FUNDP
NAMUR

CONJUGATE-GRADIENTS VERSUS MULTIGRID SOLVERS FOR DIFFUSION-BASED CORRELATION MODELS IN DATA ASSIMILATION

by S. Gratton[1,3], Ph. L. Toint[2] and J. Tshimanga[1]

Report NAXYS-14-2012 29 August 2012

naxys
Namur Center for Complex Systems

[1] ENSEEIHT, 2, rue Charles Camichel, 31000 Toulouse (France)

[2] University of Namur, 61, rue de Bruxelles, B5000 Namur (Belgium)

[3] CERFACS, 42, avenue Gaspard Coriolis, 31057 Toulouse (France)

http://www.fundp.ac.be/sciences/naxys

Conjugate-gradients versus multigrid solvers for diffusion-based correlation models in data assimilation

S. Gratton[1,2*], Ph. L. Toint[3], J. Tshimanga[1]

[1] *CERFACS / SUC URA 1875, Toulouse, France*
[2] *ENSEEIHT-INP, Toulouse, France*
[3] *NAXYS, FUNDP-University of Namur, Namur, Belgium*
*Correspondence to: CERFACS, 42 avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France. E-mail: serge.gratton@cerfacs.fr

This paper provides a theoretical and experimental comparison between conjugate-gradients and multigrid, two iterative schemes for solving linear systems, in the context of applying diffusion-based correlation models in data assimilation. In this context, a large number of such systems has to be (approximately) solved if the implicit mode is chosen for integrating the involved diffusion equation over pseudo-time, thereby making their efficient handling crucial for practical performance. It is shown that the multigrid approach has a significant advantage, especially for larger correlation lengths and/or large problem sizes.

Key Words: Covariance matrix; Correlation; Diffusion equation; Conjugate Gradient; Multigrid; Data Assimilation

Received . . .

Citation: . . .

1. Introduction

Forecasting methods in atmospheric and oceanic contexts use observations and models based on evolution equations. Most of these methods involve a data assimilation step, whose goal is to produce the “best” initial state, known as the analysis, which is necessary to run these models. As such, the data assimilation techniques are therefore central for the production of high quality forecasts. We focus in this paper on one of their major ingredients: the use of the “background-error covariance matrix” (which we denote by B) and whose role in producing the analysis is to smooth and spread information from observation points, as well as to weight the importance of the *a priori* state compared to the observation fit, see Daley (1991). In real world applications, obtaining B is a truly challenging task, and different approaches have been proposed to produce a suitable approximation. The “method of Control Variable Transforms” (CVTs) proposed by Bannister (2008), is used in most operational variational data assimilation systems, and can be viewed as an effective means of modelling multivariate aspects of B approximately in a very compact and effective way. In the same paper, Bannister also considered another method based on the correspondence between the covariance and the solution of the diffusion equation, which is acknowledged to be suitable for oceanic data assimilation. In addition, the diffusion operators allows to deal with inhomogeneity, anisotropy and complicated boundary conditions. The product of the covariance matrix by a state vector then amounts to integrating a suitable diffusion equation over a finite time period using the given state vector as initial condition, as shown in Weaver and Courtier (2001). This modelling technique has been employed by Derber and Rosati (1989) and also by Weaver and Courtier (2001) using an explicit solution of the diffusion equation, while Mirouze and Weaver (2010) and Carrier and Ngodock (2010) recently priviledged an implicit solution, because of its unconditionally stable nature. In this interesting approach, backward Euler approximation in time and finite difference approximation in space are employed to solve the diffusion equation.

More specifically, the variational data assimilation problem can be stated as an unconstrained nonlinear optimization problem whose objective function involves the inverse of B and is typically written (see Le Dimet and Talagrand (1986)) as

$$\mathcal{J}[x(t_0)] = \frac{1}{2}(x(t_0) - x^b)^{\mathrm{T}} B^{-1}(x(t_0) - x^b) + \frac{1}{2}\sum_{j=0}^{p}(\mathfrak{H}_j[x(t_j)] - y_j^o)^{\mathrm{T}} R_j^{-1}(\mathfrak{H}_j[x(t_j)] - y_j^o), \quad (1)$$

where the state vector $x(t_j)$ satisfies a discrete nonlinear model of evolution given by

$$x(t_j) = \mathcal{M}_{j,0}[x(t_0)]. \quad (2)$$

Here, the matrices B and R_j are respectively the background-error and observation-error covariance matrices, and $\mathfrak{H}_j$ represents the observation operator. This problem is typically solved using the Gauss-Newton approach for nonlinear least-squares, also known in this context under the name of incremental four-dimensional variational method (4Dvar) (see Courtier et al. (1994)). In this setting, linearization of the operators in the functional (1) gives rise to *large* convex quadratic subproblems, or, equivalently, to *large* linear systems of the form

$$(B^{-1} + \mathcal{H}^T R^{-1}\mathcal{H})x = \mathcal{H}^T R^{-1} y \quad (3)$$

which have to be solved in sequence. Each of these systems is preconditioned by B and is typically solved using an iterative algorithm. Each iteration of this algorithm involves one matrix-vector product with each of the matrices B, H, H^T and R^{-1}. When, as considered here, the matrix-vector product by B is obtained by evolving the diffusion operator over several (pseudo-)time steps using an implicit discretization scheme, each of these products requires the solution of several linear systems. Carrier and Ngodock (2010), in their experimental extension of the work of Weaver and Courtier (2001), have chosen to use the well-known conjugate-gradients algorithm as a solver for these systems. Interestingly, they reached the conclusion that the implicit solution can be up to five times more efficient in terms of CPU-time than an explicit scheme, with nearly identical results in the analysis.

Our aim in the present paper is to compare this strategy with the alternative of using a multigrid approach as linear solver in the course of the implicit integration corresponding to the product by B^{-1}. We are concerned with the case where B is large and inexact matrix-vector products are allowed, this last point being motivated by previous research on inexact matrix-vector products in range-space Krylov solvers for linear systems of type (3), see Gratton et al. (2011).

The paper is structured as follows. In Section 2, the diffusion equation model of covariance operator is briefly presented. Section 2.2 provides a review of the multigrid algorithms and their relevant features in view of the present investigation. We illustrate the relevance of our analysis with an academic application in Section 4. Conclusions and perspectives are given in Section 5.

2. Modelling a covariance matrix using a diffusion equation

The equivalence between a covariance operator and the diffusion equation constitute the key idea in the model of the covariance matrix B that we consider in this study.

2.1. *Formulation*

We consider a vector z of length $N = n^2$ representing a discrete state field over a unit square Ω. To evaluate the action of an N-order covariance matrix B on the state vector z, one uses the discrete solution of an *adequate diffusion equation*, where z defines the initial condition as we will shortly show. The principle of this modelling scheme is to link the correlation length-scales $\mathcal{L}$ in B to the (positive) diffusion parameter κ and the integration period T. As explained in Mirouze and Weaver (2010, Formula (22)), this is achieved using the relation

$$\mathcal{L} = \sqrt{2\kappa T}, \quad (4)$$

where one chooses T and then determines κ according to the desired value of $\mathcal{L}$. For the following discussion, let us consider the 2D *diffusion* equation for the scalar function $\eta(x, t)$ on a unit-square given by

$$\frac{\partial \eta}{\partial t} - \kappa\Delta\eta \;=\; 0, \;\; x \in \Omega = (0,1)^2, \;\; 0 < t \leq T, \quad (5)$$

subject to

$$\begin{aligned} \eta(x,0) &= \eta_0, \;\; x \in \Omega, & (6)\\ \eta(x,t) &= 0, \;\; x \in \partial\Omega, \;\; 0 < t \leq T, & (7) \end{aligned}$$

where, $\Delta\eta$ is the 2-dimensional (continuous) Laplacian of $\eta(x, t)$.

2.2. *The implicit Euler approach*

An explicit integration scheme can be considered for solving (5). But, as pointed out again in Mirouze and Weaver (2010), this scheme, although easy to implement, is only conditionally stable, which may lead to the use of very small time steps and therefore to a significant amount of computation for each integration. An alternative is then to use an implicit scheme, which is known to be unconditionally stable. We consider the implicit Euler approximation in time and finite difference approximation in space for the numerical solution of (5) where the (discrete version of the) initial state is $\eta_h^{(0)} \in \mathbb{R}^N$, and where

$$N = n^2$$

is the total number of interior points, with n being the number of interior grig point per edge of the unit square. Choosing p to be the number of time steps in the integration, the obtained solution $\eta_h^{(p)}$ yields the desired state.

Now define

$$h = 1/(n+1)$$

as the spatial mesh size and let

$$\delta t = T/p$$

be the time step length. Also denote by I_N and I_n the N-order and n-order identity matrices, respectively. Finally, define

$$\theta = \kappa\delta t = \frac{\mathcal{L}^2}{2p}, \quad (8)$$

and the N-order matrix

$$\Delta_h = \frac{1}{h^2}\text{blktridiag}\,[-I_n,\ T_n,\ -I_n], \tag{9}$$

the discrete Laplacian corresponding to the continuous (negative) Laplacian $-\Delta$ in (5), where $T_n = \text{tridiag}\,[-1,\ 4,\ -1]_{n\times n}$. In an implicit scheme, at each step $l = 1, \dots, p$, the corresponding state is then approximated by

$$\eta_h^{(l)} \quad = \quad (I_N + \theta\Delta_h)^{-1}\eta_h^{(l-1)}. \tag{10}$$

If we set

$$\hat{\Delta}_h = I_N + \theta\Delta_h, \tag{11}$$

we may now to simulate the action of B on z, noted $[Bz]$, by formally computing

$$[Bz] \equiv \left(\text{diag}(\hat{\Delta}_h^{-p})\right)^{-\frac{1}{2}} \left(\hat{\Delta}_h^{-p}\right) \left(\text{diag}(\hat{\Delta}_h^{-p})\right)^{-\frac{1}{2}} z. \tag{12}$$

Note the presence of the normalization factors in $\text{diag}(\hat{\Delta}_h^{-p})$ whose role is to impose the diagonal elements of B to be equal to 1, a necessary property of correlation matrices. Equivalently, (12) can be decomposed,using the recurrence (10), as

$$\begin{aligned}
\eta_h^{(0)} &= \left(\text{diag}(\hat{\Delta}_h^{-p})\right)^{-\frac{1}{2}} z, && (13)\\
\eta_h^{(l)} &= \hat{\Delta}_h^{-1}\eta_h^{(l-1)}, \qquad l = 1, \dots, p, && (14)\\
[Bz] &= \left(\text{diag}(\hat{\Delta}_h^{-p})\right)^{-\frac{1}{2}} \eta_h^{(p)}. && (15)
\end{aligned}$$

Note that, at each time step l, the computation of $(\hat{\Delta}_h)^{-1}\eta_h^{(l-1)}$ corresponds to a solution of a linear system of matrix $\hat{\Delta}_h$ and right hand side $\eta_h^{(l-1)}$. This will be the main concern of Section 3.

2.3. *Eigenpairs of $\hat{\Delta}_h$*

Our subsequent analysis also requires some knowledge of the eigenpairs of the discrete Poisson-like operators $\hat{\Delta}_h$, which we now briefly survey. We recall well-known results in the case of a square domain with homogeneous Dirichlet boundary conditions (see (Meyer 2000, p. 565)).

Proposition 1 *Define*

$$\xi_h^k = \sin^2\left(\frac{kh\pi}{2}\right), \qquad \eta_h^l = \sin^2\left(\frac{lh\pi}{2}\right), \tag{16}$$

for $k, l = 1, \dots, n$ with $h = 1/(n+1)$. The eigenvalues of Δ_h are then given by

$$\lambda^{k,l}(\Delta_h) = \frac{4}{h^2}(\xi_h^k + \eta_h^l), \tag{17}$$

with corresponding orthogonal eigenvectors (the Fourier modes)

$$\mathbf{v}_h^{k,l} = [v_h^{k,l}], \quad v_h^{k,l} = \sin(ikh\pi)\sin(jlh\pi), \tag{18}$$

for $i, j = 1, \dots, n$. Moreover,

$$\|\mathbf{v}_h^{k,l}\|_2 = \frac{n+1}{2}. \tag{19}$$

The expression of eigenvalues in (17) suggests that the extreme values are approximately given by

$$\lambda_{\min}^{k,l}(\Delta_h) \approx 2\pi^2 \qquad \text{and} \qquad \lambda_{\max}^{k,l}(\Delta_h) = \frac{4}{h^2}. \tag{20}$$

In (18), we distinguish oscillatory and smooth modes, the first being characterized by $1 \le k, l \le \frac{n-1}{2}$ and the second by $\max(k,l) > \frac{n-1}{2}$. From Proposition 1, it is straightforward to establish the following corollary.

Corollary 1 *The eigenvalues of $\hat{\Delta}_h$ are*

$$\lambda_h^{k,l}(\hat{\Delta}_h) = 1 + \theta\lambda^{k,l}(\Delta_h), \quad 1 \le k, l \le n, \tag{21}$$

with corresponding eigenvectors defined in (18).

Using (20), we can easily deduce the value of the extreme eigenvalues of $\hat{\Delta}_h$ as

$$\lambda_{\min}^{k,l}(\hat{\Delta}_h) \approx 1 + 2\pi^2\theta \quad \text{and} \quad \lambda_{\max}^{k,l}(\hat{\Delta}_h) = 1 + \frac{4\theta}{h^2}. \tag{22}$$

As a consequence of the results above, the eigenvalue decomposition of $\hat{\Delta}_h$ is given by

$$\hat{\Delta}_h = V_h\hat{\Lambda}_h V_h^T, \tag{23}$$

where the columns of V_h are the normalized (see Equation (19)) vectors $\frac{2}{n+1}\mathbf{v}_h^{k,l}$ and where $\hat{\Lambda}_h$ is the diagonal matrix containing the eigenvalues of $\hat{\Lambda}_h$ defined in (21).

3. The Conjugate Gradient and multigrid solvers

The object of this section is to discuss two strategies, the Conjugate Gradient and multigrid solvers, for computing (14) by solving the (typically large) symmetric positive definite (spd) system

$$\hat{\Delta}_h\eta_h^{(l)} \quad = \quad \eta_h^{(l-1)}, \qquad l = 1, \dots, p. \tag{24}$$

3.1. *The conjugate gradient method*

The CG method (Hestenes and Stiefel 1952) is a practical algorithm for solving large-scale linear systems when the system matrix is symmetric and positive definite. To simplify notation, we let $\hat{\boldsymbol{\Delta}}$ denote $\hat{\boldsymbol{\Delta}}_h$, in this subsection. The standard CG algorithm to solve $\hat{\boldsymbol{\Delta}}x = \eta$ is summarized as follows as Algorithm 1 on the next page.

At each iteration of this algorithm, the main computational tasks are the matrix-vector product $\hat{\boldsymbol{\Delta}}\mathbf{p}_{i-1}$ evaluation, the inner products $\mathbf{p}_{i-1}^T\hat{\boldsymbol{\Delta}}\mathbf{p}_{i-1}$ and $\mathbf{r}_i^T\mathbf{r}_i$ computation, and calculation of three vectors sums. If one includes the initialization cost, the total number of floating point operations of i iterations of the method applied on a system of size N can then be estimated as

$$4N + i(10N + 2\|\hat{\Delta}\|_0) \approx 20iN \ \text{ flops}, \tag{25}$$

where $\|\cdot\|_0$ is the number of nonzero entries in a matrix, and where we have used the fact that $\|\hat{\Delta}\|_0 \approx 5N$.

This CG algorithm has the important property that each intermediate estimate, $\mathbf{x}_i$,

Algorithm 1 The CG solver

```
$\mathbf{x}_0 \; = $ initial estimate
$\mathbf{r}_0 \; = \; \hat{\mathbf{\Delta}}\mathbf{x}_0 - \eta$
$\mathbf{p}_0 \; = \; -\mathbf{r}_0$
$i \; = \; 0$
while stopping criterion not satisfied do
    $i \; = \; i + 1$
    $\alpha_{i-1} \; = \; \mathbf{r}_{i-1}^T \mathbf{r}_{i-1} \,/\, \mathbf{p}_{i-1}^T \hat{\mathbf{\Delta}} \mathbf{p}_{i-1}$
    $\mathbf{x}_i \; = \; \mathbf{x}_{i-1} \, + \, \alpha_{i-1}\mathbf{p}_{i-1}$
    $\mathbf{r}_i \; = \; \mathbf{r}_{i-1} \, + \, \alpha_{i-1}\hat{\mathbf{\Delta}}\mathbf{p}_{i-1}$
    $\beta_i \; = \; \mathbf{r}_i^T \mathbf{r}_i \,/\, \mathbf{r}_{i-1}^T \mathbf{r}_{i-1}$
    $\mathbf{p}_i \; = \; -\mathbf{r}_i \, + \, \beta_i \mathbf{p}_{i-1}$
end while
end.
```

minimizes $F[\mathbf{x}] = \frac{1}{2}\mathbf{x}^T\hat{\Delta}\mathbf{x} - \mathbf{x}^T\eta$ over the Krylov space $\mathbf{x}_0 + \text{span}\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_i\} = \mathbf{x}_0 + \text{span}\left\{\mathbf{r}_0, \hat{\mathbf{\Delta}}\mathbf{r}_0, \ldots, \hat{\mathbf{\Delta}}^{i-1}\mathbf{r}_0\right\}$ (see Golub and Van Loan (1996) for a comprehensive review of the properties of the CG method). The stopping criterion is usually defined by a parameter ε indicating the desired reduction in the norm of the gradient relative to the initial gradient norm (Dennis and Schnabel 1983), or by a maximum allowed number L of inner iterations.

The convergence behaviour of the CG method in *exact arithmetic* is commonly described by the following two properties (see Golub and Van Loan (1996), p. 530). The first property states that the CG algorithm will terminate with $\mathbf{x}_i = \mathbf{x}^\star$ for some $i \leq r$, where $\mathbf{x}^\star$ is the exact minimizing solution and $r \leq n$ is the number of distinct eigenvalues of $\hat{\mathbf{\Delta}} \in \mathbb{R}^{n\times n}$. The second property (Nocedal and Wright 1999, Theorem 5.5) states that the error $\epsilon_i = \mathbf{x}_i - \mathbf{x}^\star$ of the estimates generated by the CG algorithm satisfies the inequality

$$\frac{\|\epsilon_i\|_{\hat{\mathbf{\Delta}}}}{\|\epsilon_0\|_{\hat{\mathbf{\Delta}}}} \leq \frac{\lambda_{n-i}(\hat{\mathbf{\Delta}}) - \lambda_1(\hat{\mathbf{\Delta}})}{\lambda_{n-i}(\hat{\mathbf{\Delta}}) + \lambda_1(\hat{\mathbf{\Delta}})} \equiv K_{cg}^i, \tag{26}$$

where

$$\|\epsilon_i\|_{\hat{\mathbf{\Delta}}} = \sqrt{\epsilon_i^T \hat{\mathbf{\Delta}} \epsilon_i}$$

is the so-called energy norm of ϵ_i.

The first property implies that the number of CG iterations cannot exceed the dimension of $\hat{\mathbf{\Delta}}$. In practice, this property translates into a faster convergence when eigenvalues of $\hat{\mathbf{\Delta}}$ are more clustered. The second property is of direct interest here, since it allows the analysis of a bound on the speed of convergence, obtained by substituting the dependence of the $\lambda_i(\hat{\mathbf{\Delta}})$ on θ (and hence on $\mathcal{L}$), analyzed in Section 2.3, within the definition of K_{cg}^i in (26). This is illustrated in Figure 1, where the value of the bound in the right-hand side of (26) is shown, for an example, as a function of the computational burden (itself a function of i via (25)) for different values of θ.

This leads to the conclusion that larger values of θ (and $\mathcal{L}$) might result in slower convergence for the CG algorithm.

3.2. The multigrid solver

Another strategy is possible that exploits the grid structure obtained by discretizing the evolution model (2) on a hierarchy of m grids arranged from the finest to the coarsest, denoted by $\Omega_h, \Omega_{2h}, \ldots, \Omega_{2^{m-1}h}$. For simplicity, assume that these grids are defined on the unit square. The system (24) may now be written for each of these grids, with corresponding matrices $\hat{\mathbf{\Delta}}_h, \hat{\mathbf{\Delta}}_{2h}, \ldots, \hat{\mathbf{\Delta}}_{2^{m-1}h}$. The principle of the multigrid (MG) approach is to exploit the fact that high frequency components of the residual of this system can only be represented on the fine grids, on which they can be readily reduced by well-chosen iterative solvers (called smoothing operators). Moreover, the low frequency components of the resulting residual (on the fine grid) may also be considered as high frequency ones on a coarser grid, where the same process may be (recursively) applied in successive "cycles", among which "V-cycles" are most common. We refer the reader to Briggs et al. (2000) for an excellent introduction to this class of effective solvers. From a formal point of view, each such V-cycle is caracterized by an iteration matrix noted M_h which describes how the error evolves from cycle to cycle, as stated by the relation

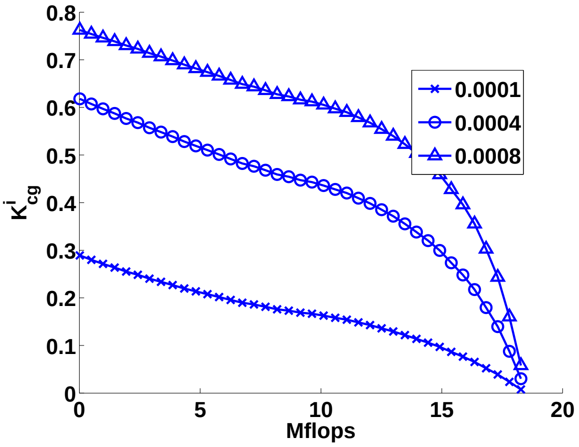

Figure 1. The value of K_{cg}^i versus the number of Mflops in CG for $\theta = 10^{-4}, 4\,10^{-4}, 8\,10^{-4}$ (corresponding to correlation lengths $0.04, 0.08, 0.11314$, respectively), and $N = 31^2$.

$$x_j - x = M_h^j(x_0 - x). \tag{27}$$

The iteration matrix itself may be written (see Trottenberg and al (Trottenberg et al. 2001, Theorem 2.4.1)) as

$$M_h \quad = \quad S_h \left(I_h - P_{2h}^h \left(I_{2h} - M_{2h} \right) \hat{\mathbf{\Delta}}_{2h}^{-1} R_h^{2h} \hat{\mathbf{\Delta}}_h \right) S_h, \tag{28}$$

with $M_{2^m h} = 0$ and where S_h is the involved smoothing operator, I_h is the identity matrix of the same order as M_h, and where P_h^{2h} and R_{2h}^h are the grid prolongation (from coarse to fine) and restriction (from fine to coarse) transfer operators, respectively. The computational cost at level h within a multigrid cycle can be estimated by

$$c_{pre}^h + 2\|R_h^{2h}\|_0 + 2\|P_{2h}^h\|_0 + 2\|\hat{\mathbf{\Delta}}_h\|_0 + c_{post}^h,$$

where c_{pre}^h is the cost of applying the "pre-smoothing" operator and c_{post}^h that of applying "post-smoothing", which correspond to the leftmost and rightmost occurences of S_h in (28), respectively. Summing up on all levels and using that facts that $c_{pre}^h = c_{post}^h \approx 2\|\hat{\mathbf{\Delta}}_h\|_0$ for a single iteration of the Gauss-Seidel smoother, that $\|R_h^{2h}\|_0 = \|P_{2h}^h\|_0 \approx 3|\Omega_h|$ if linear interpolation and its scaled transposed are used for prolongation and restriction, and again that $\|\hat{\mathbf{\Delta}}_h\|_0 \approx 5|\Omega_h|$, one then obtains that applying a multigrid V-cycle in our 2D setting (with $|\Omega_{2h}| = |\Omega_h|/4$) approximately costs

$$\frac{4}{3}(10+6+6+10+10+3)N = 60N \ \text{ flops}, \qquad (29)$$

where we also used the fact that the finest grid has N entries and included in the factor 3 in the bracket of the left-hand side to cover the cost of accumulating intermediate vectors at each level (We refer the reader to (Trottenberg et al. 2001, pp. 50-52) for an in-depth discussion of the cost of multigrid cycles). We thus conclude that, in our setting, a multigrid V-cycle is approximately 3 times as costly in flops as one iteration of CG.

Given the formulation (27), we derive an expression analogous to (26) in the form

$$\frac{\|\epsilon_j\|_{\hat{\mathbf{\Delta}}}}{\|\epsilon_0\|_{\hat{\mathbf{\Delta}}}} \le K^j_{mg}, \qquad (30)$$

where $K_{mg} = \|M^j_h\|_{\hat{\mathbf{\Delta}}}$. Here, we let again $\hat{\mathbf{\Delta}}$ to denote $\hat{\mathbf{\Delta}}_h$.

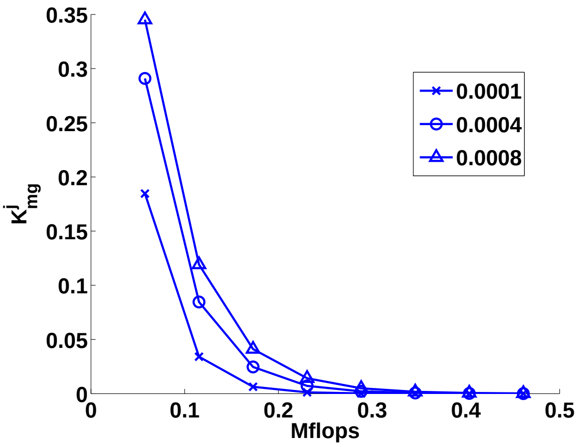

Figure 2. The value of $K^j_{mg}s$ versus the number of Mflops for $\theta = 10^{-4}, 4\,10^{-4}, 8\,10^{-4}$ (corresponding to correlation lengths $0.04, 0.08, 0.11314$, respectively), and $N = 31^2$.

Figure 2 illustrates the evolution of the upper bound on the relative error (in the energy norm) given by (30) for different values of the parameter θ, under the assumptions used to derive (29). A comparison with Figure 1 then suggests that the upper bound on the relative errors using the multigrid solver may decrease considerably faster than that obtained for conjugate-gradients, thereby providing an incentive for further investigation of the relative performance of these approaches from the point of view of the solutions themselves, rather than upper bounds on their residuals.

3.3. *A comparison of correlation shapes produced by CG and MG*

In a first simple approach to appreciate the quality of the solutions, we have computed, for different values of θ, the shape corresponding to the diffusion of (i.e., the application of B to) a Dirac pulse located at the centre of a unidimensional grid with 31 nodes, using a conjugate-gradient and a multigrid solver for approximately solving (24). The level of approximation was determined by the number of conjugate-gradient iterations (from 1 to 4) or the number of multigrid V-cycle (1). Recall that the CPU cost of each CG iteration is dominated by the computation of a matrix-vector product, while that of a V-cycle corresponds to approximately 3 CG iterations. For the sake of comparison, we have also computed the exact solution of this diffusion process by solving the systems (24) exactly (using a direct factorization algorithm). The results are pictured in Figures 3 to 5.

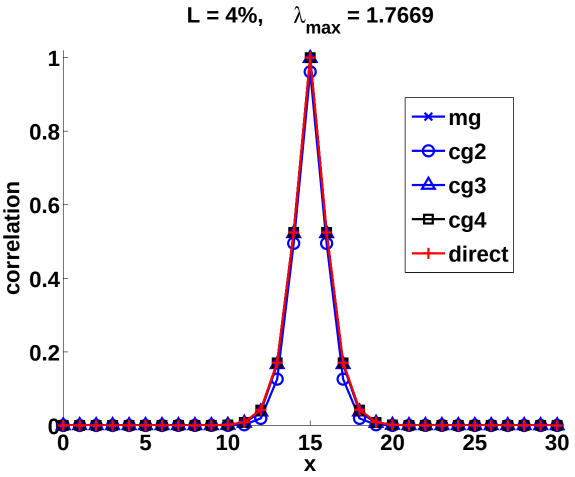

Figure 3. Diffusion of a Dirac pulse using approximate linear solvers and direct factorization for $\theta = 0.0001$; $N = 31$.

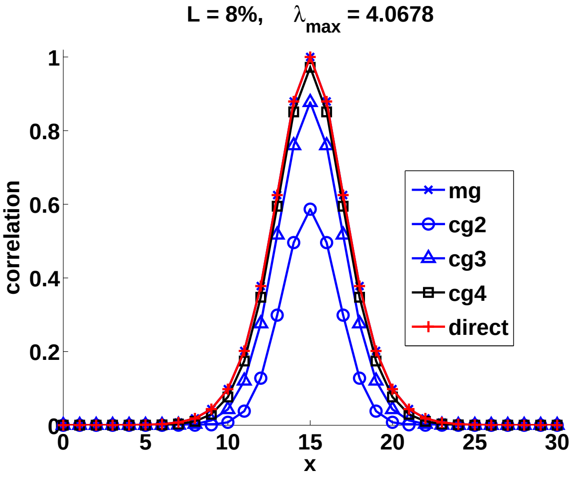

Figure 4. Diffusion of a Dirac pulse using approximate linear solvers and direct factorization for $\theta = 0.0004$; $N = 31$.

These figures show that the computed shapes using approximate linear solvers converge relatively quickly to the exact shape when the number of matrix-vector products increase, this convergence being faster for the multigrid technique than for CG, even for the case where 3 CG iterations are used (to make their estimated CPU cost comparable to that of one multigrid V-cycle). While the differences are negligible for small correlation lengths, they become more significant when $\mathcal{L}$ increases. These observations are fully coherent with what can de deduced from the comparison of the upper bounds on the residual sizes presented in the previous paragraph.

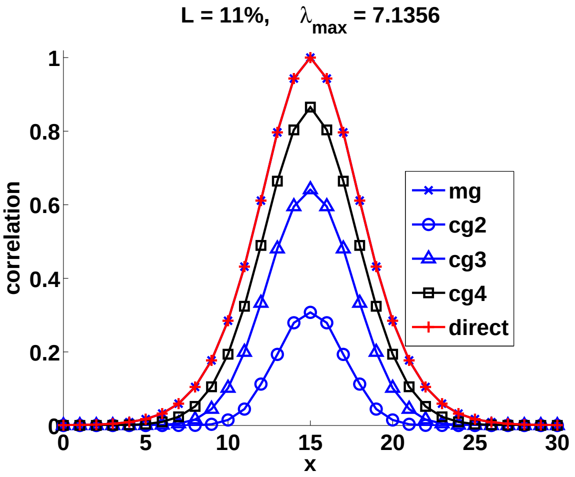

Figure 5. Diffusion of a Dirac pulse using approximate linear solvers and factorization for $\theta = 0.0008$; $N = 31$.

4. Numerical illustration in an academic data assimilation system

We finally conclude our comparison of conjugate-gradients versus multigrid solvers for diffusion-based correlation models by presenting numerical results obtained with these two methods in an academic data assimilation system. The prevision model used in this system consists of damped $2D$ shallow water equations delimited by an artificial bassin of size $L_x \times L_y$, yielding the system

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv + g\frac{\partial z}{\partial x} &= \nu \triangle u \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu + g\frac{\partial z}{\partial y} &= \nu \triangle v \\ \frac{\partial z}{\partial t} + u\frac{\partial z}{\partial x} + v\frac{\partial z}{\partial y} + z\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) &= \nu \triangle z \end{aligned}$$

with boundary and initial conditions given by

$$u(x,y,t)|_{\partial\Omega} = 0, \quad v(x,y,t)|_{\partial\Omega} = 0,$$
$$u(x,y,0) = 0, \quad v(x,y,0) = 0,$$
$$h(x,y,0) = h_0 + 50\sin(2\pi x)\sin(\pi y).$$

This system is integrated from time 0 to time T using a leapfrog scheme and initial geostrophic winds. The correlation length for this system is approximately equal to 10% of the basin width and 5% of its length and $p = 6$. Other model's physical parameters are presented in appendix.

We considered 6 test cases of increasing dimension (16^2, 32^2, 64^2, 128^2, 256^2 and 512^2) where the minimization of (1) was conducted using the Gauss-Newton algorithm (4Dvar) in which the system (3) was solved using RSFOM, an efficient strucure-exploiting "range-space" Krylov linear solver (see Gratton et al. (2011)) and the system (24) was approximately solved by either conjugate-gradients or multigrid (until the residual norm is decreased by a factor 10^{-4}) . Importantly, no significant variation was observed between the two methods in terms of quality of the solutions of the initial minimization problem (1).

Table I reports the number of Mflops used by both approaches as a function of problem size, while Figure 6 gives a graphical interpretation of the same results. If one remembers that the process measured here is the dominant cost in applying the diffusion-based correlation model, one can see that the simpler approaches used above to compare both methods are fully vindicated by these experimental results: the multigrid approach appears to yield substantial gains in computational efficiency, and these gains increase rapidly with problem size.

Size (N)	MG (Mflops)	CG (Mflops)
16^2	2.3×10^1	2.9×10^1
32^2	1.6×10^2	1.9×10^1
64^2	1.0×10^3	1.5×10^1
128^2	7.1×10^3	1.3×10^4
256^2	1.9×10^4	1.0×10^5
512^2	1.3×10^5	8.5×10^5

Table I. Number of Mflops in the solution of system (24) for a diffusion-based correlation model for the 2D shallow-water equations, as a function of problem size

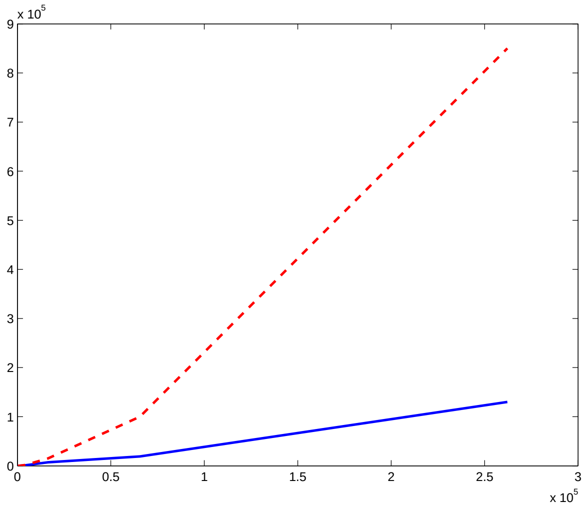

Figure 6. Number of matrix-vectors products in the solution of system (24) for a diffusion-based correlation model for the 2D shallow-water equations, as a function of problem size

5. Conclusion

We have discussed the algorithmic underpinnings of using diffusion-based correlation models in the context of the 4Dvar approach for data assimilation, and have focussed on the linear algebra tools for solving the (many) linear systems arising from the implicit integration approach for these models. More specifically, we have compared the conjugate-gradients and multigrid algorithms in this framework both from a theoretical and a numerical view point (on a 2D shallow-water test case). These comparisons clearly favour the multigrid approach, especially for larger correlation lengths and/or large problem sizes. Validating these conclusions in the context of operational systems with application-specific algorithmic variants (such as preconditioning) and complex geometries remains an important aspect of the question discussed here, but exceeds the scope of the present paper.

References

R. N. Bannister. A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. *Quarterly Journal of the Royal Meteorological Society*, 134(634):1071–1996, 2008.

W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, USA, 2nd edition, 2000.

M. J. Carrier and H. Ngodock. Background-error correlation model based on the implicit solution of a diffusion equation. *Ocean Modelling*, 5:45–53, 2010.

Ph. Courtier, J. N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120:1367–1387, 1994.

R. Daley. *Atmosperic data analysis*. Cambridge University Press, New York, USA, 1991.

J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1983. Reprinted as *Classics in Applied Mathematics 16*, SIAM, Philadelphia, USA, 1996.

J. Derber and A. Rosati. A global oceanic data assimilation system. *J. Phys. Oceanogr.*, 19:1333–1347, 1989.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

S. Gratton, Ph. L. Toint, and J. Tshimanga. Inexact range-space Krylov solvers for linear systems arising from inverse problems. *SIAM Journal on Matrix Analysis and Applications*, (to appear), 2011.

F.-X. Le Dimet and O. Talagrand. Variational assimilation of meteorological observations: Theoretical aspect. *Tellus*, 38A:97–110, 1986.

C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, USA, 2000.

I. Mirouze and A. T. Weaver. Representation of correlation functions in variational assimilation using an implicit diffusion operator. *Quarterly Journal of the Royal Meteorological Society*, 136:1421–1443, 2010.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Series in Operations Research. Springer Verlag, Heidelberg, Berlin, New York, 1999.

U. Trottenberg, C. W. Oosterlee, and A. Schüller. *Multigrid*. Elsevier, Amsterdam, The Netherlands, 2001.

A. Weaver and Ph. Courtier. Correlation modelling on the sphere using a generalized diffusion equation. *Quarterly Journal of the Royal Meteorological Society*, 127:1815–1846, 2001.

A. The shallow water model and its algorithmic solution

The system's physical parameters are presented in Table II.

L_x	32 10^6 m
L_y	8 10^6 m
Ω	$2\pi/86400$ sec
f	$2\Omega\sin(\pi/4)$
g	10 m s^{-2}
h_0	5000 m
ν	10^6

Table II. Physical parameters of the shallow-water model

The integration time is fixed to 86400 seconds for $N = 16^2, 32^2$ and 64^2, to 43200 seconds for $N = 128^2$ and to 22600 seconds for $N = 256^2$ and 512^2. Note that the correlation matrix for the complete system has a block diagonal form given by

$$B = \text{blkdiag}\,(B_u, B_v, B_z),$$

where B_u, B_v, and B_z are covariance matrices associated with variables, u, v and z, respectively. Each of these matrices is of dimension N.

The multigrid algorithm used for the experiments described in Section 4 is a slight variant of the standard V-cycle described above, in that no post-smoothing is performed ($c^h_{post} = 0$ for all levels). The number of levels is $\log(\sqrt{N})$ and the solution at the coarsest level is obtained by direct factorization. The involved matrices are computed, for all levels but the finest, from the Galerkin formula

$$\hat{\mathbf{\Delta}}_{2h} = R_h^{2h}\hat{\mathbf{\Delta}}_h P_{2h}^h$$

this computation being performed once and for all at the beginning of the calculation, together with that of the prolongation and restriction operators R_h^{2h} and P_{2h}^h, and that of the normalization factors diag$(\hat{\Delta}_h^{-p})$ in (12). Finally, both the CG and MG algorithms are terminated when solving the systems (24) as soon as a residual is found whose Euclidean norm is at most equal to 0.0001.