Balance equations as background constraints in variational assimilation

Ole Vignes, 23.04.2007, Hirlam/Aladin meeting

Outline

1. Balance equations, what and why

2. Vertical coordinate

3. Solution methods:

   (a) "Direct" inversion

   (b) Weak constraints

   (c) Strong constraints

4. Some results

5. Conclusions and further plans

Nonlinear balance equation

$$\nabla_p^2 \Phi = \left[f\nabla^2\psi + \frac{\partial f}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial \psi}{\partial y} + 2\,\frac{\partial^2\psi}{\partial x^2}\frac{\partial^2\psi}{\partial y^2} - 2\left(\frac{\partial^2\psi}{\partial x\,\partial y}\right)^2 \right]_p$$

Subscript p: horizontal derivatives *at constant pressure*.

Tangent linear version, with rotational winds instead of stream-function:

$$\nabla_p^2 \Phi' = f\left(\frac{\partial v_r'}{\partial x} - \frac{\partial u_r'}{\partial y}\right)_p + \frac{\partial f}{\partial x}v_r' - \frac{\partial f}{\partial y}u_r'$$
$$- \; 2\left(\frac{\partial \overline{v_r}}{\partial x}\frac{\partial u_r'}{\partial y} + \frac{\partial \overline{u_r}}{\partial y}\frac{\partial v_r'}{\partial x}\right)_p + 2\left(\frac{\partial \overline{v_r}}{\partial y}\frac{\partial u_r'}{\partial x} + \frac{\partial \overline{u_r}}{\partial x}\frac{\partial v_r'}{\partial y}\right)_p$$

The horizontal bar means the 'linearization point' (first guess). This gives flow dependency to the resulting increments.

Omega equation, QG version

$$\sigma \nabla_p^2 \omega' + f_0^2 \frac{\partial^2 \omega'}{\partial p^2} = -2\,\nabla_p \cdot \mathbf{Q}' + f_0 \frac{\partial f}{\partial x} \frac{\partial u_g'}{\partial p} + f_0 \frac{\partial f}{\partial y} \frac{\partial v_g'}{\partial p}$$

where

$$Q_x' = -\frac{R_d}{p} \left[\frac{\partial \overline{u_g}}{\partial x} \frac{\partial T'}{\partial x} + \frac{\partial u_g'}{\partial x} \frac{\partial \overline{T}}{\partial x} + \frac{\partial \overline{v_g}}{\partial x} \frac{\partial T'}{\partial y} + \frac{\partial v_g'}{\partial x} \frac{\partial \overline{T}}{\partial y} \right]_p$$

$$Q_y' = -\frac{R_d}{p} \left[\frac{\partial \overline{u_g}}{\partial y} \frac{\partial T'}{\partial x} + \frac{\partial u_g'}{\partial y} \frac{\partial \overline{T}}{\partial x} + \frac{\partial \overline{v_g}}{\partial y} \frac{\partial T'}{\partial y} + \frac{\partial v_g'}{\partial y} \frac{\partial \overline{T}}{\partial y} \right]_p$$

Note flow dependency and similarity to Φ'–equation.

Hopefully, increments consistent with these equations will be less modified by the digital filter initialization.

Vertical coordinate

Let η be HIRLAM's hybrid vertical coordinate.
For any scalar field $a(x, y, \eta)$, for $s = x, y$:

$$\left(\frac{\partial a}{\partial s}\right)_p = \left(\frac{\partial a}{\partial s}\right)_\eta - \frac{\partial a}{\partial p}\left(\frac{\partial p}{\partial s}\right)_\eta = \left(\frac{\partial a}{\partial s}\right)_\eta - \frac{\partial a}{\partial \eta}\frac{\partial \eta}{\partial p}\left(\frac{\partial p}{\partial s}\right)_\eta$$

Let $m = \partial p/\partial \eta$ and drop η subscript on horizontal derivatives.
Then

$$\nabla_p\, a = \nabla a - \frac{1}{m}\frac{\partial a}{\partial \eta}\nabla p$$

For any vector field $\mathbf{b}(x, y, \eta)$:

$$\nabla_p \cdot \mathbf{b} = \nabla \cdot \mathbf{b} - \frac{1}{m}\,\frac{\partial \mathbf{b}}{\partial \eta} \cdot \nabla p,$$

so

$$\nabla_p \cdot \nabla_p\, a = \nabla \cdot \left(\nabla a - \frac{1}{m}\frac{\partial a}{\partial \eta}\nabla p\right) - \frac{1}{m}\frac{\partial}{\partial \eta}\left(\nabla a - \frac{1}{m}\frac{\partial a}{\partial \eta}\nabla p\right) \cdot \nabla p.$$

"Direct" inversion

When discretized, the equations reduce to linear systems of equations of the form

$$Ax = b.$$

With some care (e.g., zero increments on boundaries), it is possible to ensure that A is symmetric and positive definite in both cases (Φ and ω). Given the size and complexity of A it is also natural to look for methods only requiring products of A with a vector, and no knowledge of its coefficient structure. Such methods are iterative, and usually variants of the conjugate gradients algorithm.

But Φ and ω are not prognostic variables in HIRLAM. To go from geopotential increments to temperature increments we may e.g. use the hypsometric equation (discretized):

$$\Phi' = R_d \int_{p_s}^{p(\eta)} T'(1 + (\delta - 1)\overline{q}) \frac{dp}{\overline{p}}.$$

To go from omega increments to divergence increments we may use the continuity equation. But this is not as simple on hybrid levels as on pressure levels:

$$\nabla_p \cdot \mathbf{v}' = \nabla \cdot \mathbf{v}' - \frac{1}{m} \frac{\partial \mathbf{v}'}{\partial \eta} \cdot \nabla p = -\frac{1}{m} \frac{\partial \omega'}{\partial \eta}.$$

Ideally, a new system of equations must be solved for the wind increments. As an approximation, in the "terrain correction term" we may use only the rotational wind and move this term to the right hand side. The effect of this is unknown.

"Direct" inversion (2)

Direct inversion of the equations has some disadvantages:

- To compute the gradient of the cost function, adjoint equations must also be solved. And unless both the forward and adjoint equations are solved accurately, the gradient of the cost function becomes inaccurate.
- Alternatively, we could program the exact adjoint of the forward solution procedure. But in case of conjugate gradients, this procedure is nonlinear. This means that all the iteration steps must be saved before the adjoint computations. It also destroys the quadratic form of the cost function.

The variational algorithm is already iterative. Can we exploit this?

Weak constraints

Define the residual:

$$r(\delta x) = \begin{bmatrix} r_\Phi \\ r_\omega \end{bmatrix} = \begin{bmatrix} b_\Phi - A_\Phi \delta\Phi \\ b_\omega - A_\omega \, \delta\omega \end{bmatrix} = L\delta x$$

Add a new penalty term to the cost function:

$$J(\delta x) = J_b + J_o + J_{be},$$

where

$$J_{be} = \frac{1}{2} r^\top W r = \frac{1}{2} \delta x^\top L^\top W L \, \delta x.$$

So far the weighting matrix W has been diagonal, but the residuals r_Φ and r_ω have been scaled to have similar magnitude.

The quadratic form of the cost function is kept.

Weak constraints (2)

The advantage of weak constraints is that no explicit inversion is needed, this is done implicitly along with the minimization of the cost function. Only forward computations of $\delta\Phi$ and $\delta\omega$ are needed, by the standard Hirlam procedure. The adjoint operator L^{T} must be coded, but this is not difficult.

The disadvantage is that the amount of balance depends on the weighting of the J_{be} term. A high weight destroys the effect of the preconditioning inherent in the transformation to the control variable that diagonalizes the background term:

$$\chi = U\delta x.$$

Strong constraints

If we want to avoid the explicit weighting of the constraint term, we can pose the assimilation problem as a constrained minimization problem:

$$\begin{array}{ll} \text{minimize} & J(\chi) \\ \text{subject to} & C\chi = 0, \end{array}$$

where $C = U^{-1}L$. Define the Lagrangian:

$$\mathcal{L}(\chi, \lambda) = J(\chi) + \lambda^{\mathsf{T}} C\chi.$$

A constrained minimizer for J must be a stationary point of $\mathcal{L}$:

$$\begin{aligned} \nabla_\chi \mathcal{L} &= \nabla_\chi J + C^{\mathsf{T}}\lambda = 0 \\ \nabla_\lambda \mathcal{L} &= C\chi = 0 \end{aligned}$$

The Hessian of the Lagrangian is not positive definite.

Strong constraints (2)

Define the merit function (quadratic with sym.pos.def. Hessian):

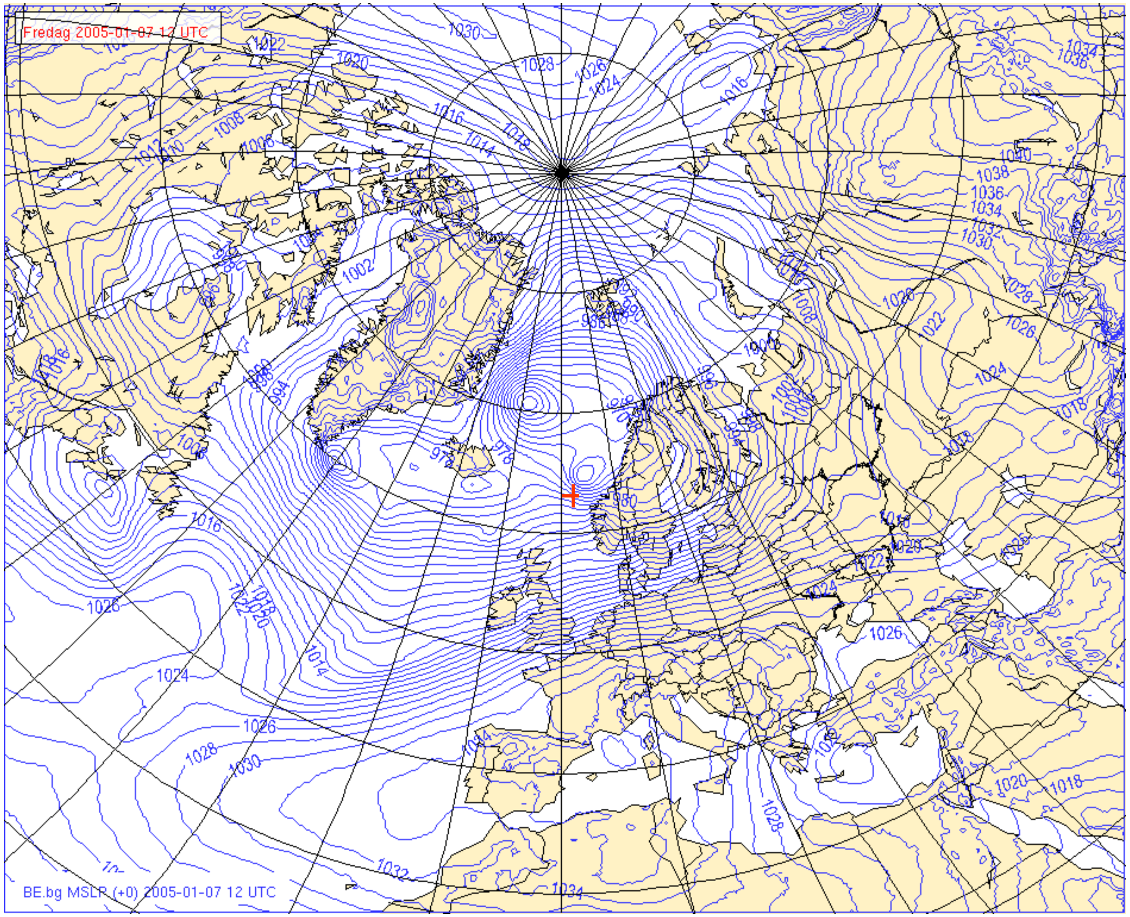
$$m(\chi, \lambda) = \frac{1}{2}\left(\nabla_\chi \mathcal{L}^\mathsf{T} \nabla_\chi \mathcal{L} + \nabla_\lambda \mathcal{L}^\mathsf{T} \nabla_\lambda \mathcal{L}\right).$$

Then we may apply a standard unconstrained minimization method (e.g. conjugate gradients) to $m(\chi, \lambda)$.

However, the constraints are not satisfied exactly by this method, we again depend on the scaling of the two gradient terms. Also, the method requires two evaluations of the cost function at each iteration, instead of one. It therefore appeared to be even more expensive than the weak constraints method.

Some results

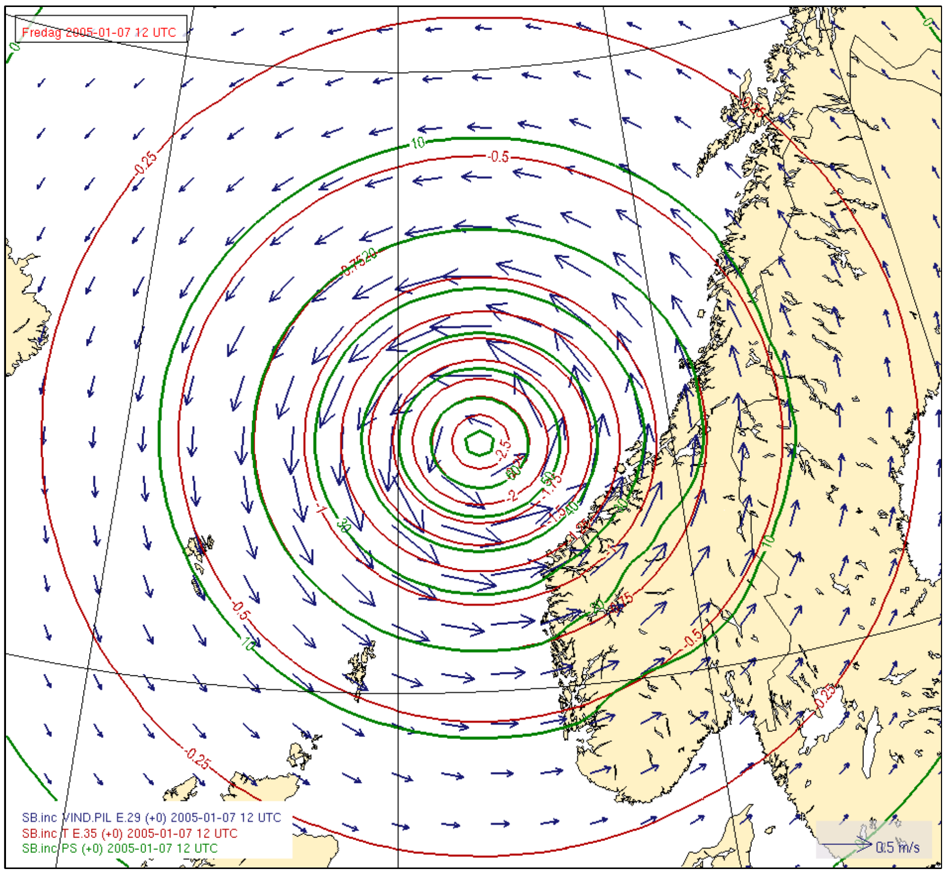
Single obs. experiment, one T increment of -5 °C at 925 hPa, 7/1-2005



Increments of T, wind, ps

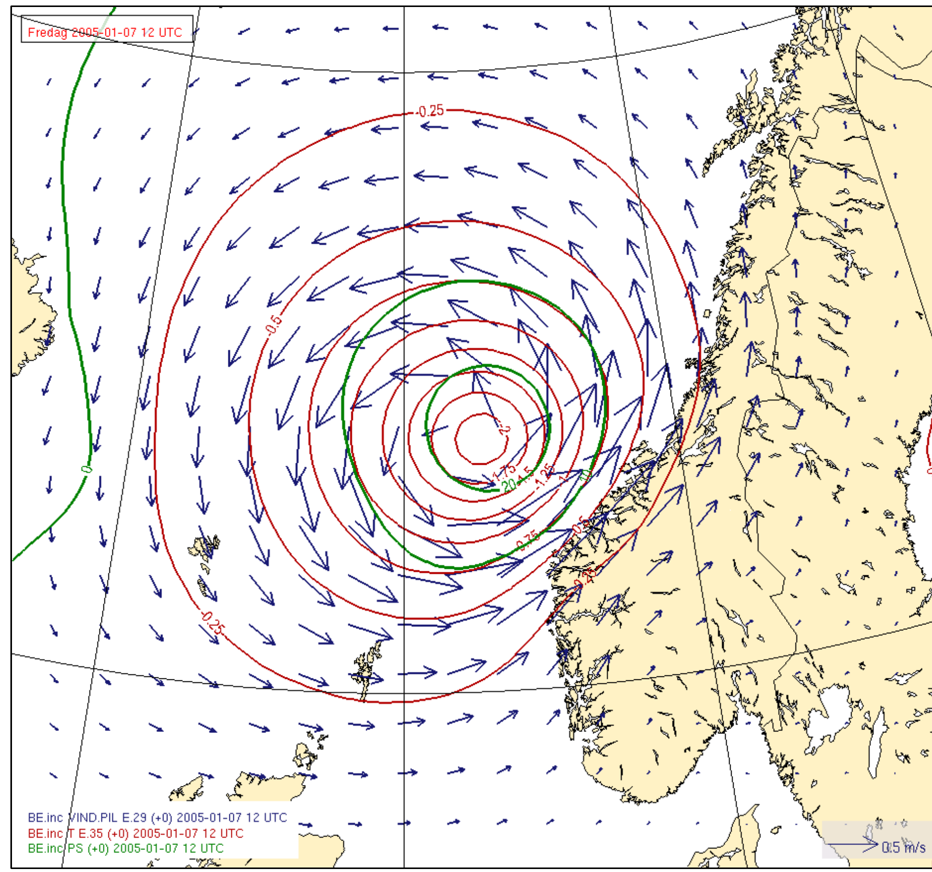
Statistical balance, Jbe = 0

3 iterations



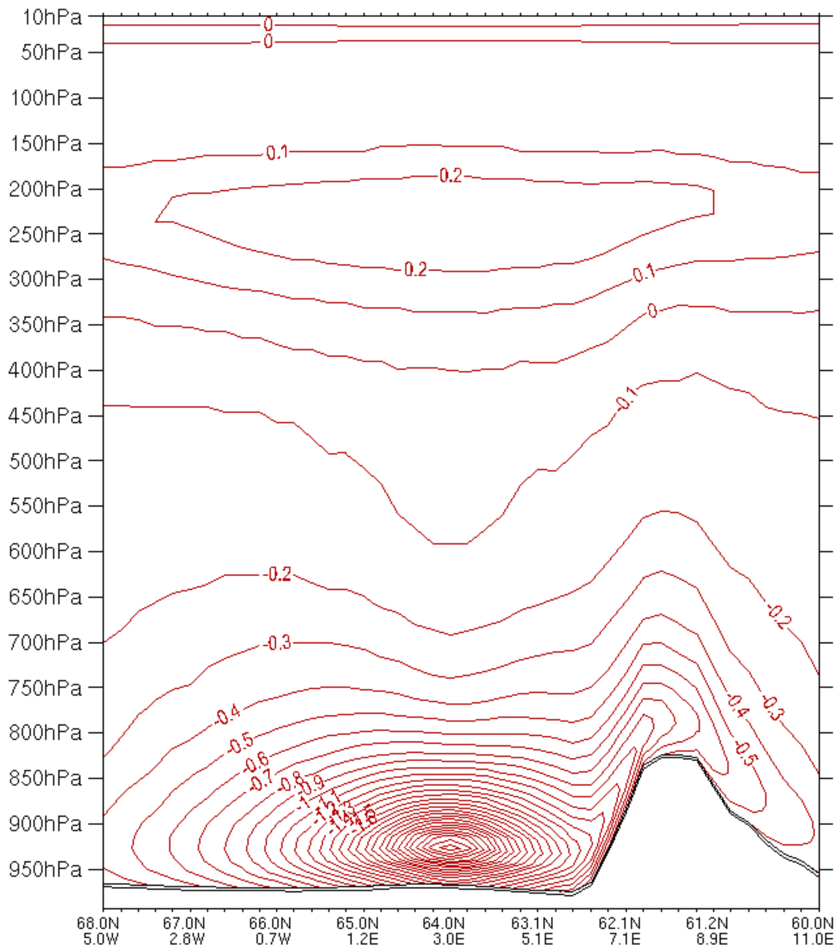
Weak constraints, Jbe weight = 10.

109 iterations

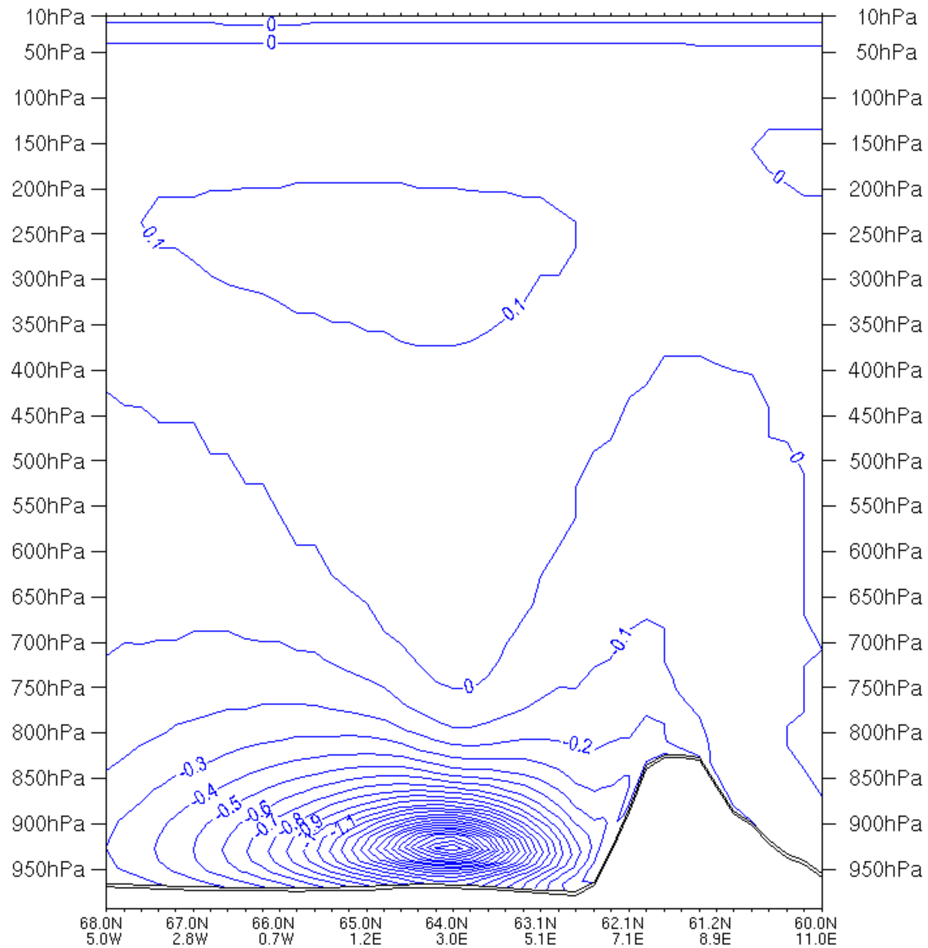


Vertical crossection of T increments

Statistical balance



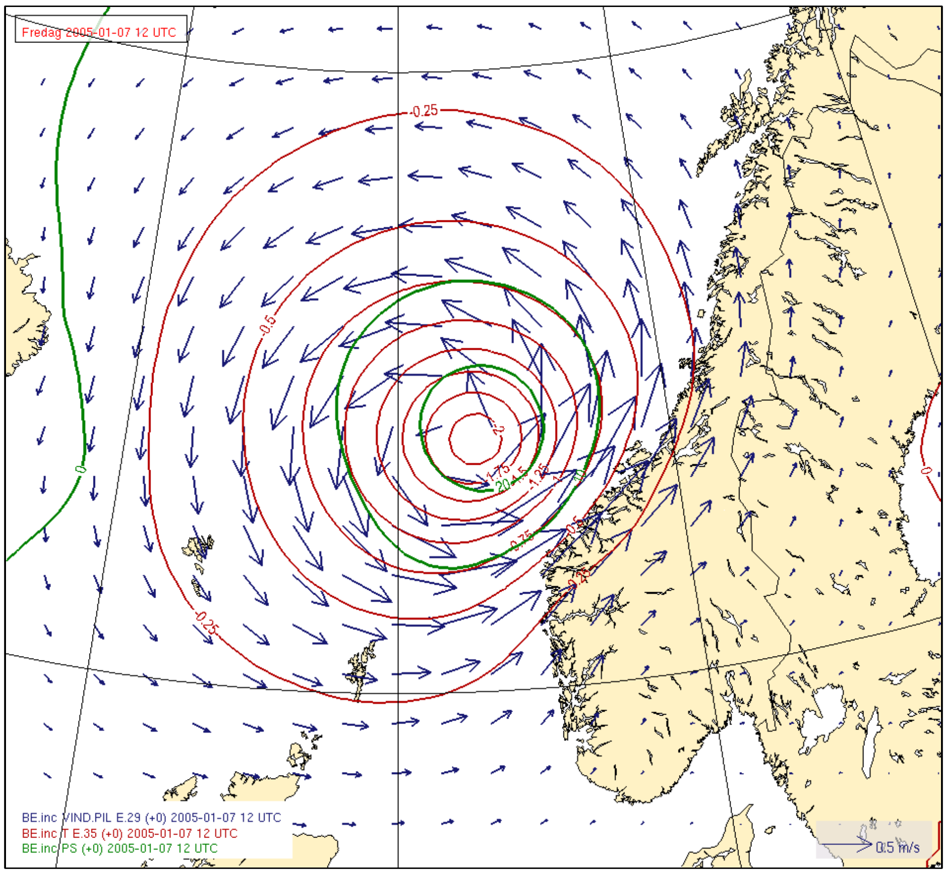
Weak constraints balance eq.



Increments of T, wind, ps

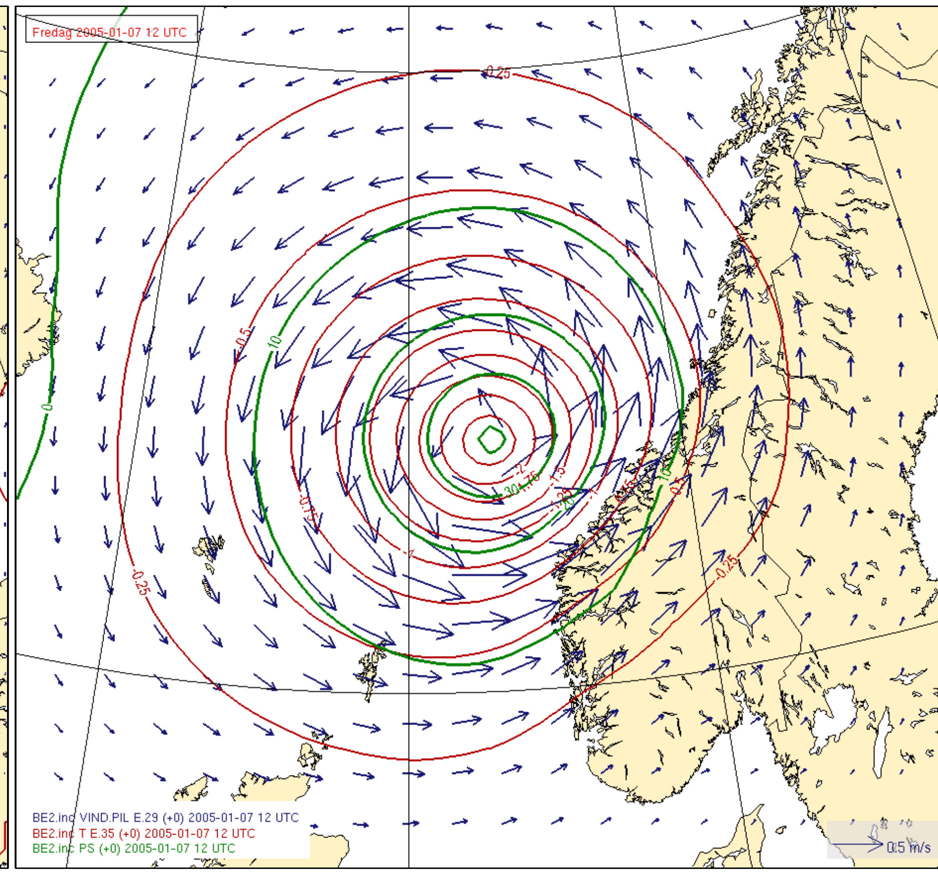
Weak constraints, weight = 10.

109 iterations



Weak constraints, weight = 1.

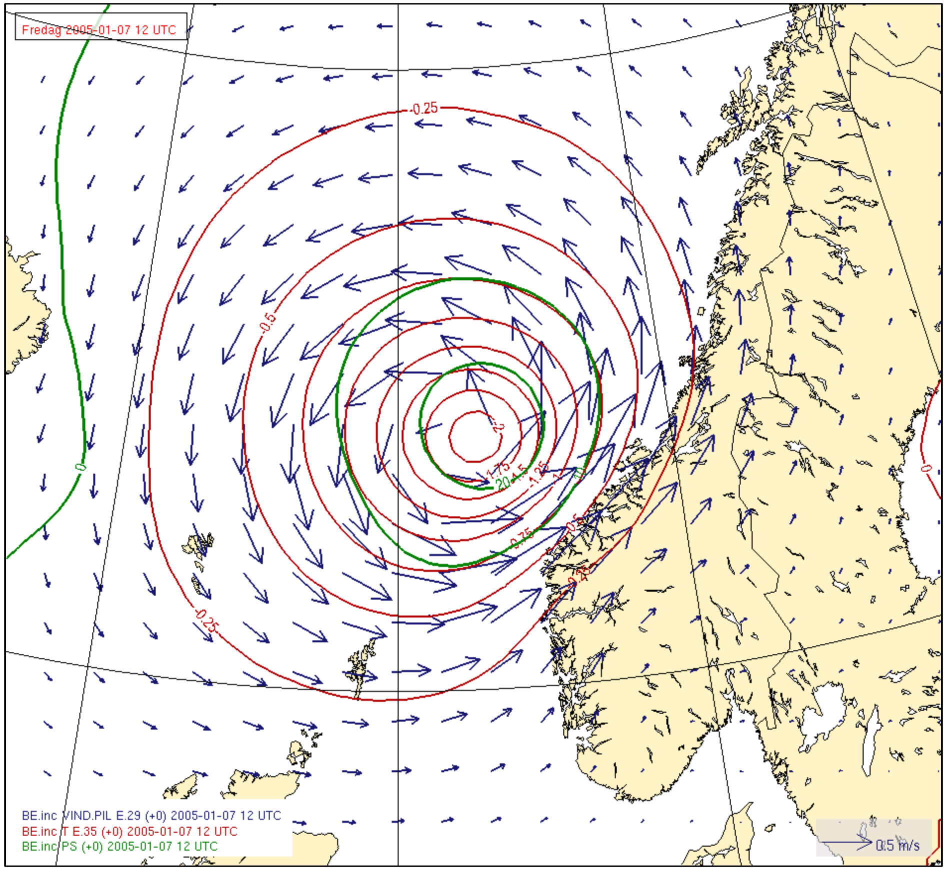
48 iterations



Increments of T, wind, ps

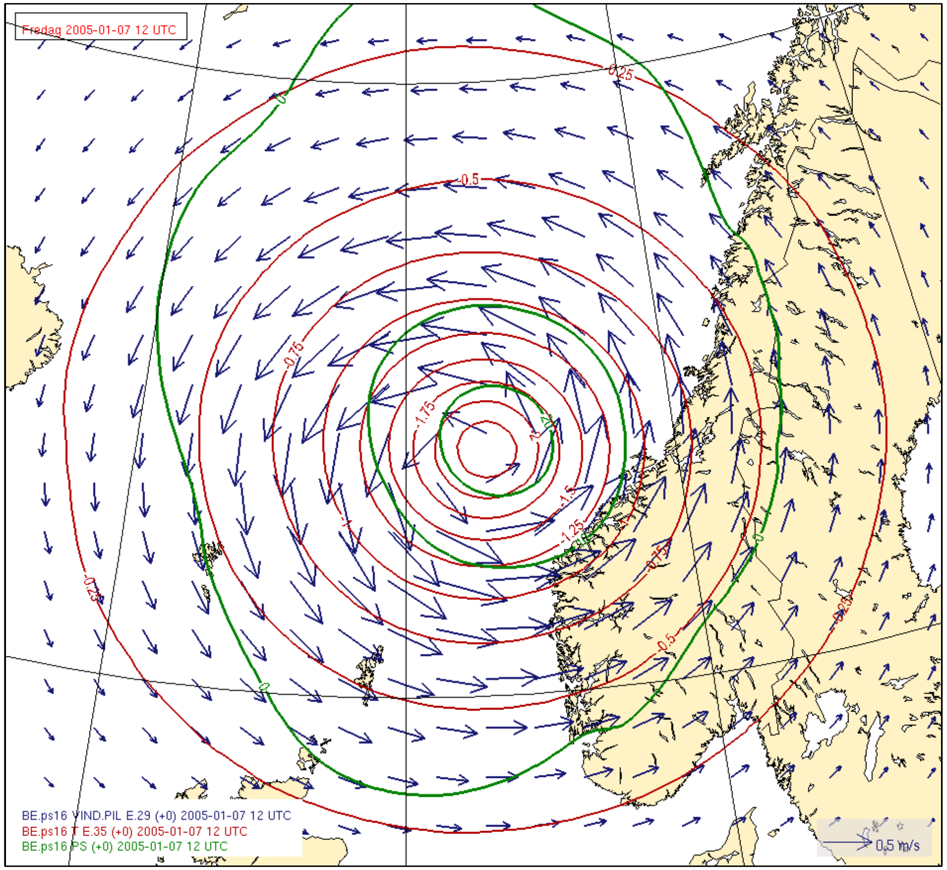
Weak constraints, weight = 10.

109 iterations



Weak constraints, dp/dx = dp/dy = 0.

24 iterations



Ps incr. with a full set of observations, 7/1-05 12z

Statistical balance, 22 iterations

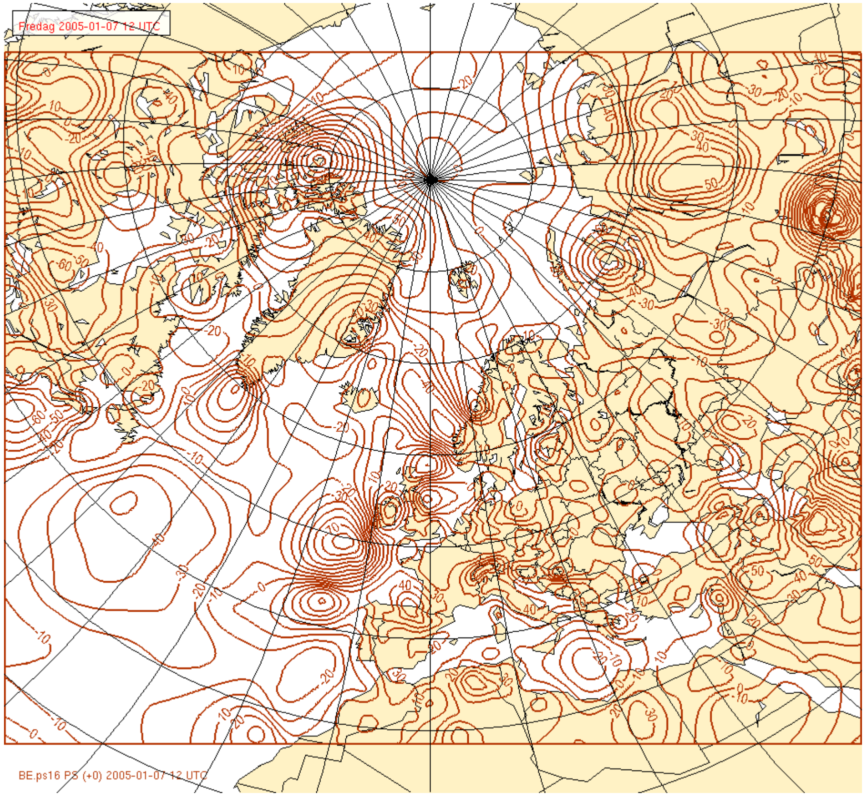
Jo = 13226

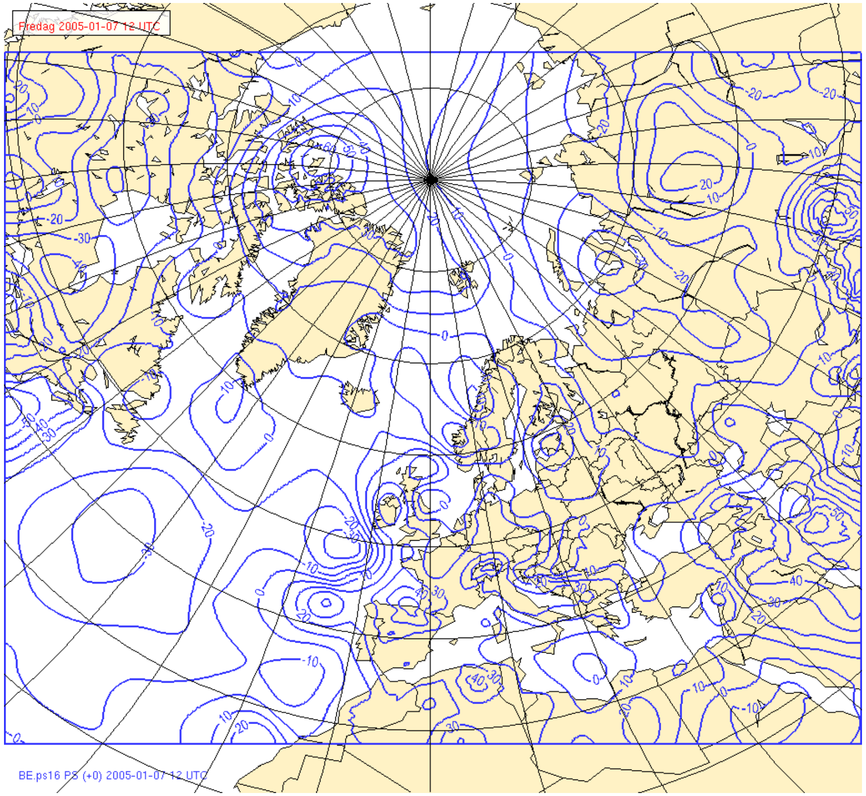
Jb = 1442

Weak constraints, weight 10., 112 iter.

Jo = 13877

Jb = 1146,   Jbe = 121





Conclusions and further plans

The balance term in the cost function destroys the effect of the preconditioning of the background term. We may need to develop preconditioning also on the Jbe term. If so, the circle is closed, such methods all rely on some form of approximate inversion.

Do an impact study to see if this path is worth pursuing further. It turned out to be more complicated than originally anticipated.

Thank you for your attention!

HIRVDA cost function, incremental formulation:

$$J(\delta\mathbf{x}) = \frac{1}{2}\delta\mathbf{x}^{\mathsf{T}}\mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}(\mathbf{y} - \mathbf{h}(\mathbf{x}_b) - \mathbf{H}\delta\mathbf{x})^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{h}(\mathbf{x}_b) - \mathbf{H}\delta\mathbf{x})$$

$\mathbf{h}$ (nonlinear) observation operator (may include model).
$\mathbf{H}$ Jacobian of $\mathbf{h}$ (may include tangent-linear model).

$\mathbf{B}$ is preconditioned by a linear transformation of variables:

$$\chi = \mathbf{U}\delta\mathbf{x}$$

Invertible transformation $(\delta\mathbf{x} = \mathbf{U}^{-1}\chi)$ that diagonalizes $\mathbf{B}$ (and $\mathbf{B}^{-1}$)

$$\mathbf{U}\mathbf{B}\mathbf{U}^{\mathsf{T}} = \mathbf{I} = (\mathbf{U}^{-1})^{\mathsf{T}}\mathbf{B}^{-1}\mathbf{U}^{-1}$$

Thus

$$J(\chi) = \frac{1}{2}\chi^{\mathsf{T}}\chi + \frac{1}{2}(\mathbf{d} - \mathbf{H}\mathbf{U}^{-1}\chi)^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\mathbf{U}^{-1}\chi)$$

Discrete p–level gradient (Tartu):

$$(G_x\phi)_{i+1/2,j,k} = \frac{1}{\overline{h_x}^x}\left[\delta_x\phi - \frac{\overline{(\delta_x p)\Delta_\eta\overline{\phi}^x}^\eta}{\overline{\Delta_\eta p}^x}\right]_{i+1/2,j,k}$$

$$(G_y\phi)_{i,j+1/2,k} = \frac{1}{\overline{h_y}^y}\left[\delta_y\phi - \frac{\overline{(\delta_y p)\Delta_\eta\overline{\phi}^y}^\eta}{\overline{\Delta_\eta p}^y}\right]_{i,j+1/2,k}$$

Discrete p–level divergence:

$$(D\cdot\mathbf{v})_{i,j,k} = \frac{1}{h_x h_y}\left[\delta_x(\overline{h_y}^x u) - \frac{\overline{\overline{h_y}^x(\Delta_\eta u)\delta_x p}^{x\eta}}{\Delta_\eta p} + \delta_y(\overline{h_x}^y v) - \frac{\overline{\overline{h_x}^y(\Delta_\eta v)\delta_y p}^{y\eta}}{\Delta_\eta p}\right]_{i,j,k}$$

To compute $A\Phi = \nabla_p^2 \Phi$:

```
SUBROUTINE DEL2P(PHI,APHI)
...
CALL HALOSWAP(PHI)
CALL GRADP(PHI,GX,GY)
CALL HALOSWAP(GX)
CALL HALOSWAP(GY)
CALL DIVP(GX,GY,APHI)
END
```