0.331 s		

Table 3.5: Comparison of cost function values at each iteration applying the ROPP 1dVar to bending angle observations and Profile 5 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to *erf* in the Abel transform.

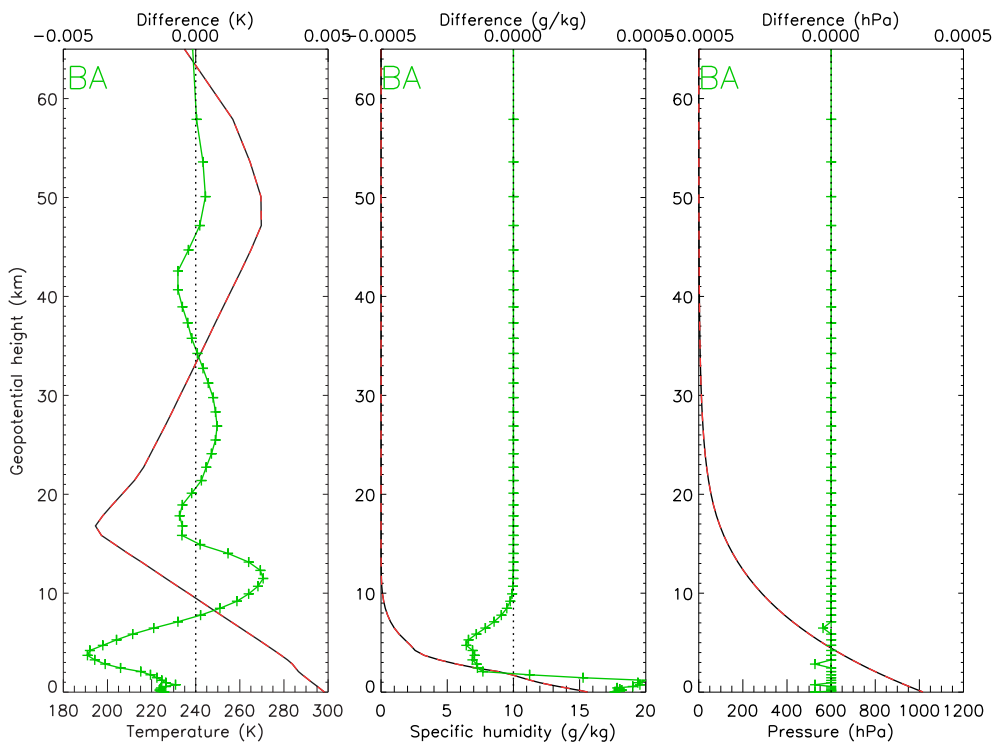


Figure 3.1: Comparison of retrieved profiles of temperature, humidity and pressure for Profile 1 in the ROPP 1dVar module test IT-1DVAR-03 using the DCDFLIB and polynomial approximation to *erf* in the Abel transform. The difference in profiles resulting from the difference of *erf* calculation is plotted in green.

Bibliography

- [1] Abramowitz, M. and Stegun, I.A. (1965) Handbook of mathematical functions, Dover.

GRAS SAF Reports

SAF/GRAS/METO/REP/GSR/001	Mono-dimensional thinning for GPS Radio Occultation
SAF/GRAS/METO/REP/GSR/002	Geodesy calculations in ROPP
SAF/GRAS/METO/REP/GSR/003	ROPP minimiser - minROPP
SAF/GRAS/METO/REP/GSR/004	Error function calculation in ROPP
SAF/GRAS/METO/REP/GSR/005	Refractivity calculations in ROPP
SAF/GRAS/METO/REP/GSR/006	Levenberg-Marquardt minimisation in ROPP
SAF/GRAS/METO/REP/GSR/007	Abel integral calculations in ROPP
SAF/GRAS/METO/REP/GSR/008	ROPP thinner algorithm
SAF/GRAS/METO/REP/GSR/009	Refractivity coefficients used in the assimilation of GPS radio occultation measurements
SAF/GRAS/METO/REP/GSR/010	Latitudinal Binning and Area-Weighted Averaging of Irregularly Distributed Radio Occultation Data
SAF/GRAS/METO/REP/GSR/011	ROPP 1dVar validation

GRAS SAF Reports are accessible via the GRAS SAF website <http://www.grassaf.org>.