Preconditioning Saddle-Point Formulation of the variational data assimilation

M. Fisher[1], S. Gratton[2,3], S. Gürol[3]

[1]ECMWF, Reading, UK
[2]ENSEEIHT, Toulouse, France
[3]CERFACS, Toulouse, France

Colloque National sur l'Assimilation de données, Toulouse, France

1 December 2014

From Mike's presentation we have seen that:

- Incremental 4D-Var suffers from parallelization in the time dimension.
- Solution: Saddle-point approach

From Mike's presentation we have seen that:

- Incremental 4D-Var suffers from parallelization in the time dimension.
- Solution: Saddle-point approach

What will we discuss in this talk?

- Can we maintain good convergence properties of 4D-Var?
- Can we further accelerate the convergence rate?

Preconditioning of saddle point approach

Outline

- Saddle point approach of 4D-Var
- Preconditioning of saddle point formulation
- Numerical results
- Conclusions

Saddle Point Approach

- Let us consider weak-constraint 4D-Var as a constrained problem and write the Lagrangian function. Then the stationary point of $\mathcal{L}$ satisfies the system of equations that can be written in a matrix form as:

$$\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

- This system is called the saddle-point formulation of 4D-Var.
- $\mathbf{L} = \begin{pmatrix} I & & & & \\ -M_1 & I & & & \\ & -M_2 & I & & \\ & & \ddots & \ddots & \\ & & & -M_N & I \end{pmatrix}$ is an n-by-n matrix.
- $\mathbf{H} = diag(\mathbf{H_0}, \mathbf{H_1}, \ldots, \mathbf{H_N})$ is an n-by-m matrix.
- $\mathbf{D} = diag(\mathbf{B}, \mathbf{Q_1}, \ldots, \mathbf{Q_N})$ is an n-by-n matrix.
- $\mathbf{R} = diag(\mathbf{R_0}, \mathbf{R_1}, \ldots, \mathbf{R_N})$ is an m-by-m matrix.

Saddle Point Approach

- In matrix form:

$$\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

where $\mathcal{A}$ is a $(2n+m)$-by-$(2n+m)$ indefinite symmetric matrix.

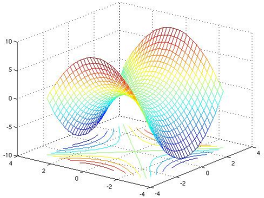
Saddle Point Approach

- In matrix form:

$$\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

  where $\mathcal{A}$ is a $(2n+m)$-by-$(2n+m)$ indefinite symmetric matrix.

- The solution of this problem is a saddle point



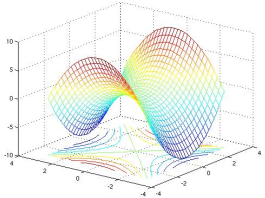
Saddle Point Approach

- In matrix form:

$$\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

  where $\mathcal{A}$ is a $(2n+m)$-by-$(2n+m)$ indefinite symmetric matrix.

- The solution of this problem is a saddle point



→ This approach is time-parallel.

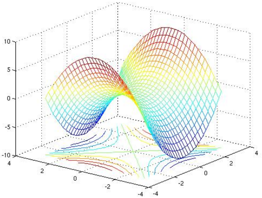
Saddle Point Approach

- In matrix form:

$$\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}$$

  where $\mathcal{A}$ is a $(2n+m)$-by-$(2n+m)$ indefinite symmetric matrix.

- The solution of this problem is a saddle point



→ This approach is time-parallel.

→ Solution algorithm: GMRES method with a preconditioner.

Outline

- Saddle point approach of 4D-Var
- Preconditioning of saddle point formulation
- Numerical results
- Conclusions

Preconditioning

- A preconditioner attempts to improve the spectral properties of the system matrix $\mathcal{A}$.

Preconditioning

- A preconditioner attempts to improve the spectral properties of the system matrix $\mathcal{A}$.
- When using GMRES, a clustered spectrum often results in rapid convergence, especially the departure from normality of the preconditioned matrix is not too high (Benzi et. al 2005).

Preconditioning

- A preconditioner attempts to improve the spectral properties of the system matrix $\mathcal{A}$.
- When using GMRES, a clustered spectrum often results in rapid convergence, especially the departure from normality of the preconditioned matrix is not too high (Benzi et. al 2005).

$\rightarrow$ When solving an indefinite saddle point system with GMRES, it is crucial to find an efficient preconditioner.

Preconditioning

- A preconditioner attempts to improve the spectral properties of the system matrix $\mathcal{A}$.
- When using GMRES, a clustered spectrum often results in rapid convergence, especially the departure from normality of the preconditioned matrix is not too high (Benzi et. al 2005).

$\rightarrow$ When solving an indefinite saddle point system with GMRES, it is crucial to find an efficient preconditioner.

Efficient preconditioner $\mathcal{P}$

- is an approximation to $\mathcal{A}$
- the cost of constructing and applying the preconditioner should be less than the gain in computational cost
- exploits the block structure of the problem for saddle point systems

Preconditioning

- A preconditioner attempts to improve the spectral properties of the system matrix $\mathcal{A}$.
- When using GMRES, a clustered spectrum often results in rapid convergence, especially the departure from normality of the preconditioned matrix is not too high (Benzi et. al 2005).

→ When solving an indefinite saddle point system with GMRES, it is crucial to find an efficient preconditioner.

Efficient preconditioner $\mathcal{P}$

- is an approximation to $\mathcal{A}$
- the cost of constructing and applying the preconditioner should be less than the gain in computational cost
- exploits the block structure of the problem for saddle point systems

Implementation

- Solving a system $\mathcal{A}\,\mathbf{u} = \mathbf{f}$ with a preconditioner $\mathcal{P}$ requires solving

$$(\mathcal{P}^{-1}\mathcal{A})\mathbf{u} = \mathcal{P}^{-\mathbf{1}}\mathbf{f}$$

Preconditioning Saddle Point Systems

$$\mathcal{A} = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix}$$

- Block preconditioners (Kuznetsov (1995), Murphy, Golub and Wathen (2000), Bramble and Pasciak (1988))

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S} \end{pmatrix}, \quad \mathcal{P} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{0} & \mathbf{S} \end{pmatrix}$$

  where $\mathbf{S} = \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathbf{T}}$ is the Schur complement (the unpreconditioned 4D-Var Hessian).

- Constraint preconditioners (Bergamaschi et. al (2004), Gould and Wathen (2000), Benzi et al. 2005)

$$\mathcal{P} = \begin{pmatrix} \widetilde{\mathbf{A}} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix}$$

  It is assumed that solving the system involving $\mathcal{P}$ is significantly easier than solving the original system.

- Hermitian and skew Hermitian splitting of $\mathcal{A}$, stationary iterative methods, multilevel methods, ... (Benzi et al (2005))

Preconditioning Saddle Point Formulation of 4D-Var

$$\mathcal{A} = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix}$$

- The inexact constraint preconditioner proposed by (Bergamaschi et. al. 2005) is promising for our application. The preconditioner can be chosen as:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \widetilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \widetilde{\mathbf{L}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

where

- $\widetilde{\mathbf{L}}$ is an approximation to the matrix $\mathbf{L}$
- $\widetilde{\mathbf{B}} = [\widetilde{\mathbf{L}}^{\mathrm{T}} \quad \mathbf{0}]$ is a full row rank approximation of the matrix $\mathbf{B} \in \mathbb{R}^{n \times (m+n)}$

Preconditioning Saddle Point Formulation of 4D-Var

$$\underbrace{\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}_k} \underbrace{\begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta \mathbf{x} \end{pmatrix}}_{\mathbf{u}} = \underbrace{\begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{f}_k}$$

When solving a sequence of saddle point systems, can we further improve the preconditioning for the outer loops $k > 1$?

Can we find low-rank updates for the inexact constraint preconditioner that approximates $\mathcal{A}^{-1}$ or its effect on a vector?

Preconditioning Saddle Point Formulation of 4D-Var

- For $k = 1$, we have the inexact constraint preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix}$$

Preconditioning Saddle Point Formulation of 4D-Var

- For $k = 1$, we have the inexact constraint preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix}$$

- For $k > 1$, we want to find a low-rank update $\Delta\mathbf{B} = \mathbf{B} - \widetilde{\mathbf{B}}$ and use the updated preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^{\mathrm{T}} \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix}$$

Preconditioning Saddle Point Formulation of 4D-Var

- For $k = 1$, we have the inexact constraint preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix}$$

- For $k > 1$, we want to find a low-rank update $\Delta\mathbf{B} = \mathbf{B} - \widetilde{\mathbf{B}}$ and use the updated preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^{\mathrm{T}} \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix}$$

$\longrightarrow$ GMRES performs matrix-vector products with $\mathcal{A}$:

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}_k} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}}_{\mathbf{u}_j^{(k)}} = \underbrace{\begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}}_{\mathbf{f}_j^{(k)}}$$

Preconditioning Saddle Point Formulation of 4D-Var

- For $k = 1$, we have the inexact constraint preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix}$$

- For $k > 1$, we want to find a low-rank update $\Delta\mathbf{B} = \mathbf{B} - \widetilde{\mathbf{B}}$ and use the updated preconditioner:

$$\mathcal{P} = \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^{\mathrm{T}} \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix}$$

$\rightarrow$ GMRES performs matrix-vector products with $\mathcal{A}$:

$$\underbrace{\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix}}_{\mathcal{A}_k} \underbrace{\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}}_{\mathbf{u}_j^{(k)}} = \underbrace{\begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}}_{\mathbf{f}_j^{(k)}}$$

$\rightarrow$ We can use the pairs $(\mathbf{u}_j^{(k)}, \mathbf{f}_j^{(k)})$ to find an update $\Delta\mathbf{B}$.

Preconditioning Saddle Point Formulation of 4D-Var

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^{\mathrm{T}} \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} \mathbf{A}\mathbf{u}_1 + \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{u}_2 \\ \widetilde{\mathbf{B}}\mathbf{u}_1 \end{pmatrix} + \begin{pmatrix} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 \\ \Delta\mathbf{B}\mathbf{u}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 \\ \Delta\mathbf{B}\mathbf{u}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{b} - \mathbf{A}\mathbf{u}_1 - \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{u}_2 \\ \mathbf{c} - \widetilde{\mathbf{B}}\mathbf{u}_1 \end{pmatrix}$$

Preconditioning Saddle Point Formulation of 4D-Var

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{B}}^{\mathrm{T}} \\ \widetilde{\mathbf{B}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^{\mathrm{T}} \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} \mathbf{A}\mathbf{u}_1 + \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{u}_2 \\ \widetilde{\mathbf{B}}\mathbf{u}_1 \end{pmatrix} + \begin{pmatrix} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 \\ \Delta\mathbf{B}\mathbf{u}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 \\ \Delta\mathbf{B}\mathbf{u}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{b} - \mathbf{A}\mathbf{u}_1 - \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{u}_2 \\ \mathbf{c} - \widetilde{\mathbf{B}}\mathbf{u}_1 \end{pmatrix}$$

- Let's define the vectors $\mathbf{r}_b$ and $\mathbf{r}_c$ as

$$\mathbf{r}_b = \mathbf{b} - \mathbf{A}\mathbf{u}_1 - \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{u}_2$$
$$\mathbf{r}_c = \mathbf{c} - \widetilde{\mathbf{B}}\mathbf{u}_1$$

- Then we have

$$\Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 = \mathbf{r}_b \tag{1}$$
$$\Delta\mathbf{B}\mathbf{u}_1 = \mathbf{r}_c \tag{2}$$

$\rightarrow$ We want to find an update $\Delta\mathbf{B}$ satisfying these equations.

Preconditioning Saddle Point Formulation of 4D-Var

- A rank-1 update to $\mathbf{\Delta B}$ can be given by

$$\boxed{\mathbf{\Delta B} = \alpha \mathbf{v}\mathbf{w}^{\mathrm{T}}}$$

where $\mathbf{v} \in \mathbb{R}^{n}$ and $\mathbf{w} \in \mathbb{R}^{n+m}$.

Preconditioning Saddle Point Formulation of 4D-Var

- A rank-1 update to $\Delta\mathbf{B}$ can be given by

$$\boxed{\Delta\mathbf{B} = \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}}}$$

where $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^{n+m}$.

- Substituting this relation into equations (1) and (2), we get

$$\begin{aligned} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 &= \mathbf{r}_b \quad \Leftrightarrow \quad \alpha\mathbf{w}\mathbf{v}^{\mathrm{T}}\mathbf{u}_2 = \mathbf{r}_b \\ \Delta\mathbf{B}\mathbf{u}_1 &= \mathbf{r}_c \quad \Leftrightarrow \quad \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}}\mathbf{u}_1 = \mathbf{r}_c \end{aligned}$$

from which we obtain that

$$\mathbf{w} = \mathbf{r}_b/\alpha\mathbf{v}^{\mathrm{T}}\mathbf{u}_2$$

$$\mathbf{v} = \mathbf{r}_c/\alpha\mathbf{w}^{\mathrm{T}}\mathbf{u}_1$$

Preconditioning Saddle Point Formulation of 4D-Var

- A rank-1 update to $\Delta\mathbf{B}$ can be given by

$$\boxed{\Delta\mathbf{B} = \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}}}$$

where $\mathbf{v} \in \mathbb{R}^n$ and $\mathbf{w} \in \mathbb{R}^{n+m}$.

- Substituting this relation into equations (1) and (2), we get

$$\begin{aligned} \Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 &= \mathbf{r}_b \quad \Leftrightarrow \quad & \alpha\mathbf{w}\mathbf{v}^{\mathrm{T}}\mathbf{u}_2 &= \mathbf{r}_b \\ \Delta\mathbf{B}\mathbf{u}_1 &= \mathbf{r}_c \quad \Leftrightarrow \quad & \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}}\mathbf{u}_1 &= \mathbf{r}_c \end{aligned}$$

from which we obtain that

$$\mathbf{w} = \mathbf{r}_b/\alpha\mathbf{v}^{\mathrm{T}}\mathbf{u}_2$$
$$\mathbf{v} = \mathbf{r}_c/\alpha\mathbf{w}^{\mathrm{T}}\mathbf{u}_1$$

- With the choice of

$$\boxed{\mathbf{w} = \mathbf{r}_b} \quad \text{and} \quad \boxed{\mathbf{v} = \mathbf{r}_c}$$

we can show that α is compatible and given as

$$\boxed{\alpha = 1/\mathbf{v}^{\mathrm{T}}\mathbf{u}_2 = 1/\mathbf{w}^{\mathrm{T}}\mathbf{u}_1}$$

Preconditioning Saddle Point Formulation of 4D-Var

- As a result, an inexact constraint preconditioner $\mathcal{P}$ can be updated from

$$\mathcal{P}_{j+1} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^T \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \alpha\mathbf{w}\mathbf{v}^{\mathrm{T}} \\ \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}} & \mathbf{0} \end{pmatrix},$$

where $\mathbf{w} = \mathbf{r}_b$, $\mathbf{v} = \mathbf{r}_c$ and $\alpha = 1/\mathbf{v}^{\mathrm{T}}\mathbf{u}_2$.

Preconditioning Saddle Point Formulation of 4D-Var

- As a result, an inexact constraint preconditioner $\mathcal{P}$ can be updated from

$$\mathcal{P}_{j+1} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^T \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \alpha\mathbf{w}\mathbf{v}^{\mathrm{T}} \\ \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}} & \mathbf{0} \end{pmatrix},$$

  where $\mathbf{w} = \mathbf{r}_b$, $\mathbf{v} = \mathbf{r}_c$ and $\alpha = 1/\mathbf{v}^{\mathrm{T}}\mathbf{u}_2$.

- We can rewrite this formula as

$$\mathcal{P}_{j+1} = \mathcal{P}_j + \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{w} \\ \mathbf{v} & \mathbf{0} \end{pmatrix}}_{\mathbf{F}} \underbrace{\begin{pmatrix} \alpha\mathbf{w}^T & \mathbf{0} \\ \mathbf{0} & \alpha\mathbf{v}^T \end{pmatrix}}_{\mathbf{G}}$$

  where $\mathbf{F}$ is an $(2n+m)$-by-2 matrix and $\mathbf{G}$ is an 2-by-$(2n+m)$ matrix.

Preconditioning Saddle Point Formulation of 4D-Var

- As a result, an inexact constraint preconditioner $\mathcal{P}$ can be updated from

$$\mathcal{P}_{j+1} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \Delta\mathbf{B}^T \\ \Delta\mathbf{B} & \mathbf{0} \end{pmatrix} = \mathcal{P}_j + \begin{pmatrix} \mathbf{0} & \alpha\mathbf{w}\mathbf{v}^{\mathrm{T}} \\ \alpha\mathbf{v}\mathbf{w}^{\mathrm{T}} & \mathbf{0} \end{pmatrix},$$

  where $\mathbf{w} = \mathbf{r}_b$, $\mathbf{v} = \mathbf{r}_c$ and $\alpha = 1/\mathbf{v}^{\mathrm{T}}\mathbf{u}_2$.

- We can rewrite this formula as

$$\mathcal{P}_{j+1} = \mathcal{P}_j + \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{w} \\ \mathbf{v} & \mathbf{0} \end{pmatrix}}_{\mathbf{F}} \underbrace{\begin{pmatrix} \alpha\mathbf{w}^T & \mathbf{0} \\ \mathbf{0} & \alpha\mathbf{v}^T \end{pmatrix}}_{\mathbf{G}}$$

  where $\mathbf{F}$ is an $(2n+m)$-by-2 matrix and $\mathbf{G}$ is an 2-by-$(2n+m)$ matrix.

- Using the Sherman-Morrison-Woodbury formula on this equation gives the inverse update as

$$\boxed{\mathcal{P}_{j+1}^{-1} = \mathcal{P}_j^{-1} - \mathcal{P}_j^{-1}\mathbf{F}(\mathbf{I}_2 + \mathbf{G}\mathcal{P}_j^{-1}\mathbf{F})^{-1}\mathbf{G}\mathcal{P}_j^{-1}}$$

Preconditioning Saddle Point Formulation of 4D-Var

- We have shown that it is possible to find a low-cost low-rank update for the inexact constraint preconditioner.

Preconditioning Saddle Point Formulation of 4D-Var

- We have shown that it is possible to find a low-cost low-rank update for the inexact constraint preconditioner.
- This update amounts to the two-sided-rank-one (TR1) update proposed by Griewank and Walther (2002).

TR1 update:

- It generalizes the classical symmetric rank-one update.
- It maintains the validity of all previous secant conditions.
- It is invariant with respect to linear transformations

Preconditioning Saddle Point Formulation of 4D-Var

- We have shown that it is possible to find a low-cost low-rank update for the inexact constraint preconditioner.
- This update amounts to the two-sided-rank-one (TR1) update proposed by Griewank and Walther (2002).

TR1 update:

- It generalizes the classical symmetric rank-one update.
- It maintains the validity of all previous secant conditions.
- It is invariant with respect to linear transformations
- It has no least change characterization in terms of any particular matrix norm.

Preconditioning Saddle Point Formulation of 4D-Var

- We have shown that it is possible to find a low-cost low-rank update for the inexact constraint preconditioner.
- This update amounts to the two-sided-rank-one (TR1) update proposed by Griewank and Walther (2002).

TR1 update:

- It generalizes the classical symmetric rank-one update.
- It maintains the validity of all previous secant conditions.
- It is invariant with respect to linear transformations
- It has no least change characterization in terms of any particular matrix norm.

→ Next slides are dedicated to find the least-Frobenius norm update.

Frobenius norm : $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}^2|}$

Preconditioning Saddle Point Formulation of 4D-Var

- Remember that we want to find an update such that

$$\Delta \mathbf{B}^{\mathrm{T}} \mathbf{u}_2 = \mathbf{r}_b \quad (1)$$

$$\Delta \mathbf{B}\, \mathbf{u}_1 = \mathbf{r}_c \quad (2)$$

Preconditioning Saddle Point Formulation of 4D-Var

- Remember that we want to find an update such that

$$\Delta\mathbf{B}^{\mathrm{T}}\mathbf{u}_2 = \mathbf{r}_b \qquad (1)$$

$$\Delta\mathbf{B}\,\mathbf{u}_1 = \mathbf{r}_c \qquad (2)$$

- Any solution $\Delta\mathbf{B}$ satisfying Equation (1) can be written as [Lemma 2.1](Sun 1999)

$$\Delta\mathbf{B}^{\mathrm{T}} = \mathbf{r}_b{\mathbf{u}_2}^{\dagger} + \mathbf{S}(\mathbf{I} - \mathbf{u}_2\mathbf{u}_2^{\dagger}),$$

where $\dagger$ denotes the pseudo-inverse and $\mathbf{S}$ is an $(n+m) \times n$ matrix. Inserting this relation into (2) yields

$${\mathbf{u}_2}^{\mathrm{T}\dagger}{\mathbf{r}_b}^{\mathrm{T}}\mathbf{u}_1 + (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger}{\mathbf{u}_2}^{\mathrm{T}})\mathbf{S}^{\mathrm{T}}\mathbf{u}_1 = \mathbf{r}_c.$$

Preconditioning Saddle Point Formulation of 4D-Var

- Remember that we want to find an update such that

$$\Delta \mathbf{B}^{\mathrm{T}} \mathbf{u}_2 = \mathbf{r}_b \quad (1)$$
$$\Delta \mathbf{B}\, \mathbf{u}_1 = \mathbf{r}_c \quad (2)$$

- Any solution $\Delta\mathbf{B}$ satisfying Equation (1) can be written as [Lemma 2.1](Sun 1999)

$$\Delta \mathbf{B}^{\mathrm{T}} = \mathbf{r}_b {\mathbf{u}_2}^{\dagger} + \mathbf{S}(\mathbf{I} - \mathbf{u}_2 \mathbf{u}_2^{\dagger}),$$

where $\dagger$ denotes the pseudo-inverse and $\mathbf{S}$ is an $(n+m) \times n$ matrix. Inserting this relation into (2) yields

$${\mathbf{u}_2}^{\mathrm{T}\dagger} {\mathbf{r}_b}^{\mathrm{T}} \mathbf{u}_1 + (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger} {\mathbf{u}_2}^{\mathrm{T}}) \mathbf{S}^{\mathrm{T}} \mathbf{u}_1 = \mathbf{r}_c .$$

- If this equation admits one solution, its least Frobenius norm solution,

$$\min_{\mathbf{S}^{\mathrm{T}} \in \mathbb{R}^{m \times n}} \|(\mathbf{r}_c - {\mathbf{u}_2}^{\mathrm{T}\dagger} \mathbf{r}_b^{\mathrm{T}} \mathbf{u}_1) - (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger} {\mathbf{u}_2}^{\mathrm{T}}) \mathbf{S}^{\mathrm{T}} \mathbf{u}_1\|_F ,$$

can be written as [Lemma 2.3](Sun 1999)

$$(\mathbf{S}^{\mathrm{T}})^* = (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger} {\mathbf{u}_2}^{\mathrm{T}})^{\dagger} (\mathbf{r}_c - {\mathbf{u}_2}^{\mathrm{T}\dagger} \mathbf{r}_b^{\mathrm{T}} \mathbf{u}_1) {\mathbf{u}_1}^{\dagger} .$$

Preconditioning Saddle Point Formulation of 4D-Var

- Substituting the solution for **S** into $\Delta\mathbf{B}$ yields that

$$\boxed{\Delta\mathbf{B}^* = {\mathbf{u}_2}^{\mathrm{T}\dagger}\mathbf{r}_b^{\mathrm{T}} + (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger}{\mathbf{u}_2}^{\mathrm{T}})\mathbf{r}_c{\mathbf{u}_1}^{\dagger}}$$

Preconditioning Saddle Point Formulation of 4D-Var

- Substituting the solution for **S** into $\Delta\mathbf{B}$ yields that

$$\boxed{\Delta\mathbf{B}^* = \mathbf{u}_2{}^{\mathrm{T}\dagger}\mathbf{r}_b^{\mathrm{T}} + (\mathbf{I} - \mathbf{u}_2{}^{\mathrm{T}\dagger}\mathbf{u}_2{}^{\mathrm{T}})\mathbf{r}_c\mathbf{u}_1{}^{\dagger}}$$

- This formula is not invariant with respect to linear transformations.

Preconditioning Saddle Point Formulation of 4D-Var

- Substituting the solution for **S** into $\Delta\mathbf{B}$ yields that

$$\boxed{\Delta\mathbf{B}^* = {\mathbf{u}_2}^{\mathrm{T}\dagger}\mathbf{r}_b^{\mathrm{T}} + (\mathbf{I} - {\mathbf{u}_2}^{\mathrm{T}\dagger}{\mathbf{u}_2}^{\mathrm{T}})\mathbf{r}_c{\mathbf{u}_1}^{\dagger}}$$

- This formula is not invariant with respect to linear transformations.
- We want to find such a formula by solving the following variational problem:

$$\min \|\mathbf{W}_1^{-1}\Delta\mathbf{B}\mathbf{W}_2^{-1}\|_F$$
$$\text{s.t. } \Delta\mathbf{B}^T\mathbf{u}_2 = \mathbf{r}_b$$
$$\Delta\mathbf{B}\,\mathbf{u}_1 = \mathbf{r}_c$$

where $\mathbf{W}_1$ is any symmetric positive definite matrix such that $\mathbf{W}_1\mathbf{W}_1^T\mathbf{u}_2 = \mathbf{c}$, and $\mathbf{W}_2$ is any symmetric positive definite matrix such that $\mathbf{W}_2^T\mathbf{W}_2\mathbf{u}_1 = \mathbf{b}$. For instance, $\mathbf{W}_1$ can be considered as $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ and $\mathbf{W}_2$ can be considered as $\mathbf{A}$.

Preconditioning Saddle Point Formulation of 4D-Var

- Substituting the solution for $\mathbf{S}$ into $\Delta\mathbf{B}$ yields that

$$\Delta\mathbf{B}^* = \mathbf{u}_2{}^{\mathrm{T}\dagger}\mathbf{r}_b^{\mathrm{T}} + (\mathbf{I} - \mathbf{u}_2{}^{\mathrm{T}\dagger}\mathbf{u}_2{}^{\mathrm{T}})\mathbf{r}_c\mathbf{u}_1{}^{\dagger}$$

- This formula is not invariant with respect to linear transformations.
- We want to find such a formula by solving the following variational problem:

$$\begin{aligned} \min \|\mathbf{W}_1^{-1}\Delta\mathbf{B}\mathbf{W}_2^{-1}\|_F \\ \text{s.t. } \Delta\mathbf{B}^T\mathbf{u}_2 = \mathbf{r}_b \\ \Delta\mathbf{B}\,\mathbf{u}_1 = \mathbf{r}_c \end{aligned}$$

where $\mathbf{W}_1$ is any symmetric positive definite matrix such that $\mathbf{W}_1\mathbf{W}_1^T\mathbf{u}_2 = \mathbf{c}$, and $\mathbf{W}_2$ is any symmetric positive definite matrix such that $\mathbf{W}_2^T\mathbf{W}_2\mathbf{u}_1 = \mathbf{b}$. For instance, $\mathbf{W}_1$ can be considered as $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ and $\mathbf{W}_2$ can be considered as $\mathbf{A}$.

- The solution is given as:

$$\Delta\mathbf{B}^* = \frac{\mathbf{c}\mathbf{r}_b^T}{\mathbf{u}_2^T\mathbf{c}} + \frac{\mathbf{r}_c\mathbf{b}^T}{\mathbf{u}_1^T\mathbf{b}} - \frac{\mathbf{u}_2^T\mathbf{r}_c\mathbf{c}\mathbf{b}^T}{\mathbf{u}_2^T\mathbf{c}\mathbf{u}_1^T\mathbf{b}}$$

- This formula can be rewritten as

$$\Delta \mathbf{B} = \begin{bmatrix} \frac{\mathbf{c}}{\mathbf{u}_2^T \mathbf{c}} & \mathbf{r_c} & -\frac{\mathbf{c}}{\mathbf{u}_2^T \mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{r_b}^T \\ \\ \frac{\mathbf{b}^T}{\mathbf{u}_1^T \mathbf{b}} \\ \\ \frac{\mathbf{u}_2{}^T \mathbf{r}_c \mathbf{b}^T}{\mathbf{u}_1^T \mathbf{b}} \end{bmatrix} = \mathbf{V}\mathbf{W}^{\mathrm{T}},$$

- This formula can be rewritten as

$$\Delta \mathbf{B} = \begin{bmatrix} \frac{\mathbf{c}}{\mathbf{u}_2^T \mathbf{c}} & \mathbf{r_c} & -\frac{\mathbf{c}}{\mathbf{u}_2^T \mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{r_b}^T \\ \\ \frac{\mathbf{b}^T}{\mathbf{u}_1^T \mathbf{b}} \\ \\ \frac{\mathbf{u_2}^T \mathbf{r}_c \mathbf{b}^T}{\mathbf{u}_1^T \mathbf{b}} \end{bmatrix} = \mathbf{V}\mathbf{W}^{\mathrm{T}},$$

- The preconditioner can be updated by using the following formula

$$\mathcal{P}_1 = \mathcal{P}_0 + \begin{pmatrix} \mathbf{0} & \mathbf{W}\mathbf{V}^T \\ \mathbf{V}\mathbf{W}^T & \mathbf{0} \end{pmatrix} = \mathcal{P}_0 + \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{V} & \mathbf{0} \end{pmatrix}}_{\mathbf{F}} \underbrace{\begin{pmatrix} \mathbf{W}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^T \end{pmatrix}}_{\mathbf{G}}$$

- The inverse formula is then given by

$$\boxed{\mathcal{P}_F^{-1} = \mathcal{P}_0^{-1} - \mathcal{P}_0^{-1}\mathbf{F}(\mathbf{I}_6 + \mathbf{G}\mathcal{P}_0^{-1}\mathbf{F})^{-1}\mathbf{G}\mathcal{P}_0^{-1}}$$

where $\mathbf{F}$ is an $(2n+m)$-by-6 matrix and $\mathbf{G}$ is an 6-by-$(2n+m)$ matrix.

Numerical Results

Implementation platform

- We used the Object Oriented Prediction System (OOPS) developed by ECMWF
- OOPS consists of simplified models of a real-system

The model

- It is a two-layer quasi-geostraphic model with 1600 grid-points

Implementation details

- There are 100 observations of stream function every 3 hours, 100 wind observations plus 100 wind-speed observations every 6 hours
- The error covariance matrices are assumed to be horizontally isotropic and homogeneous, with Gaussian spatial structure
- The analysis window is 24 hours, and is divided into 8 subwindows
- 3 outer loops with 10 inner loops each are performed

Methods

1. Standard weak-constrained 4D-Var formulation
   $\longrightarrow$ Solution method is preconditioned conjugate-gradients

Methods

1. Standard weak-constrained 4D-Var formulation
   $\longrightarrow$ Solution method is preconditioned conjugate-gradients
2. Saddle point formulation with an updated inexact constraint preconditioner
   $\longrightarrow$ Solution method is GMRES
   $\longrightarrow$ The initial preconditioner is chosen as

$$\mathcal{P}_0 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \widetilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \widetilde{\mathbf{L}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \text{with} \quad \widetilde{\mathbf{L}} = \begin{pmatrix} \mathbf{I} & & & \\ -\mathbf{I} & \mathbf{I} & & \\ & \ddots & \ddots & \\ & & -\mathbf{I} & \mathbf{I} \end{pmatrix}.$$

Methods

1. Standard weak-constrained 4D-Var formulation
   → Solution method is preconditioned conjugate-gradients
2. Saddle point formulation with an updated inexact constraint preconditioner
   → Solution method is GMRES
   → The initial preconditioner is chosen as

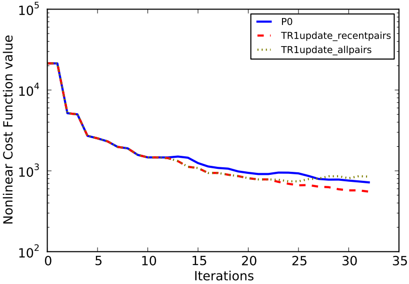
$$\mathcal{P}_0 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \widetilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \widetilde{\mathbf{L}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \text{with} \quad \widetilde{\mathbf{L}} = \begin{pmatrix} \mathbf{I} & & & \\ -\mathbf{I} & \mathbf{I} & & \\ & \ddots & \ddots & \\ & & -\mathbf{I} & \mathbf{I} \end{pmatrix}.$$

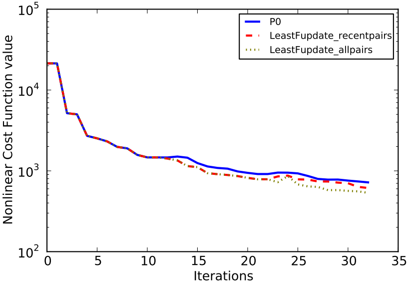
$$\mathcal{P}_0^{-1} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{L}}^{-\mathrm{T}} \\ \mathbf{0} & \mathbf{R}^{-1} & \mathbf{0} \\ \widetilde{\mathbf{L}}^{-1} & \mathbf{0} & -\widetilde{\mathbf{L}}^{-1}\mathbf{D}\widetilde{\mathbf{L}}^{-\mathrm{T}} \end{pmatrix} \quad \text{and} \quad \widetilde{\mathbf{L}}^{-1} = \begin{pmatrix} \mathbf{I} & & & \\ \mathbf{I} & \mathbf{I} & & \\ \vdots & \ddots & \ddots & \\ \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} \end{pmatrix}.$$

Second-level preconditioners:

1. $\mathcal{T}_k$: The preconditioner obtained by using the TR1 update
2. $\mathcal{F}_k$: The preconditioner obtained by using the least-Frobenius update

The performance of the second level preconditioners





- Second-level preconditioners obtained by using updates may accelerate the convergence
- When all pairs are used the least-Frobenius update is more stable.

Overall performance compared with the standard 4DVar formulation

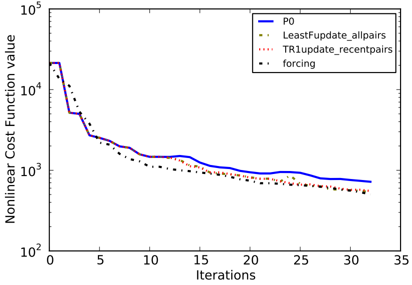


Figure: Nonlinear cost function values along iterations

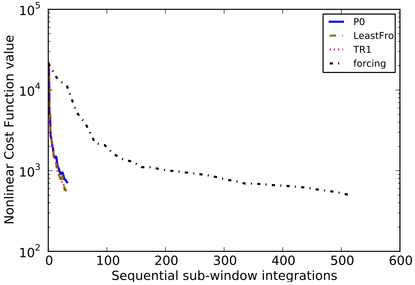


Figure: Nonlinear cost function values along sequential subwindow integrations

- At each iteration the standard 4DVar formulation requires one application of $\mathbf{L}^{-1}$, followed by one application of $\mathbf{L}^{-\mathrm{T}}$ (16 sequential subwindow integrations)
- At each iteration of saddle point formulation require one subwindow integration (provided that $\mathbf{L}^{-1}$ and $\mathbf{L}^{-\mathrm{T}}$ are applied simultaneously)

Conclusions

- The saddle point formulation of weak-constraint 4D-Var allows parallelisation in the time dimension.
- Finding an effective preconditioner is a key issue in solving the saddle point systems.
- The inexact constraint preconditioner can be used to precondition the saddle point formulation of 4D-Var.
- When solving a sequence of saddle point systems, a low-rank low-cost update formulas can be found to further improve preconditioning.
- The preconditioned GMRES algorithm for saddle point formulation is competitive with the existing algorithms and has the potential to allow 4D-Var to remain computationally viable on next-generation computer architectures.

Thank you for your attention !