A primal-dual approach of weak-constrained variational data assimilation (Iterate) History matters

Seminar in the Optimization Division University of Linköping

Serge Gratton,
E. Simon, Ph.L. Toint, S. Gurol

University of Toulouse, INPT-IRIT

August 2017

Outline

In forecasting problems, a dynamical system

$$\begin{cases} \frac{\partial u}{\partial t} = f(t, u) \\ u(t_0) = u_0 \end{cases}$$

involves a nonlinear differential operator f.

Vector u consists of state variables, e.g.

- velocity components
- pressure
- density
- temperature
- gravitational potential

Goal : predict the state of the system at a future time from

- dynamical integration model
- observational data are very often needed

Applications : climate, meteorology, oceanography, neutronics, finance, ...

The dynamical integration model predicts the state of the system given the (initial) state at an earlier time.

$\longrightarrow$ integrating may lead to very large prediction errors
(inexact physics, discretization errors, approximated parameters)

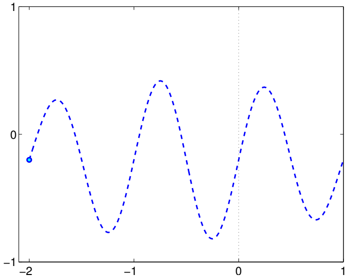
Observational data are used to improve accuracy of the forecasts.

$\longrightarrow$ but the data are inaccurate (measurement noise, under-sampling)

$\longrightarrow$ 10^7 observations (10^9 variables) processed every day : structured big data problem

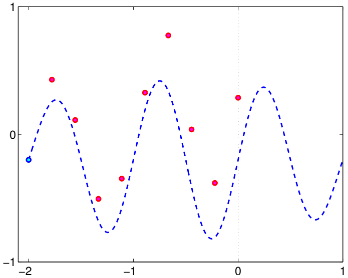
$\longrightarrow$ Need to be solved within a prescribed CPU time on a parallel computer

Data assimilation chart



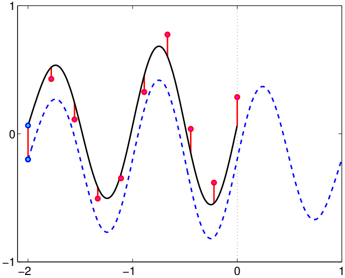
- starting from a priori knowledge on the state, the forward model is run : expensive in computer time, not always very parallel

Data assimilation chart



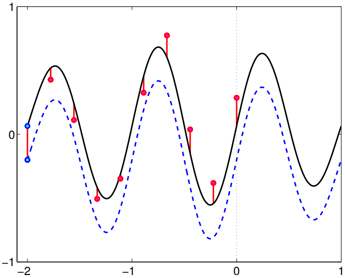
- starting from a priori knowledge on the state, the forward model is run : expensive in computer time, not always very parallel
- data are processed : screened, agglomerated

Data assimilation chart



- starting from a priori knowledge on the state, the forward model is run : expensive in computer time, not always very parallel
- data are processed : screened, agglomerated
- adjustment with of model with respect to observations : best value to be found by some "form of optimization"

Data assimilation chart



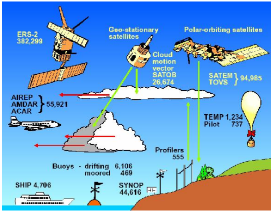
- starting from a priori knowledge on the state, the forward model is run : expensive in computer time, not always very parallel
- data are processed : screened, agglomerated
- adjustment with of model with respect to observations : best value to be found by some "form of optimization"
- predictions are then issued

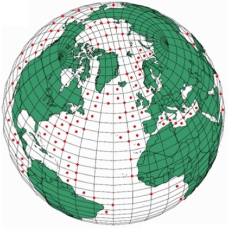
Solve a large-scale non-linear weighted least-squares problem :

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N} \|\mathcal{H}_j(\mathcal{M}_j(x)) - y_j\|^2_{R_j^{-1}}$$

where

- $x \equiv x(t_0)$ is the control variable in $\mathbb{R}^n$, $n \sim 10^9$
- $\mathcal{M}_j$ are model operators : $x(t_j) = \mathcal{M}_j(x(t_0))$
- $\mathcal{H}_j$ are observation operators : $y_j \approx \mathcal{H}_j(x(t_j))$ in $\mathbb{R}^n$, $n \sim 10^7$
- the observations y_j and the background x_b are noisy
- B and R_j are covariance matrices
- No model error here : the dynamical system is supposed to be known exactly





Most popular solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem :

$$\min_{x\in\mathbb{R}^n} J(x) = \frac{1}{2}||x - x_b||^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N}||\mathcal{H}_j(\mathcal{M}_{0,j}(x)) - y_j||^2_{R_j^{-1}}$$

Typically solved by a standard **Gauss-Newton method** known as **Incremental 4D-Var** in data assimilation community (series of paper by Courtier, Talagrand)

Most popular solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem :

$$\min_{x\in\mathbb{R}^n} J(x) = \frac{1}{2}||x - x_b||^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N} ||\mathcal{H}_j(\mathcal{M}_{0,j}(x)) - y_j||^2_{R_j^{-1}}$$

Typically solved by a standard **Gauss-Newton method** known as **Incremental 4D-Var** in data assimilation community (series of paper by Courtier, Talagrand)

1. Solve the **linearized subproblem** at iteration k

$$\boxed{\min_{\delta x_k\in\mathbb{R}^n} J(\delta x_k) = \tfrac{1}{2}\|\delta x_k - x_b + x_k\|^2_{B^{-1}} + \tfrac{1}{2}\|H_k\delta x_k - d_k\|^2_{R^{-1}}}$$

Most popular solution algorithm

→ **Large-scale regularized nonlinear least-squares** problem :

$$\min_{x \in \mathbb{R}^n} J(x) = \frac{1}{2}||x - x_b||^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N} ||\mathcal{H}_j(\mathcal{M}_{0,j}(x)) - y_j||^2_{R_j^{-1}}$$

Typically solved by a standard **Gauss-Newton method** known as **Incremental 4D-Var** in data assimilation community (series of paper by Courtier, Talagrand)

1. Solve the **linearized subproblem** at iteration k

$$\boxed{\min_{\delta x_k \in \mathbb{R}^n} J(\delta x_k) = \frac{1}{2}||\delta x_k - x_b + x_k||^2_{B^{-1}} + \frac{1}{2}||H_k \delta x_k - d_k||^2_{R^{-1}}}$$

2. Perform update $x_{k+1} = x_k + \delta x_k$

Structure of the linearized problem and "dual approach"

- The **exact solution** is either

$$x_b - x_k + \underbrace{(B^{-1} + H_k^T R^{-1} H_k)^{-1} H_k^T R^{-1} (d_k - H_k(x_b - x_k))}_{\text{linear system in } \mathbb{R}^n}$$

or, by duality with respect to the observation term

$$x_b - x_k + BH_k^T \underbrace{(R + H_k B H_k^T)^{-1} (d_k - H_k(x_b - x_k))}_{\text{Lagrange mult. : requires solving a linear system in } \mathbb{R}^m}$$

- These equations are the heart of most data assimilation systems
- When solved directly they are considered as impractical in large scale systems
- Dual form is more than appealing for regularized under-determined systems (10^7 observations but 10^9 variables)

Iterative (primal) approach

Iterative minimization

1. **Iteratively** solve with PCG

$$(B^{-1} + H_k^T R^{-1} H_k) s_k = H_k^T R^{-1} (d_k - H_k(x_b - x_k))$$

2. **Set** $\delta x_k = x_b - x_k + s_k$

- A good preconditionner is B
- It is possible to derive to prove convergence with approximate solution of the linear system. But
  - Repelling fixed points may exist (different from Newton's method) !
  - Step-size control enables local convergence : trust-region, linesearch
  - Truncated iterative linear algebra methods are essential
  - Preconditioning is crucial for an acceptable (inner-)iteration count, i.e. controlled computational time

The "dual approach"

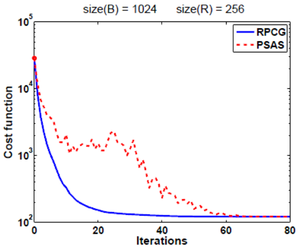
Iterative minimization

1. **Iteratively** solve

$$(R + H_k B H_k^T)\lambda_k = d_k - H_k(x_b - x_k)$$

2. **Set** $\delta x_k = x_b - x_k + BH_k^T\lambda_k$

- A good preconditioner is R^{-1}
- Non monotonic function values along iterations for the dual
- The effect of truncation may be catastrophic in the dual solvers
- This weakness of the method can be completely overcome by change of scalar product in dual CG (G., Tshimanga 2009)



Restricted PCG (version 1)

Initialization

$\lambda_0 = 0$, $\widehat{r}_0 = R^{-1}(d - H(x_b - x))$, $\widehat{z}_0 = G\widehat{r}_0$,
$\widehat{p}_1 = \widehat{z}_0$, $k = 1$

Loop on k

1. $\widehat{q}_i = \widehat{A}\widehat{p}_i$
2. $\alpha_i = < \widehat{r}_{i-1}, \widehat{z}_{i-1} >_M / < \widehat{q}_i, \widehat{p}_i >_M$
3. $\lambda_i = \lambda_{i-1} + \alpha_i \widehat{p}_i$
4. $\widehat{r}_i = \widehat{r}_{i-1} - \alpha_i \widehat{q}_i$
5. $\beta_i = < \widehat{r}_{i-1}, \widehat{z}_{i-1} >_M / < \widehat{r}_{i-2}, \widehat{z}_{i-2} >_M$
6. $\widehat{z}_i = G\widehat{r}_i$
7. $\widehat{p}_i = \widehat{z}_{i-1} + \beta_i \widehat{p}_{i-1}$

- $\widehat{A} = I_m + R^{-1}HBH^T$
- G is the preconditioner.
- M is the inner-product.
- RPCG Algorithm : $M = HBH^T$ lead to a mathematically equivalent algorithm to the primal one preconditioned by F : preserves monotonic decrease of quadratic cost
- $BH^{\mathrm{T}}G = FH^{\mathrm{T}}$: G should be symmetric w.r.t. to M
- B^{-1} not involved

Restricted PCG (version 2)

Initialization steps

Loop : WHILE

1. $\widehat{q}_{i-1} = R^{-1} t_{i-1} + \widehat{p}_{i-1}$
2. $\alpha_{i-1} = w_{i-1}^{\mathrm{T}} \widehat{r}_{i-1} \,/\, \widehat{q}_{i-1}^{\mathrm{T}} t_{i-1}$
3. $\widehat{v}_i = \widehat{v}_{i-1} + \alpha_{i-1} \widehat{p}_{i-1}$
4. $\widehat{r}_i = \widehat{r}_{i-1} + \alpha_{i-1} \widehat{q}_{i-1}$
5. $\widehat{z}_i = G \widehat{r}_i$
6. $w_i = HBH^{\mathrm{T}} \widehat{z}_i$
7. $\beta_i = w_i^{\mathrm{T}} \widehat{r}_i \,/\, w_{i-1}^{\mathrm{T}} \widehat{r}_{i-1}$
8. $\widehat{p}_i = -\widehat{z}_i + \beta_i \widehat{p}_{i-1}$
9. $t_i = -w_i + \beta_i t_{i-1}$

Explanation

Theorem

Let

1. *F primal, G dual preconditioner. Suppose $BH^{\mathrm{T}}G = FH^{\mathrm{T}}$.*
2. $v_0 = x^b - x_0$.

Primal CG vectors write

$$r_i = H^{\mathrm{T}}\widehat{r}_i, p_i = BH^{\mathrm{T}}\widehat{p}_i, q_i = H^{\mathrm{T}}\widehat{q}_i, \ldots$$

Note that For "exact" preconditioners

$$BH^{\mathrm{T}}\left(I + R^{-1}HBH^{\mathrm{T}}\right)^{-1} = \left(B^{-1} + H^{\mathrm{T}}R^{-1}H\right)^{-1}H^{\mathrm{T}}$$

"With-hat" quantities can be generated by CG on the dual system with inner-product HBH^T

The method is parallel. It however offers limited parallelism not enough for modern computers.

The advent of a new problem stimulates new developments...

Weak-constraint 4D-Var

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|x_0 - x_b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\|\mathcal{H}_j(x_j) - y_j\|^2_{\mathbf{R}_j^{-1}} + \frac{1}{2}\sum_{j=1}^{N}\|\underbrace{x_j - \mathcal{M}_j(x_{j-1})}_{q_j}\|^2_{\mathbf{Q}_j^{-1}}$$

- $\mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_N \end{pmatrix} \in \mathbb{R}^n$ is the control variable (with $x_j = x(t_j)$)
- x_b is the background given at the initial time (t_0).
- $y_j \in \mathbb{R}^{m_j}$ is the observation vector over a given time interval
- $\mathcal{H}_j$ maps the state vector x_j from model space to observation space
- $\mathcal{M}_j$ represents an integration of the numerical model from time t_{j-1} to t_j
- B, R_j and Q_j are the covariance matrices of background, observation and model error. B and Q_j impractical to "invert"

We can work with longer time windows, accumulate more observations, forget the influence of the regularization term, but larger problems

The linearized subproblem (inner loop)

- The linearized problem at the k-th outer loop is given by

$$\boxed{\min_{\delta \mathbf{x}} \; \frac{1}{2}\|\delta x_0 - b^{(k)}\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\left\|H_j^{(k)}\delta x_j - d_j^{(k)}\right\|^2_{\mathbf{R}_j^{-1}} + \frac{1}{2}\sum_{j=1}^{N}\|\underbrace{\delta x_j - M_j^{(k)}\delta x_{j-1}}_{\delta q_j} - c_j^{(k)}\|^2_{\mathbf{Q}_j^{-1}}}$$

- $\boldsymbol{\delta}\mathbf{x} = \begin{pmatrix} \delta x_0 \\ \delta x_1 \\ \vdots \\ \delta x_N \end{pmatrix} \in \mathbb{R}^n$ is the increment.

- The vectors $b^{(k)}$, $c_j^{(k)}$ and $d_j^{(k)}$ are defined by

$$\begin{aligned} b^{(k)} &= x_b - {x_0}^{(k)} \\ c_j^{(k)} &= q_j^{(k)} \\ d_j^{(k)} &= \mathcal{H}_j({x_j}^{(k)}) - y_j \end{aligned}$$

and are calculated at the outer loop.

Rewriting the linearized subproblem

$$\boxed{\min_{\delta \mathbf{x} \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{L}\delta\mathbf{x} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta\mathbf{x} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}}$$

where

- $\mathbf{L} = \begin{pmatrix} I & & & & \\ -M_1 & I & & & \\ & -M_2 & I & & \\ & & \ddots & \ddots & \\ & & & -M_N & I \end{pmatrix}$
- $\mathbf{d} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_N \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} b \\ c_1 \\ \vdots \\ c_N \end{pmatrix}$
- $\mathbf{H} = diag(\mathbf{H_0}, \mathbf{H_1}, \ldots, \mathbf{H_N})$
- $\mathbf{D} = diag(\mathbf{B}, \mathbf{Q_1}, \ldots, \mathbf{Q_N})$ and $\mathbf{R} = diag(\mathbf{R_0}, \mathbf{R_1}, \ldots, \mathbf{R_N})$

Rewriting the linearized subproblem

$$\boxed{\min_{\delta\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{L}\delta\mathbf{x}-\mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta\mathbf{x}-\mathbf{d}\|^2_{\mathbf{R}^{-1}} = q_{st}(\delta\mathbf{x})}$$

- $\mathbf{L}\delta\mathbf{x} = \begin{pmatrix} I & & & & \\ -M_1 & I & & & \\ & -M_2 & I & & \\ & & \ddots & \ddots & \\ & & & -M_N & I \end{pmatrix} \begin{pmatrix} \delta x_0 \\ \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_N \end{pmatrix} = \begin{pmatrix} \delta x_0 \\ \delta x_1 - M_1\delta x_0 \\ \delta x_2 - M_2\delta x_1 \\ \vdots \\ \delta x_N - M_N\delta x_{N-1} \end{pmatrix}$

- Matrix-vector products with $\mathbf{L}$ can be parallelized in the time dimension

Rewriting the linearized subproblem

- Making change of variables

$$\boxed{\delta \mathbf{p} = \mathbf{L}\delta \mathbf{x}}$$

the subproblem can also be rewritten as

$$\boxed{\min_{\delta \mathbf{p} \in \mathbb{R}^n} \frac{1}{2}\|\delta \mathbf{p} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\mathbf{L}^{-1}\delta \mathbf{p} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}}$$

- $\delta \mathbf{x} = \mathbf{L}^{-1}\delta \mathbf{p}$ is sequential $\rightarrow \delta x_j = M_j \delta x_{j-1} + \delta q_j$

The linearized subproblems

State Formulation

$$\min_{\delta \mathbf{x}} \frac{1}{2}\|\mathbf{L}\delta \mathbf{x} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta \mathbf{x} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$$

- Matrix-vector products with $\mathbf{L}$ can be parallelized in the time dimension.
- Solution algorithm : Preconditioned Lanczos or PCG type methods.
- Preconditioning is difficult since

$$\mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-\mathbf{T}}(\mathbf{L}^{\mathbf{T}}\mathbf{D}^{-1}\mathbf{L})\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}$$

  can be ill-conditioned depending on the accuracy of $\widetilde{\mathbf{L}}^{-\mathbf{1}}$.

Forcing Formulation

$$\min_{\delta \mathbf{p}} \frac{1}{2}\|\delta \mathbf{p} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\mathbf{L}^{-\mathbf{1}}\delta \mathbf{p} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$$

- Matrix-vector products with $\mathbf{L}^{-\mathbf{1}}$ is sequential.
- Solution algorithm : Preconditioned Lanczos or PCG type methods.
- Preconditioning is straightforward. The structure is similar to the strong-constraint case.

Inverse of convariance matrices involved : expensive operation for new systems, where these matrices are sums of matrices

Saddle Point Approach

- Let us consider weak-constraint 4D-Var as a constrained problem :

$$\min_{(\delta \mathbf{p}, \delta \mathbf{w})} \frac{1}{2}\|\delta \mathbf{p} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\delta \mathbf{w} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$$
$$\text{subject to} \quad \delta \mathbf{p} = \mathbf{L}\delta \mathbf{x} \quad \text{and} \quad \delta w = \mathbf{H}\delta \mathbf{x}$$

- We can write the *Lagrangian function* for this problem as

$$\begin{aligned}\mathcal{L}(\delta w, \delta p, \boldsymbol{\lambda}, \boldsymbol{\mu}) = & \frac{1}{2}\|\delta \mathbf{p} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\delta \mathbf{w} - \mathbf{d}\|^2_{\mathbf{R}^{-1}} \\ & + \boldsymbol{\lambda}^T(\delta p - \mathbf{L}\delta x) + \boldsymbol{\mu}^T(\delta w - \mathbf{H}\delta x)\end{aligned}$$

- The stationary point of $\mathcal{L}$ satisfies the following equations :

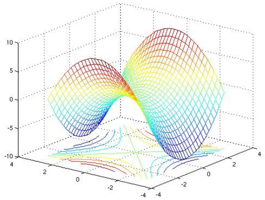
$$\begin{aligned}\mathbf{D}^{-1}(\mathbf{L}\delta \mathbf{x} - \mathbf{b}) + \boldsymbol{\lambda} &= 0 \\ \mathbf{R}^{-1}(\mathbf{H}\delta \mathbf{x} - \mathbf{d}) + \boldsymbol{\mu} &= 0 \\ \mathbf{L}^T\boldsymbol{\lambda} + \mathbf{H}^T\boldsymbol{\mu} &= 0\end{aligned}$$

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, with no inverse of covariance matrix involved



→ Solution algorithm : iterative method (MINRES, GMRES, ...) with a preconditioner.

The original Saddle method : M. Fisher

Consider the solution of the subproblem

$$r(\delta\lambda, \delta\mu, \delta x) = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\lambda \\ \delta\mu \\ \delta x \end{pmatrix} = 0$$

Saddle-original (SAQ0)

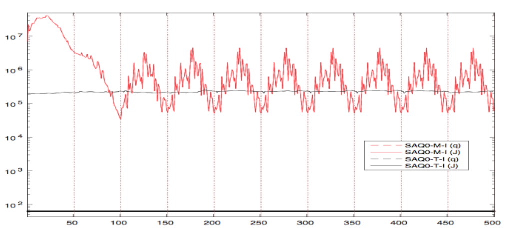
While not converged :

1. **Compute** $J(x_k)$ and $g_k = \nabla_x J(x_k)$
2. **Apply** the preconditioned GMRES algorithm to solve the system $r(\delta\lambda, \delta\mu, \delta x) = 0$. Terminate the iterations if $\|r(\delta\lambda, \delta\mu, \delta x)\| \leq \varepsilon_r(\|b\| + \|d\|)$ or $j = n_{inner}$ to yield δx_k
3. **Set** $\delta x_{k+1} = x_k + \delta x_k$

Possible preconditioners, $S = \tilde{\mathbf{L}}^{\mathrm{T}}\mathbf{D}^{-1}\tilde{\mathbf{L}}$, $\tilde{\mathbf{L}} \sim \mathbf{L}$ (square, nonsingular),

$$P_M = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \tilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \tilde{\mathbf{L}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad P_B = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{S}^{-1} \end{pmatrix}, \quad P_T = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \tilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{0} & \mathbf{0} & -\mathbf{S}^{-1} \end{pmatrix}$$

The original Saddle method



1. We choose $M = I$ in the preconditionner $\widetilde{L}$. We represent the original nonlinear least-square function J, its GN approximation q_{st}
2. None of the method reduces J significantly. The method with P_M diverges slightly
3. The curve for q_{st} and J are noticeably the same. The problem nonlinearity cannot be blamed
4. The non-monotonic behaviour of q_{st} and J is obvious. Stopping cannot solely rely on maximum number of iterations

A better stopping criterion for GMRES

Saddle-globalized (SAQ1)

While not converged :

1. **Compute** $J(x_k)$ and $g_k = \nabla_x J(x_k)$
2. **Apply** the preconditioned GMRES algorithm to solve the system $r(\delta\lambda, \delta\mu, \delta x) = 0$. Terminate at iteration j if $q_{st}(0) - q_{st}(\delta x) \geq \max\left(\varepsilon_q \min\left(1, \|g_k\|^2\right), \theta_j\right)$ to yield δx_k
3. Perform a backtracking linesearch minimisation of J along δx_k yielding a step-length α_k
4. **Set** $\delta x_{k+1} = x_k + \alpha_k \delta x_k$

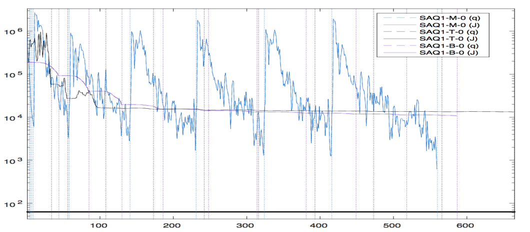
- The sequence θ_j goes to zero and forces GMRES not to stop prematurely
- The stopping criterion involves the computation of the quadratic : one needs to apply L, L^{-1}, H, R^{-1}
- The GMRES algorithm may need more iterations than previsously, making the GMRES calls potentially more expensive

Saddle globalized

Remember $q_{st}(0) - q_{st}(\delta x) \geq \max\left(\varepsilon_q \min\left(1, \|g_k\|^2\right), \theta_j\right)$

- From the termination criterion one gets $\varepsilon_q \|g_k\|^2 \leq -g_k^T \delta x_k - \frac{1}{2}\delta x_k^T \nabla^2 q_{\text{st}}(x_k)\delta x_k$
- From the positive definiteness of $\nabla^2 q_{\text{st}}$, we deduce $-g_k^T \delta x_k \geq \varepsilon_q \|g_k\|^2$
- The strict convexity of q_{st} and $-g_k^T \delta x_k \geq \frac{1}{2}\delta x_k^T \nabla^2 q_{\text{st}}(x_k)\delta x_k$ ensures that $\|\delta x_k\| \leq \frac{2}{\nu_{\text{min}}}\|g_k\|$
- We therefore get that $-g_k^T \delta x_k \geq \kappa_1 \|g_k\|^2$ and $\|\delta x_k\| \leq \kappa_2 \|g_k\|$, in other words, δx_k is gradient related
- A cosine condition and the convergence of the linesearch naturally follows

Saddle globalized

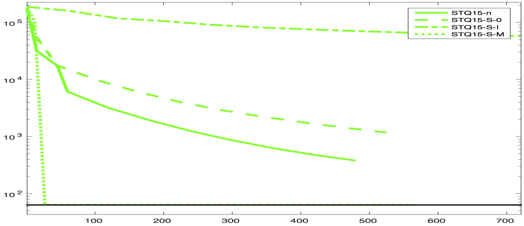


1. The performance of the preconditioners of type “B” and “T” is again poor
2. It is possible to check convergence periodically, and not at each iteration. This may increase the number of GMRES iterations, but also save evaluations of q_{st}. We call SAQℓ the corresponding algorithm
3. It would have been possible to check the gradient-relatedness property, but this would require the knowledge of κ_1 and κ_2.
4. The non-monotonic behaviour of q_{st} and J is obvious. Stopping cannot solely rely on maximum number of iterations

The three formulations and their preconditioners

- Saddle formulation (SA) involves $\begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{H}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}$ preconditioned e.g. by
$$P_M = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \tilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \tilde{\mathbf{L}}^{\mathrm{T}} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$
- State formulation (ST) $\min_{\delta\mathbf{x}} \ \frac{1}{2}\|\mathbf{L}\delta\mathbf{x} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta\mathbf{x} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$ preconditioned by the approximate Schur comp. $\widetilde{\mathbf{L}}^T\mathbf{D}^{-1}\widetilde{\mathbf{L}}$.
- Forcing (FO) is $\min_{\delta\mathbf{p}} \ \frac{1}{2}\|\delta\mathbf{p} - \mathbf{b}\|^2_{\mathbf{D}^{-1}} + \frac{1}{2}\|\mathbf{H}\mathbf{L}^{-\mathbf{1}}\delta\mathbf{p} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}$ preconditioned by $\mathbf{D}$.
- At each iteration of ST, $\mathbf{D}^{-1}$ is used. It is used for convergence check in SA.
- FO requires the sequential $\mathbf{L}^{-\mathbf{1}}$ and $\mathbf{L}^{-\mathbf{T}}$ at each iteration
- The main operations that are expected to influence the performance are anticipated to be operations involving the 3 above inverse operators.

A comment on the state formulation



- Even if $\tilde{\mathbf{L}}$ is "close" to $\mathbf{L}$, $\left(\widetilde{\mathbf{L}}^T\mathbf{D}^{-1}\widetilde{\mathbf{L}}\right)^{-1}$ may not be a good preconditioner of $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$
- Exemple $\mathbf{L} = \begin{pmatrix} 1 & 0 \\ 2+\alpha & 1 \end{pmatrix}$, $\tilde{\mathbf{L}} = \begin{pmatrix} 1 & 0 \\ \alpha & 1 \end{pmatrix}$, $\mathbf{D} = \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix}$,
- The condition of $\tilde{\mathbf{L}}^{-1}\tilde{\mathbf{L}}^{-T}\mathbf{L}^T\mathbf{L}$ have a finite limit when α goes to $+\infty$. Those of $\tilde{\mathbf{L}}^{-1}\mathbf{D}\tilde{\mathbf{L}}^{-T}\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ are not bounded

Numerical experiments

- We developped 2 data assimilation systems based on the Burgers equation and on a Quasi-Geostrophic model. Both are usual test cases in the DA literature
- To assess the parallel performance we consider 2 data layouts
  - A fully distributed layout. Corresponds to a MPI implementation, that exhibits the maximal degree of data distribution. Parallelism in computation is limited to avoid expensive exchanges of vector fields accross the interconnecting network. Example L_i and L_i^T are not done in parallel.
  - A hybrid memory framework where the distribution is made along the time dimension and the 3D fields are gobally accessible. Corresponds to a mixed MPI-OpenMP strategy
- Winning method : for a given ρ, the method that achieves $J(x_0) - J(x_f) \leq \rho\,(J(x_0) - J(x_\star))$ in a minimal elapsed time

The Burgers equations

- We consider the one dimensional dynamical system

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} - \nu\dfrac{\partial^2 u}{\partial x^2} = f(x) \\ \qquad (x,t) \in]0,1[\times\mathbb{R}_+^* \\ u(0,t) = u(1,t) = 0, \quad t > 0 \\ u(x,0) = k\sin(\pi x)\sin(\pi(1-x)); \\ \qquad x \in]0,1[\end{cases}$$

- The field u is partially observed in space and time
- This system is a fundamental partial differential equation occurring in various areas of applied mathematics as a prototype for conservation equations that can develop shock waves

The quasi-geostrophic model

Potential vorticity q_i is given in the 2-layer model by (ψ_i is the stream function)

$$q_1 = \nabla^2\psi_1 - \frac{f_0^2 L^2}{g' H_1}(\psi_1 - \psi_2) + \beta y, \qquad q_2 = \nabla^2\psi_2 - \frac{f_0^2 L^2}{g' H_2}(\psi_2 - \psi_1) + \beta y + R_s,$$

Conservation of potential vorticity gives

$$\frac{D_i q_i}{Dt} = 0, \quad i = 1, 2$$

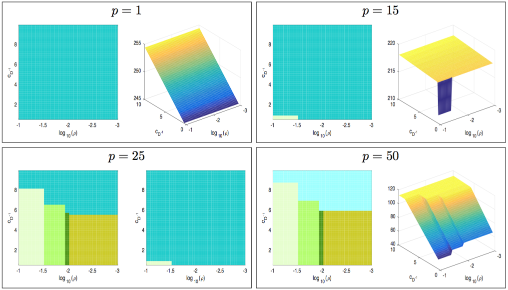
where D_i/Dt, is the total derivative, defined by

$$\frac{D_i}{Dt} = \frac{\partial}{\partial t} + u_i \frac{\partial}{\partial x} + v_i \frac{\partial}{\partial y} \quad \text{and} \qquad u_i = -\frac{\partial \psi_i}{\partial y}, \qquad v_i = \frac{\partial \psi_i}{\partial x}$$

are the horizontal velocity components in each layer. The model equation consist in solving for ψ_i.

The observations are observations of the non-dimensional stream functions, vector wind and wind speed. This simple system is used in studies since adequately captures important aspects of large-scale dynamics in the atmosphere.

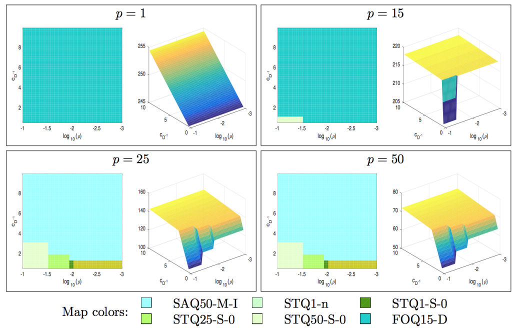
Burgers system fully distributed data layout



Map colors: SAQ50-M-I, STQ1-n, STQ1-S-0, STQ25-S-0, STQ50-S-0, FOQ15-D

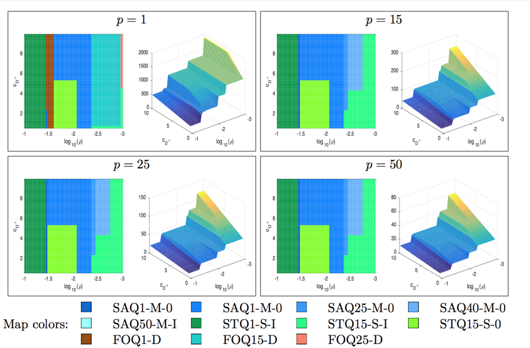
1. Forcing FOQ15-D dominates for sequential computations. It looses wrt to other when parallelism increases
2. Saddle point approaches are clearly better when parallelism increases and when c_{D-1} is high
3. State formulations may be affordable when c_{D-1} is moderate and requested accuracy is not too high
4. When c_{D-1} is moderate the algorithms using the state formulation

Burgers system temporal distribution



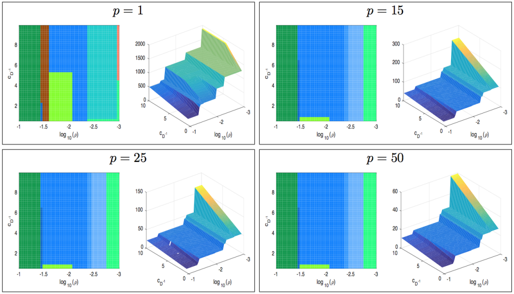
1. Further gain in elasped time is obtained with the hybrid model
2. The speed up is now from 250 to 70 cost units ($p = 50$)
3. The trends obtained with the previous model are amplified
4. The saddle formulation SAQ50-M-I outperfoms the other methods for when number of processors grows

QG system fully distributed data layout



1. When p is slow, tight competition between state and forcing
2. State formulations perform best for high and for low accuracy requirements
3. Saddle formulation better for moderate values of the accuracy
4. Excellent speed ud of the methods, from 2000 to 60 cost units ($p = 50$)
5. Improving accuracy is costly

QG system temporal distribution



Map colors: SAQ1-M-0, SAQ1-M-0, SAQ25-M-0, SAQ40-M-0, SAQ50-M-I, STQ1-S-I, STQ15-S-I, STQ15-S-0, FOQ1-D, FOQ15-D, FOQ25-D

1. Nearly same conclusions as before
2. Range of efficiency of the saddle formulation for intermediate values is enlarged

Preconditioning saddle Point Formulation of 4D-Var

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

- $\mathbf{B}$ is the most computationally expensive block and calculations involving $\mathbf{A}$ are relatively cheap.
- The inexact constraint preconditioner

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

where
  - $\widetilde{L}$ is an approximation to the matrix $\mathbf{L}$
  - $\widetilde{B} = [\widetilde{L}^{\mathrm{T}} \quad \mathbf{0}]$ is a full row rank approximation of the matrix $\mathbf{B} \in \mathbb{R}^{n\times(m+n)}$
- Update $\widetilde{B}$ using secant information (so-called "pairs") as in Quasi-Newton methods. Gives raise to a minimum Frobenius norm formula for rectangular matrices.

Preconditioning Saddle Point Formulation of 4D-Var

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

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

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

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

$\rightarrow$ In the previous iteration, we perform matrix-vector products with $\mathcal{A}$ and we have pairs satisfying

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

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

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

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

  $\rightarrow$ This update is not unique

- Among all updates, the update that we have introduced is not the least Frobenius norm update

Minimum F-norm preconditioning saddle point

- Starting from

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

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

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

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

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

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

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

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

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

Numerical Results with OOPS QG-model

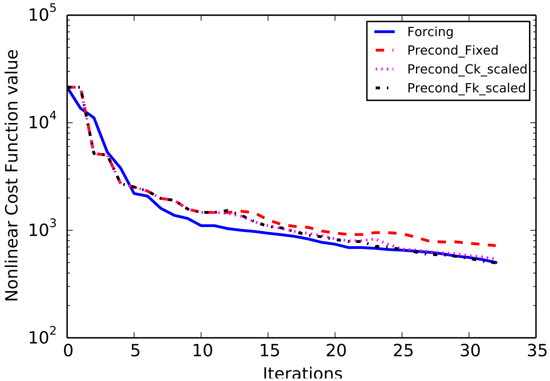


Figure – Nonlinear cost function values along iterations

→ Last 8 pairs were used to construct the preconditioner

Conclusions

- We considered parallel performance of nonlinear least-squares solvers for Data Assimilation. Three stategies are considered : state, forcing, saddle.
- Original saddle-point formulation is problematic for weakly constrained 4D-Var. This is due to the poor correlation between residual reduction and function or quadratic model decrease. The problem can be cured by a suitable globalization strategy focusing on quadratic reduction.
- We explored the parallel performance of the globalized algorithms using two simple data layouts and parallel programming situations : MPI and OpenMP+MPI, where communication of full fields accross the interconnecting network is minimized.
- Cost of evaluating D^{-1} and accuracy level of the quadratic minimization appear as important factors for analysing the respective merits of the 3 methods.
- For both Burgers and QG there is no clear winner for all values of the parameters. Application dependent issue.
- To be done :
  - use approximate D^{-1} in the linear solver. Many questions : symmetry, convergence, positive definiteness,
  - experiments in a real system.

Some references

- S. Gratton, P. Laloyaux, A. Sartenaer, Derivative-free optimization for large-scale nonlinear data assimilation problems, Quarterly Journal of the Royal Meteorological Society 140(680) :943-957, 2014
- S. Gratton, M. Rincon-Camacho, E. Simon, and Ph.L. Toint, Observations thinning in data assimilation, EURO Journal on Computational Optimization (1) :31-51, 2015
- S. Gratton, V. Malmedy and Ph.L. Toint, Using approximate secant equations in limited memory methods for multilevel unconstrained optimization, Computational Optimization and Applications, 51(3) :967-979, 2012
- M. Fisher, S. Gratton, S. Gurol, Y. Trémolet, X. Vasseur, Low rank updates in preconditioning the saddle point systems arising from data assimilation problems, accepted in OMS
- E. Bergou, S. Gratton, and L.N. Vicente, Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation, SIAM/ASA Journal on Uncertainty Quantification 4 :924-951, 2016