Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Computer Science 18 (2013) 1235 – 1244
Procedia Computer Science

International Conference on Computational Science, ICCS 2013

DD-OceanVar: a Domain Decomposition fully parallel Data Assimilation software for the Mediterranean Forecasting System

Luisa D'Amore[a,*], Rossella Arcucci[b], Luisa Carracciuolo[c], Almerico Murli[b,d]

[a] *University of Naples, Federico II, Complesso Universitario Monte S.Angelo, 80126 Naples, ITALY*
[b] *CMCC, Euro-Mediterranean Center on Climate Change, via Augusto Imperatore, 16 - 73100 Lecce, ITALY*
[c] *Istituto di Chimica e Tecnologia di Polimeri, CNR, Via Campi Flegrei, 34, 80078 Naples, ITALY*
[d] *SPACI, Southern Partnership for Advanced Computational Infrastructures, c/o Complesso Universitario Monte Sant'Angelo, Via Cintia, 80126, Naples, ITALY*

Abstract

OceanVar is a Data Assimilation (DA) software which is being used in Italy within the Mediterranean Forecasting System (MFS) to combine observational data (Sea level anomaly, sea-surface temperatures, etc.) with backgrounds produced by computational models of ocean currents for the Mediterranean Sea (namely, the NEMO framework). OceanVAR is based on a three-dimensional variational approach. We describe computational efforts aimed to design a *fully parallel* OceanVar software, based on Domain Decomposition (DD), which involves modification of the variational scheme on each sub domain. Our approach aims to face to the ever greater multi-level parallelism and scalability of the current and of the next generation of leadership computing facility systems, while fulfilling the specific requirements of OceanVar within the MFS.

Keywords: Data Assimilation; Domain Decomposition; parallel software;

1. Introduction

The Mediterranean Forecasting System (MFS) is a daily 10-day forecast system in operational use since 1998, and its ocean general circulation model (OGCM) is based on the Ocean Parallelise (OPA) code, which has subsequently been set up for the Mediterranean Sea (NEMO framework) [1]. Within the MFS, OceanVar is the operational assimilation software involving a wide range of observational data, from satellite-observed sea level anomalies (SLAs) and sea-surface temperatures (SSTs), to temperature and salinity profiles from expendable bathytermographs (XBTs,) and Argo-float profilers. The heat flux, on the other hand, is corrected by relaxing the modeled surface-layer temperatures towards the satellite-observed SST data.

This work describes the computational efforts towards the design of a *fully* parallel OceanVAR software to be coupled with NEMO within the MFS. The parallel approach is based on Domain Decomposition (DD) involving the physical domain, the regularization functional operator and the vector solution. The software will be therefore named as DD-OceanVar (Domain Decomposition OceanVar).

We show a simulated test case aimed to validate the effectiveness of the proposed approach both in terms of the numerical results and in terms of execution time. We are currently working on the employment of DD-OceanVar into the real configuration of the NEMO model within the MFS.

[*] Corresponding author. Tel.: +39-081-675625 ; fax: +39-081-7662106.
E-mail address: luisa.damore@unina.it.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.290

2. The domain decomposition approach

To implement a Data Assimilation algorithm in oceanography, it is natural to think about Domain Decomposition (DD) approach: each sub domain of the decomposition could be associated to a geographical area which could have its own observational and computational resources.

In NEMO, the DD approach agrees with the physical characteristics of the ocean model: the 3D domain is laid out on local processor memories following a 2D horizontal topological splitting. Further, each sub domain computes its own surface and bottom boundary conditions and it has a side wall overlapping interface which defines the lateral boundary conditions for computations in the inner sub domain. The overlapping area consists of the two rows at each edge of the sub domain. After a computation, a communication phase starts: each processor sends to its neighboring processors the updated values of the points corresponding to the interior overlapping area of its neighboring sub domain (i.e. the innermost of the two overlapping rows). The communication is done through message passing.

In DD-OceanVar we use this DD approach and the same overlapping area for the exchange of information on boundary conditions. These information, which are obtained on the edges, are crucial to solve the problem of data assimilation on sub domains as explained in the next section. About observations, in [2] are described three conceivable strategies for their spatial distribution. In DD-OceanVar we use, at this stage, one of these strategies named "model distribution". Moreover, our idea is to reformulate the DA regularization functional according to the physical domain decomposition. In this way we obtain a set of DA-subproblems, each one defined in a sub domain, that should be solved concurrently.

2.1. DD-OceanVAR computational model

Let t_k, $k = 0, 1, \ldots, n$ be a sequence of observation times and, for each k, let $x_{M_k} \equiv x_M(t_k) \in \Re^N$ be the vector denoting the state of the Mediterranean sea system at time t_k, where:

$$x_k = [T, S, \eta, u, v]^\star,$$

and T is the three-dimensional temperature field, S the three-dimensional salinity field, η the two-dimensional free surface elevation, and u,v are the total horizontal velocity components and where with $\star$ we denote the vector transposed.

At each time step t_k, let:

$$y_k = \mathcal{H}_k(x_k)$$

be the observations ($\mathcal{H}_k : \Re^N \mapsto \Re^p$ is the non-linear operator collecting the observations at time t_k [7]). The computational model of OceanVAR is the following non-linear least square problem [3]:

$$x_{DA}(t_k) = argmin_{x_k} J(x_k) \tag{1}$$

where

$$J(x_k) = \|\mathcal{H}_k(x_k) - y_k\|^2_{\mathbf{R}} + \|x_k - x_{M_k}\|^2_{\mathbf{B}} \tag{2}$$

x_{DA} is the so-called *analysis* (i.e. the estimation of the vector x_k at time t_k), **R** and **B** are the covariance matrices, whose elements provide the estimate of the errors on y_k and on x_{M_k}, respectively.

Let us consider the following overlapping decomposition of the physical domain Ω:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \tag{3}$$

such that $\Omega_i \cap \Omega_j = \Omega_{ij} \neq 0$ if Ω_i and Ω_j are adjacent. According to this decomposition we consider $J(x)$ onto the sub domain Ω_i, i.e. $J_i(x)$. The new issue induced by DD is to join the solution on the overlapping domains correctly. For that, we add a term to the cost function $J_i(\cdot)$ to enforce coupling between the domains [4]. The computational model of DD-OceanVar on Ω_i is:

$$x_{DA_i}(t_k) = argmin_x J_i(x_k) \tag{4}$$

where

$$J_i(x_k) = ||\mathcal{H}_{\mathbf{k_i}}(\mathbf{x}_{k_i}) - \mathbf{y}_{k_i}||^2_{\mathbf{R_i}} + ||\mathbf{x}_{k_i} - \mathbf{x_M}_{k_i}||^2_{\mathbf{B_i}} + ||\mathbf{x}_{k_i}/\Omega_{ij} - \mathbf{x}_{k_j}/\Omega_{ij}||^2_{\mathbf{B_{ij}}} \tag{5}$$

where $x_{DA_i}, \mathbf{x_{M_i}}, \mathbf{y_i}, \mathbf{x_i}, \mathbf{R_i}, \mathbf{B_i}, \mathcal{H}_i(x_i)$ are the quantities defined in (1) on Ω_i and $\mathbf{x}_i/\Omega_{ij}$, $\mathbf{x}_j/\Omega_{ij}$,$\mathbf{B_{ij}}$ are the restriction on Ω_{ij} of the quantities defined in (1).
Let

$$\tilde{x}_{DA_i} = \begin{cases} x_{DA_i} & on \quad \Omega_i \\ 0 & on \quad \Omega - \Omega_i \end{cases} \tag{6}$$

be the vector extension of x_{DA_i} on Ω, we define the *DD-analysis* as:

$$x^{DD}_{DA} = \sum_{i=1}^{n} \tilde{x}_{DA_i} \tag{7}$$

So, let $d = [y_k - H(x_k)]$ be the *misfit*, by using the following linearization of $\mathcal{H}$:

$$\mathcal{H}(x) = \mathcal{H}(z) + H(x - z)$$

where H is the matrix obtained by the first order approximation of the Jacobian of $\mathcal{H}$ and, by setting $v_i = V_i^T \delta x_i$, the *preconditioned* (see [5]) cost function is:

$$J_i(v_i) = \frac{1}{2} v_i^T v_i + \frac{1}{2}(H_i V_i v_i - d_i)^T R_i^{-1}(H_i V_i v_i - d_i) + \frac{1}{2}(V_{ij} v_i^+ - V_{ij} v_i^-)^T (V_{ij} v_i^+ - V_{ij} v_i^-) \tag{8}$$

where v_i^+ and v_i^- are shown in Figure 1.

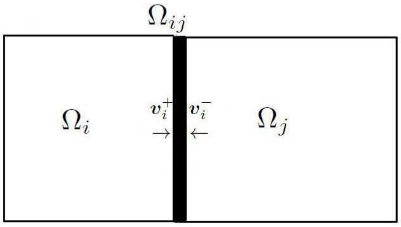

Fig. 1. quantities v_i^+ and v_i^-

On each subdomain of Ω, the operator J_i ($\forall i = 1, ...$), is minimized using the L-BFGS method (Figure 2).

2.2. DD-OceanVAR algorithm

Parallel algorithm performs, on each sub domain Ω_i, the DA computations needed to obtain x_{DA_i}: at the end of each iteration the exchange of boundary conditions is performed. The parallel algorithm running on each processor is described in Algorithm 1.
The main aspect of this algorithm is that it requires only few modifications on the existing sequential algorithm (see [6]). These modifications consist in changing the cost function $J_i(\cdot)$ and adding the exchange of the boundary conditions at each step of minimization process.
So, the basic idea of the DD-OceanVAR software is to split the discretization mesh into several smaller meshes (see Figure 3) and solve the DA problem by addressing independent local problems. Each processor has its own

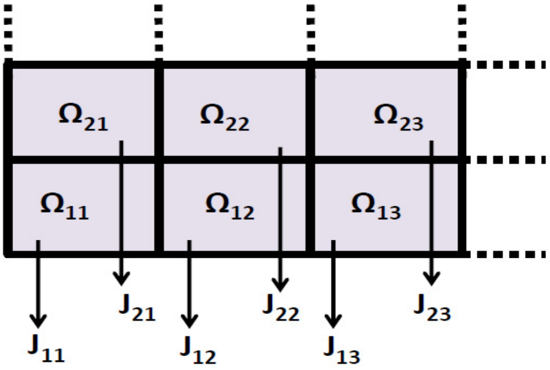

Fig. 2. DD-OceanVAR solutions computed solving concurrently the minimizing problem of each J_i with $i = (1,1),(1,2),(2,1),...$ defined in subdomains Ω_i

Algorithm 1 The DD-OceanVAR scheme

```
1: Acquire the observations $\mathbf{y}_i$ and the model vector $\mathbf{x}_{\mathbf{M}i}$
2: Define the operators $\mathbf{H}_i$
3: Compute $\mathbf{d}_i \leftarrow \mathbf{y}_i - \mathbf{H}_i\mathbf{x}_{\mathbf{M}i}$
4: Estimate the operators $\mathbf{R}_i$ and $\mathbf{B}_i$
5: Compute the matrix $\mathbf{V}_i$ from $\mathbf{B}_i$
6: Define the initial value of $\mathbf{x}_{DA_i}$
7: Compute $\mathbf{v}_{iopt} \leftarrow \mathbf{V}_i^{+}\mathbf{x}_{DA_i}$
8: repeat
9:     Send and Receive the boundary conditions from the adjacent domains needed during $J_i(\mathbf{v})$ evaluation
10:    Compute $J_{iopt} \leftarrow J_i(\mathbf{v}_{iopt})$
11:    Compute $gradJ_{iopt} \leftarrow \nabla J_i(\mathbf{v}_{iopt})$
12:    Compute new values for $\mathbf{v}_{iopt}$ by the L-BFGS steps from $J_{iopt}$ and $gradJ_{iopt}$
13: until (Convergence on the $\mathbf{v}_{iopt}$ values are obtained)
14: Compute $\mathbf{x}_{DA_i} \leftarrow \mathbf{x}_{\mathbf{M}i} + \mathbf{V}_i\mathbf{v}_{iopt}$
```

local memory and solves the model equation over a sub domain. The sub domain boundary conditions are defined by data exchange between processors which is exploited by explicit statements of the message passing paradigm (see point 9 of Algorithm 1).

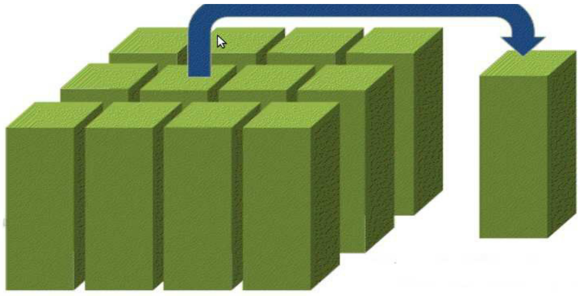

Fig. 3. 2D domain decomposition distribution of the discretization mesh

From the modeler's point of view, the algorithm running on each processor/sub domain is almost identical to the "mono-processor/domain" code.
Let us denote the processors number as $nproc = p \times q$. We assume a uniform decomposition along the (i, j)-axis, that is the i-axis is divided by p and the j-axis by q. Hence, if the size of Ω is $n_x \times n_y \times n_z$, then the size of sub domains are $nloc_x$, $nloc_y$, $nloc_z$ where:

$$nloc_x = \frac{n_x}{p} + 2o_x \tag{9}$$

$$nloc_y = \frac{n_y}{q} + 2o_y \tag{10}$$

$$nloc_z = n_z \tag{11}$$

These dimensions include the internal domain and the overlapping rows.

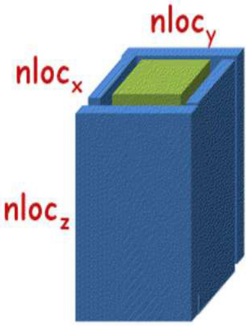

Fig. 4. sub-domains with overlapping

By using n_{xpp} and n_{ypp} to denote the position of the $(1, 1)$ grid-point of each sub domain in the global domain, each element of the local array x^{loc} corresponds to the element of the global array x^{glob}, as following:

$$x^{glob}(i + n_{xpp} - 1; j + n_{ypp} - 1; k) = x^{loc}(i; j; k) \tag{12}$$

where $1 < i < nloc_x$, $1 < j < nloc_y$, and $1 < k < nloc_z$.

About observations, we distribute them according to the "model distribution". Sub domains are associated with different geographic regions. We carry out a check on the geographical location of the observed data to attribute the observations to the processor related to the geographical region in which the measurement was made.

It is worth noting that in a more general framework, DD approach should be applied to each sub domain, in such a way each sub domain is processed in parallel and so on. Furthermore, each mono-domain code could be a multi-level parallel code (see [5]) instead of the sequential original code. By this way, we refer to *fully parallel* implementation, too.

3. Validation of the DD approach: a case study

In order to validate the proposed DD approach we described some results related to the quality of the numerical results and in terms of reduction in computation time [1]. All considerations are made on the basis of a case study

[1] All tests are performed on an AMD Opteron 6278@2.4GHz based system, with 512 GB of RAM memory, available in the context of the SCoPE Computing Infrastructure at the University of Naples Federico II.

Table 1. Estimation of the magnitude of the elements of e_{M_k} for different values of n.

n	$\left\|h_{true} - h_{M_k}\right\|_\infty$	$\left\|u_{true} - u_{M_k}\right\|_\infty$	$\left\|v_{true} - v_{M_k}\right\|_\infty$
64	4.999763e-03	4.999357e-03	4.999357e-03
128	4.999910e-03	4.999468e-03	4.999505e-03

based on a shallow water model as a simplified version of NEMO forecasting one: for that reason the shallow water model can be considered a consistent tool to get a *proof of concept* of the validity of the approach.

The shallow model equations are derived from the principles of conservation of mass and conservation of momentum. The independent variables are the time, t, and two space coordinates, x and y. The dependent variables are the fluid height or depth, h, and the two-dimensional fluid velocity field, u and v. With the proper choice of units, the conserved quantities are mass, which is proportional to h, and momentum, which is proportional to u and v. We assume that $n_x = n_y = n$ and $n_z = 3$, so the problem size is $N = n^2 * 3$. Finally, the time step used for the temporal discretization of the model is $dt = 0.01$.

The domain decomposition used is coherent with the software NEMO where, fixed p and q, the local dimensions of sub domains (considering overlapping) are defined as in (9) and (10).

We use as reference values the solution to the forecasting model denoted by x_{true_k}, where $k = 0, 1, ..., n$ counts the time step t_k:

$$x_{true_k} = [h_{true}, u_{true}, v_{true}]^\star,$$

For each time step, the operator H defined in (4) is assumed to be a piecewise linear interpolation operator whose coefficients are computed using the points of model domain nearest the observation values.

The standard deviation of the observation error is generally assumed equal to 0.5 [6]. So, the covariance matrix R is a diagonal matrix with elements equals to 0.5.

Let

$$x_{M_k} = [h_{M_k}, u_{M_k}, v_{M_k}]^\star,$$

be the vector containing the numerical solution of the forecasting model at time t_k and the error vector $e_{M_k} = |x_{true_k} - x_{M_k}|$ (see Table 1 for an estimation of the magnitude of the elements of e_{M_k}).

Then the covariance matrix B is given by the relationship:

$$B = e_{M_k} e_{M_k}^T.$$

To compute the matrix V in (8) we apply the TSVD (Truncated SVD) to the matrix B. So, we get

$$B = UdU^T = Ud^{\frac{1}{2}} d^{\frac{1}{2}} U^T = (Ud^{\frac{1}{2}})(Ud^{\frac{1}{2}})^T$$

and by posing $V = Ud^{\frac{1}{2}}$ we get V such that

$$B = VV^T.$$

Observation values are randomly chosen among the values of x_{M_k}.

Table 2 shows the quantities $\left\|h_{true} - h_{Comp}\right\|_\infty$, $\left\|u_{true} - u_{Comp}\right\|_\infty$ and $\left\|v_{true} - v_{Comp}\right\|_\infty$ such that

$$h_{Comp} = h_{M_k} + dh; \quad u_{Comp} = u_{M_k} + du; \quad v_{Comp} = v_{M_k} + dv$$

where dh, du and dv are obtained by DA performed starting from h_{M_k} , u_{M_k} and v_{M_k}.

We meant evaluate how well the DA solution $[h, u, v]_{Comp}$ approximates the model solution $[h, u, v]_{true}$. We observe that the results performed by using DD-OceanVar are consistent with those shown in Table 1 i.e. all the values of dh, du and dv have the same magnitude of the error between the model solution and the numerical model solution. Moreover the DA solution become more and more accurate (i.e. the values of $\left\|[h, u, v]_{true} - [h, u, v]_{Comp}\right\|_\infty$ decrease) as the number of sub domains increases (i.e. the dimension of each sub domain decreases).

Table 2. Values of $\left\|h_{true} - h_{Comp}\right\|_\infty$, $\left\|u_{true} - u_{Comp}\right\|_\infty$ e $\left\|v_{true} - v_{Comp}\right\|_\infty$ for different values of $nproc$ and of n.

n	$nproc = p \times q$	$\left\|h_{true} - h_{Comp}\right\|_\infty$	$\left\|u_{true} - u_{Comp}\right\|_\infty$	$\left\|v_{true} - v_{Comp}\right\|_\infty$
64	$nproc = 1, p = 1, q = 1$	5.778788e-03	5.778319e-03	5.778319e-03
	$nproc = 2, p = 2, q = 1$	5.455278e-03	5.454835e-03	5.454488e-03
	$nproc = 4, p = 2, q = 2$	5.239452e-03	5.243863e-03	5.244196e-03
	$nproc = 8, p = 4, q = 2$	5.120223e-03	5.127610e-03	5.127936e-03
128	$nproc = 1, p = 1, q = 1$	–	–	–
	$nproc = 2, p = 2, q = 1$	6.442963e-03	6.442394e-03	6.442441e-03
	$nproc = 4, p = 2, q = 2$	5.850961e-03	5.850444e-03	5.850183e-03
	$nproc = 8, p = 4, q = 2$	5.484419e-03	5.472367e-03	5.478658e-03

Table 3 shows the execution time required for the calculation of the quantities dh, du e dv. Times include the definition and the evaluation, on each sub domain Ω_i, of the functionals J_i. In order to evaluate the performance of the parallel algorithm we state the following:

Proposition *The scale-up factor of the described DD approach is* $S_{nproc} = nproc^2$

Proof: The most expensive computation of Algorithm 1 is the TSVD needed to obtain V_i, requiring $O(N^3)$ flops, on a problem of size N. Hence, the computation time $T_{loc}^{TSVD}(N_{loc})$ for performing the TSVD on each sub domain of size $N_{loc} = n_z \times \frac{n_x}{p} \times \frac{n_y}{q}$ is:

$$T_{loc}^{TSVD}(N_{loc}) = O\left(\left(n_z \times \frac{n_x}{p} \times \frac{n_y}{q}\right)^3\right).$$

In the DD approach we perform $nproc$ TSVD on problems of size N_{loc}, hence the total computation time T_{nproc}^{DD} should be:

$$T_{nproc}^{DD} = O\left(p \times q\left(n_z \times \frac{n_x}{p} \times \frac{n_y}{q}\right)^3\right).$$

or else:

$$T_{nproc}^{DD} = O\left(\frac{\left(n_z \times n_x \times n_y\right)^3}{(p \times q)^2}\right) = O\left(\frac{\left(n_z \times n_x \times n_y\right)^3}{(nproc)^2}\right).$$

Than the *scale-up* factor, of the parallel algorithm, computed respect to $nproc_1$ is:

$$S_{nproc,nproc_1}^{DD} = \frac{T_{nproc_1}^{DD}}{T_{nproc}^{DD}} = \frac{nproc^2}{nproc_1^2}$$

♣

In Table 3 we observe that as $nproc$ increases the execution time scales as $O(nproc^2)$, according to the *scale-up* factor.
Finally, in Figure 5 a graphical representation of the quantities characterizing the solution is showed for $n = 64$.

4. Conclusions and future work

We proposed a Domain Decomposition based approach to Data Assimilation. The novel approach consists on a new formulation of the three dimensional variational scheme on each subdomain. Our idea is to *"decompose"*

Table 3. Total execution time and *scale-up* factor for different values of *nproc* and of *n*.

n	$nproc = p \times q$	Total execution (secs) - T_{nproc}	T_1/T_{nproc}	$S^{DD}_{nproc,1}$
64	$nproc = 1, p = 1, q = 1$	5.3888e+03	1.0	1
	$nproc = 2, p = 2, q = 1$	9.5397e+02	5.6	4
	$nproc = 4, p = 2, q = 2$	2.0800e+02	25.9	16
	$nproc = 8, p = 4, q = 2$	6.6651e+01	80.8	64

n	$nproc = p \times q$	Total execution (secs) - T_{nproc}	T_2/T_{nproc}	$S^{DD}_{nproc,2}$
128	$nproc = 1, p = 1, q = 1$	–	–	–
	$nproc = 2, p = 2, q = 1$	6.0008e+04	1.0	1
	$nproc = 4, p = 2, q = 2$	6.2254e+03	9.6	4
	$nproc = 8, p = 4, q = 2$	2.3894e+03	25.1	16

the DA functional according to the physical domain decomposition obtaining a set of DA-subproblems, each one defined in a sub domain, to be solved concurrently. We use a case study based on the shallow water model to build a simulated case (a sort of *"phantom experiment"*) to verify the correctness of the solution. We are currently working to the implementation of this approach in the actual configuration of the NEMO model within the MFS.

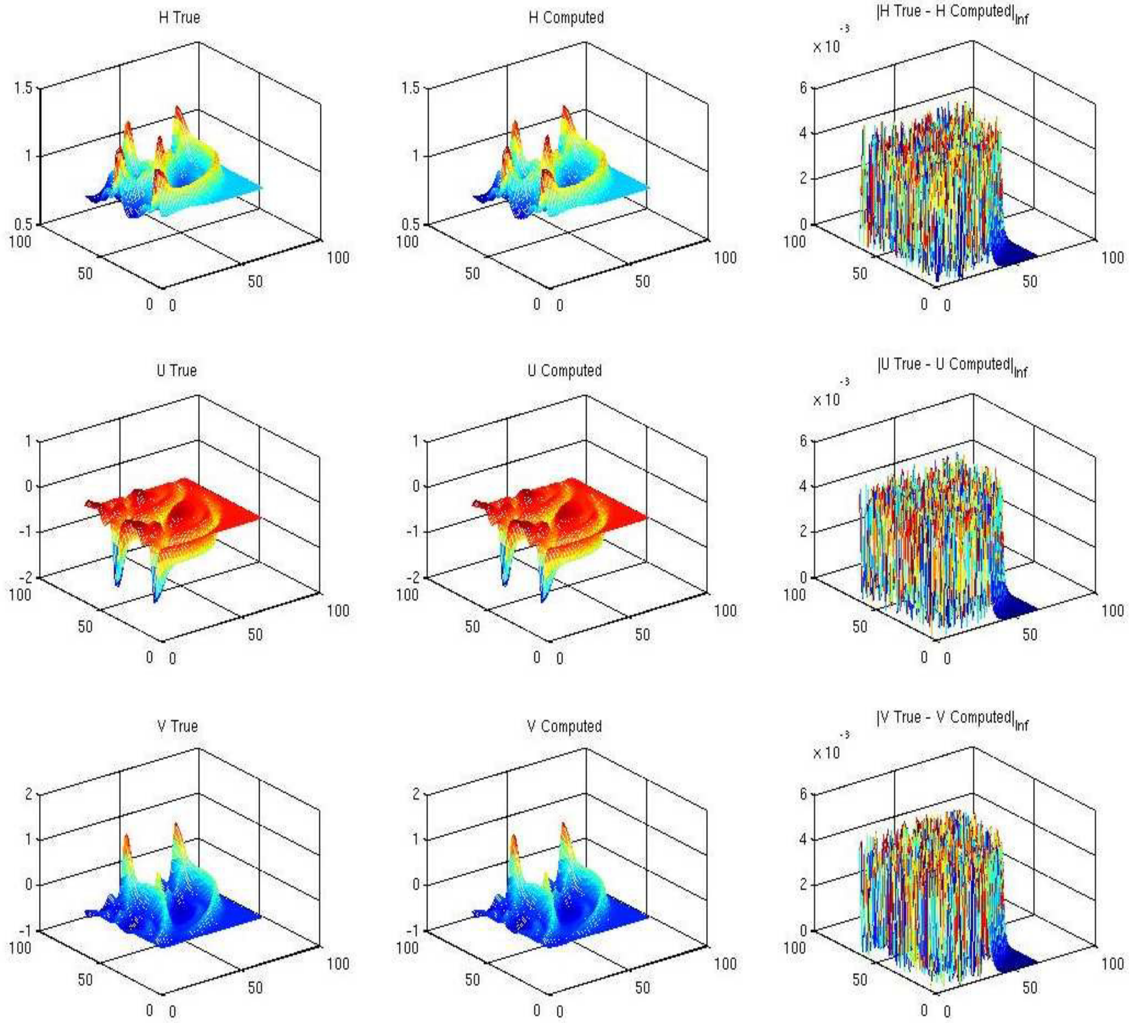

Fig. 5. Values of h_{true}, h_{Comp}, $\left\|h_{true} - h_{Comp}\right\|_{\infty}$, u_{true}, u_{Comp}, $\left\|u_{true} - u_{Comp}\right\|_{\infty}$, v_{true}, $v_{Computed}$ e $\left\|v_{true} - v_{Comp}\right\|_{\infty}$ obtained by setting $nproc = 4, p = 2, q = 2$ and $n = 64$

References

[1] *The NEMO System Home Page* - http://www.nemo-ocean.eu

[2] J. Rantakokko, *Strategies for parallel variational data assimilation* - Parallel Computing 23, 1997, pp. 2017-2039 - Elsevier

[3] S. Dobricic, N. Pinardi, *An oceanographic three-dimensional variational data assimilation scheme* - Ocean Modelling 22, 2008, pp. 89-105, Elsevier

[4] Y. Tremolet, F. X. Le Dimet, *Parallel algorithms for variational data assimilation and coupling models* - Parallel Computing 22, 1996, pp. 657-674, Elsevier

[5] L. D'Amore, R. Arcucci, L. Marcellino and A. Murli, *HPC computation issues of the incremental 3D variational data assimilation scheme in OceanVar software* - Journal of Numerical Analysis, Industrial and Applied Mathematics vol. 7, no. 3-4, 2012, pp. 91-105 .

[6] N. Gustafsson, X. Huang, X. Yang, K. Mogensen, M. Lindskog, O. Vignes, T. Wilhelmsso and S. Thorsteinsson, *Four-dimensional variational data assimilation for a limited area model* - Tellus - vol. 24, 2012

[7] L.D'Amore, R.Arcucci, L.Marcellino, A.Murli - *A Parallel Three-dimensional Variational Data Assimilation Scheme* - Numerical Analysis and Applied Mathematics, AIP C.P. 1389, 1829-1831