

COMPUTATION OF TWO-PHASE FLOW IN STEAM GENERATOR USING DOMAIN DECOMPOSITION AND LOCAL ZOOM METHODS

M. Belliard *, M. Grandotto

*Commissariat à l'Energie Atomique (CEA)
DEN/DTP/STH, Laboratoire de Thermohydraulique Appliquée
C.E. CADARACHE, BT 219, 13108 Saint Paul-lez-Durance Cedex, France*

Abstract

We present flow simulations in the Steam Generator of a pressurized water nuclear reactor using Domain Decomposition (DDM) and local zoom methods on workstation cluster.

Concerning the DDM, we use a Dirichlet-Neumann approach jointly with FEM for averaged mixture balance equations. The algorithm, based on parallel or sequential iteration-by-subdomain method, works with overlapping or nonoverlapping subdomains and with conforming or nonconforming meshing. With DDM, the computational problem size is easily increased to about 100,000 mesh cells and the CPU time is strongly reduced.

Concerning the Local Zoom computations, the used Local Defect Correction Method (LDC), in 3D local hierarchical multigrid context, is shown. The LDC computation results are compared with the classical full domain computation results (with high or low spatial resolution). We conclude in an improvement of the accuracy on the full domain with a high coherence between the zoom and the full domain.

Key words: Steam Generator, Zoom, Domain Decomposition Method (DDM), LDC Method, Local Hierarchical Multigrid, Two-phase Flows

* Corresponding author. Tel.: +33.(0)4.42.25.23.17, Fax : +33.(0)4.42.25.77.67
Email address: michel.belliard@cea.fr (M. Belliard).

1 INTRODUCTION

We present an application of Domain Decomposition Methods (DDM) and Local Zoom methods in the context of two-phase flow computation. Our application is the simulation of French Nuclear Steam Generator (SG) risers where liquid water is boiling, see Fig. 1. Simulations are required for evaluations of SG performances and safety analysis. The difficulties are brought by differences in geometrical and physical scales, turbulent and 3D flows through complex internal structures, exchange between the two phase,..

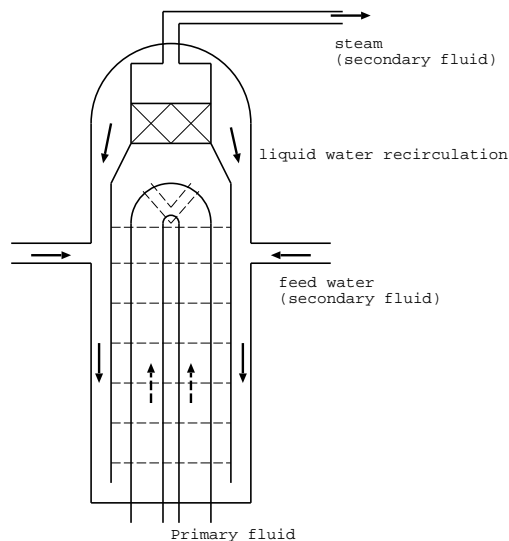


Fig. 1. French nuclear steam generator (scheme)

In industrial background, the problem is simplified by a gas-liquid mixture approach, like in the GENEPI software, see Grandotto et al. (1989), Grandotto and Obry (1996) and Obry et al. (1990). However, the number of data to store for each control volume limits the spatial resolution. Our goal is to

- increase by typically a factor three the simulation possibilities (memory) and reduce the CPU time,
- deal with non conforming grids and
- increase locally the simulation accuracy (without the computation at high resolution of the whole domain)

by using a memory distributed parallel implementation of DDM and local zoom methods on cluster workstations.

Roughly, this paper is divided in three parts. The first part is dedicated to a brief presentation of the two-phase fluid flow model, the numerical aspects and the coupling strategy. The second one deals with the description of the used DDM approach, illustrated with typical numerical results. Finally, in the third part, we present the Local Defect Correction (LDC) method used for the

zoom computations in a 3D local hierarchical multigrid context and we give some numerical results concerning a zoom simulation of the U-tube bundle region of a SG Mock-up.

2 NOMENCLATURE

- \vec{G} : mixture mass flux ($= \rho \vec{v}$)
- \vec{g} : gravity (m s^{-2})
- H : mixture specific enthalpy (J kg^{-1})
- H_{ls} : saturated liquid specific enthalpy (J kg^{-1})
- L : latent heat (J kg^{-1})
- P : pressure (Pa)
- Q : heat source (W m^{-3})
- t : time (s)
- \vec{v} : mixture velocity (m s^{-1})
- \vec{v}_R : relative velocity (gas minus liquid, m s^{-1})
- x : static quality ($\equiv \frac{H-H_{ls}}{L}$)
- β : porosity ($= \Omega_m / \Omega$)
- χ_T : turbulent diffusion coefficient for the mixture energy equation ($\text{kg m}^{-1} \text{s}^{-1}$)
- μ_T : two-phase turbulent dynamic viscosity ($\text{kg m}^{-1} \text{s}^{-1}$)
- ρ : mixture density (kg m^{-3})
- $\bar{\Lambda}$: two-phase friction tensor (s^{-1})
- Ω : elementary volume (m^3)
- Ω_m : mixture volume (m^3)
- φ^i : nodal function
- \cdot_r : restricted quantity

3 TWO-PHASE FLUID MODEL

After averaging the mass, momentum and energy equations for each phase (see Hughes and Chen (1977)), we sum them to get a mixture description of the two-phase flow. Provided that the following assumptions hold,

- surface tension, viscous and turbulent dissipation are neglected and pressure terms are neglected in enthalpy balance equation,
- same pressure for steam and liquid,
- eddy viscosity model,

we get :

(1) mass balance

$$\beta \partial_t \rho + \vec{\nabla} \cdot (\beta \rho \vec{v}) = 0 \quad (1)$$

(2) momentum balance

$$\begin{aligned} & \beta \rho \partial_t \vec{v} + \beta \rho (\vec{v} \cdot \vec{\nabla}) \vec{v} \\ & + \text{div}(\beta x(1-x)\rho \vec{v}_R^{\rightarrow} \otimes \vec{v}_R^{\rightarrow}) = \beta \rho \vec{g} - \beta \bar{\Lambda} \rho \vec{v} \\ & - \beta \vec{\nabla} P + \text{div}(\beta \mu_T (\vec{\nabla} \vec{v} + \vec{\nabla}^t \vec{v})) \end{aligned} \quad (2)$$

(3) enthalpy balance

$$\begin{aligned} & \beta \rho \partial_t H + \beta \rho (\vec{v} \cdot \vec{\nabla}) H + \text{div}(\beta x(1-x)\rho L \vec{v}_R^{\rightarrow}) \\ & = \beta Q + \text{div}(\beta \chi_T \vec{\nabla} H) \end{aligned} \quad (3)$$

We solve in H, P and \vec{v} variables. In order to compute ρ , x and L in function of H and P, we need water thermodynamic tables. The μ_T , χ_T , $\bar{\Lambda}$, \vec{v}_R^{\rightarrow} terms are obtained by the use of a large set of semi-empirical closure relations, see Obry et al. (1990). The most often used are the Schlichting model for μ_T and the drift-flux Lellouche-Zolotar model for \vec{v}_R^{\rightarrow} . The heat source Q in the enthalpy equation is linked to the resolution of an energy balance equation for the primary flow. To evaluate this term, we include other correlations about the heat exchange coefficient and the wall temperature.

According to the hyperbolic kind of the flow equations, we use Dirichlet boundary conditions at entry of the domain (mass flux and enthalpy) and Neumann ones at exit (pressure). The others sides of the domain are impermeable walls. Generally we consider them adiabatic and with no shear stress.

4 NUMERICAL CONSIDERATIONS

We approach the stationary flow by a transient computation. In our applications, we can neglect wave effects. So, we eliminate the time term in Equation (1). The numerical scheme is based on unstructured finite element method (FEM) with tri-linear hexahedral elements and a Crank-Nicholson time scheme. The H and \vec{v} variables take values at the nodes. In contrast, the P variable is defined by element. Concerning the equation coefficients (x , ρ , \vec{v}_R^{\rightarrow} , ...), they are generally defined by element, except β which is a nodal field. The stress and thermal flux terms are integrated by parts. We perform the lumping of the mass matrix, see Gresho et al. (1978) and use under integration for computing the element matrices. The diffusion terms are implicit like the friction one (momentum). Generally, the advection and drift terms

are explicit. As in Gresho et al. (1984), we include a BTM correction to increase stability of the central difference advection scheme. At each time step (\equiv outer iteration to solve non linear coupled equations), we first solve the primary fluid energy equation (fully implicit FEM) to get the enthalpy source term Q . Thus, we resolve the enthalpy equation and then the coupled mass-momentum ones by the Chorin-Gresho method, see Gresho and Chan (1990). We use a conjugated gradient method (CGM) preconditioned by the diagonal (\equiv inner iterations).

5 THE COUPLING STRATEGY

Based on the master/slaves concept, we have developed an application with the CEA code-linker software called ISAS, see De Gramont and Toumi (1996), which uses the PVM software. In this application, several GENEPI tasks are running simultaneously, coupled at every 'nc' time iterations by the ISAS task (coupling period). So, we mimic a Multiple Instructions Multiple Data (MIMD) application, with distributed memory, on a workstation network.

For each GENEPI task, the user can set up 'coupled boundaries'. The coupled boundaries are either 2D or 3D regions. Each coupled boundary is defined by a boundary condition type (as Dirichlet, LDC, ...), a GENEPI task name (providing the data) and an operator name (as interpolation, restriction, ...). After some information exchanges, here through the ISAS Master task, each GENEPI slave task knows which data it needs and which ones it must provide at each coupling iteration.

6 DOMAIN DECOMPOSITION METHOD

We are interested in the opportunities brought by DDM concerning the deal of non conforming grids (increasing mesh cells number in some particular regions independently of the other ones) and of large mesh cells number. In an other way, we wish to get a parallel version of the GENEPI software without significant changes and involving only a small number of processors, typically those available in a local workstation network. Taking this into account and due to the kind of equations and boundary conditions, we choose a parallel version of the Dirichlet-Neumann method, see Quarteroni (1990) and Le Tallec (1994), to solve Equations (1) to (3). We also use DDM approach for the primary fluid energy equation.

The computational kernel is advection - diffusion problem. The corresponding

interface conditions are both stress and variables continuity. For mass flux and specific enthalpy this writes :

$$\left\{ \begin{array}{l} \vec{G}_1 = \vec{G}_2, \\ \bar{\sigma}_1 \vec{n}_1 = \bar{\sigma}_2 \vec{n}_2, \end{array} \right. \quad (4)$$

and

$$\left\{ \begin{array}{l} H_1 = H_2, \\ \chi_{T1} \vec{\nabla} H_1 \cdot \vec{n}_1 = \chi_{T2} \vec{\nabla} H_2 \cdot \vec{n}_2, \end{array} \right. \quad (6)$$

with

$$\bar{\sigma} = -P\bar{I} + \mu_T (\vec{\nabla} \vec{v} + \vec{\nabla}^t \vec{v}). \quad (8)$$

One can notice the use of the mass flux (nodal value) instead of velocity in Equation (4) in order to fasten the mass flow conservation. The choice of the boundary conditions account for local direction of characteristic lines :

- outflow \Leftrightarrow Neumann
- inflow \Leftrightarrow Dirichlet.

The time step is independent for each task. Moreover, we deal with overlapping and nonoverlapping cases. To increase stability and convergence, we introduce a relaxation process on the values (stresses and variables) imposed on the coupled boundaries. For the overlapping case, we perform linear interpolations or direct affectations of fields on the coupled boundaries. Here, one uses DDM as local preconditioner (\equiv Jacobi) instead of partitioned version of CGM.

7 NUMERICAL TESTS

7.1 Tests definition

In the background field of the SG simulation, we choose the CEA CLOTAIRE mock-up, Campan and Bouchter (1988). The riser part forms an half cylinder of 0.62 m in diameter and 9.16 m in height. The inside is filled with U shaped tube bundle, 7.2 m in height, in which travels the hot primary flow. One

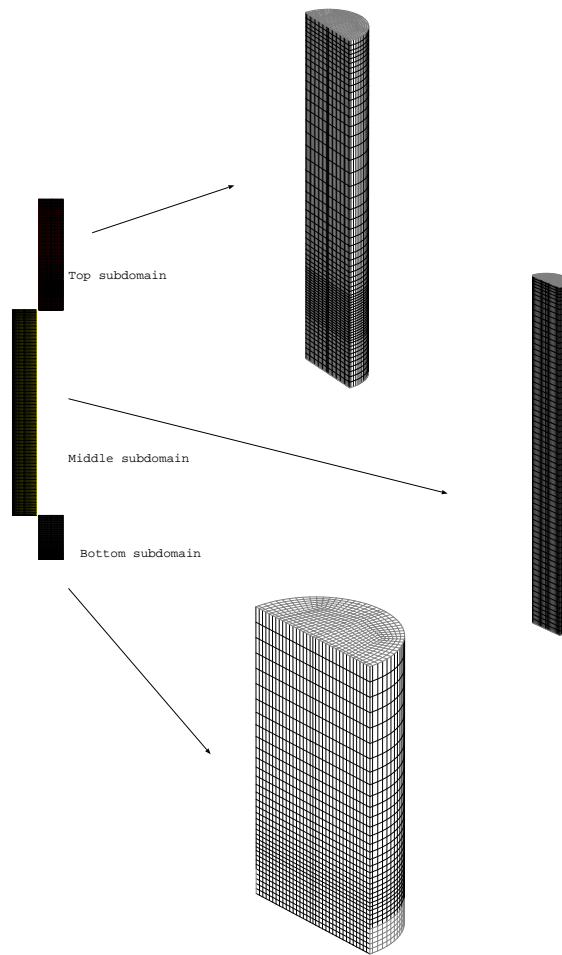


Fig. 2. Three subdomains partition meshing

flow distribution baffle, nine tube support plates and one anti-vibrating bar are respectively located at the bottom, straight and curved part of the tube bundle.

7.2 Domain partition and meshing

Three partitions were tried. For all cases, the domain carving is following the mean flow direction.

- The first one involves three non overlapping subdomains : one for the incoming flow region (named bottom), one for the straight part (named middle) and one for the tube bend plus the exit flow region (named top). In each subdomain the meshing is the best available, so the assembly is non conforming, see Fig. 2. The spatial resolution of subdomains one and three are increased, so the cubic root of the mean cell volume is of order of the pitch of the tube bundle. The total number of cells is around 70,000.

- The two other partitions involve two subdomains (named top and bottom) and are similar considering the number of cells ($\approx 7,000$) and the conforming of the global grid, but they differ by the existence of overlapping.

7.3 DDM numerical features

In each case, one processor is devoted to one subdomain. So, the required total memory is spread over the processors. For instance, 480 Mo is needed in the case of the three subdomains partition. Like this is around half of the total memory capacity available for all the computer's users, running the simulation is not an easy task. In contrast, the situation is well better when asking 240, 120 and 120 Mo for three processors. For nonoverlapping subdomains, the relaxation coefficient for the coupled boundaries is equal to 0.4. The interface conditions are applied each 10 outer iterations (\equiv time steps).

7.4 Results

7.4.1 Three subdomains partition

Here, one compares the DDM results with the full domain computation ones. Physical models and numerical parameters are identical. Asking a relative difference error of $5 \cdot 10^{-3} s^{-1}$, the steady state is reach in 681 time steps.

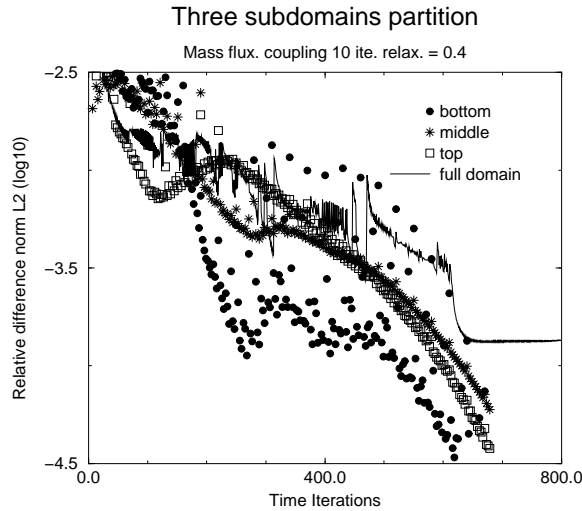


Fig. 3. Convergence history of the variables for the three subdomains partition

The full domain simulation needs around 1660 time steps to verify the same criteria, see Table 1. Consequently, the amount of saved time iterations is about 60 % due to DDM preconditioning. The Figure 3 shows the relative

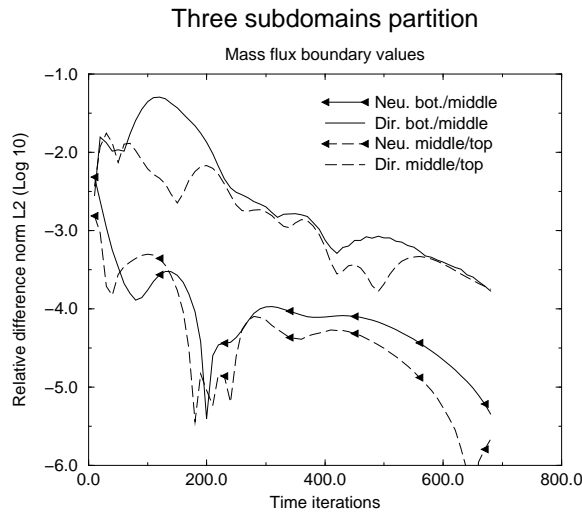


Fig. 4. Convergence history of the boundary values (mass flux) for the three subdomains partition

difference evolution of \vec{G} (the quantity with the lower convergence rate) : $|\vec{G}^{n+1} - \vec{G}^n|/|\vec{G}^n|$. Here n is the outer iteration number and the norm is the L2 one.

For the coupled boundaries, one shows similar quantities on Figure 4, but 'n' is now the coupling iteration number. For Neumann coupled boundaries, the mass flux is replaced by the stress $\bar{\sigma}$ (see Equation 8).

The CPU time for the slowest task (\equiv top, tube bend region) is 12655 s on a SUN UltraSparc 2, 200 MHz. By comparison to the overall computation, one saves 70 % of the CPU time ! With identical time step number, the speed-up and the parallel efficiency are around 2 and 0.65 respectively. The CPU overhead is about 20 % per time step, that is reasonable. It must be emphasized that this CPU overhead may penalize the computation if the coupling period value is too much decreased. In the tube bend region, the DDM and the full domain computations are very similar with in particular the same time step. But in consequence of increased time steps in the upstream regions, the coupled boundary get more rapidly right values in the DDM case, see Table 1 (Δt and \diamond).

Comparing some global quantities on the riser exit area and the variable fields themselves in each subdomain, we can quantify the accuracy of the DDM results. Concerning the variables, the L2 norm relative difference ($|X_{fd} - X_{ddm}|/|X_{fd}|$ where X holds for variables and fd for full domain simulation) is maximal in the straight part region for the mass flux and the specific enthalpy (about 4.10^{-2} and 10^{-3} respectively) and in the incoming flow region for the pressure (around 4.10^{-4}), see Table 2. Concerning the global quantities

on the exit, the relative differences for the mean static quality, the mean void fraction and the gas flow rate are less than 10^{-2} . In respect to the mixture flow rate, the relative difference is around 5.10^{-4} .

****Here: Table 1 ****

All in all, we consider these results as satisfactory with regard to analysis of industrial SG. In addition, it is difficult to evaluate the part of the flow instability in these results. For instance, the points associated with the maximal relative mass flux difference are located in flow mixing regions such as downstream from the flow distribution baffle or near the boiling front.

****Here: Table 2 ****

7.4.2 *Two subdomains partition*

The results concerning the two nonoverlapping subdomains partition lead to similar conclusions: outer iteration saving $\in [30; 60]$ %, CPU time saving $\in [50; 80]$ %, efficiency $\in [50; 65]$ %.

So, one concentrates now on the simulation features of the overlapping subdomains case. It can be viewed as an extension of the iterative method used for nonoverlapping cases. Our test computations of the mock-up with two subdomains partition have shown that : the overlapping increases the computation robustness and that the discrepancies in the overlapping region are weak, see Fig. 5.

For instance and concerning the computation robustness, the overlapping case allows us to converge even if the initial spatial distribution of the specific enthalpy are discontinuous at the subdomain's interface. This is a severe test because enthalpy variation leads to mixture velocity and pressure variations through the density ones. With nonoverlapping subdomains partition, the same computation needs under relaxation (interface condition application) to converge. If not, a flow inversion occurs and the computation falls.

8 LOCAL ZOOM COMPUTATION

We want to perform the computation of a local subdomain with a fine mesh and, in the same time, to use this high resolution information in order to increase the accuracy of the simulation of the whole domain. To doing this, we use local hierarchical meshes (local multigrid) and the Local Defect Correction method.

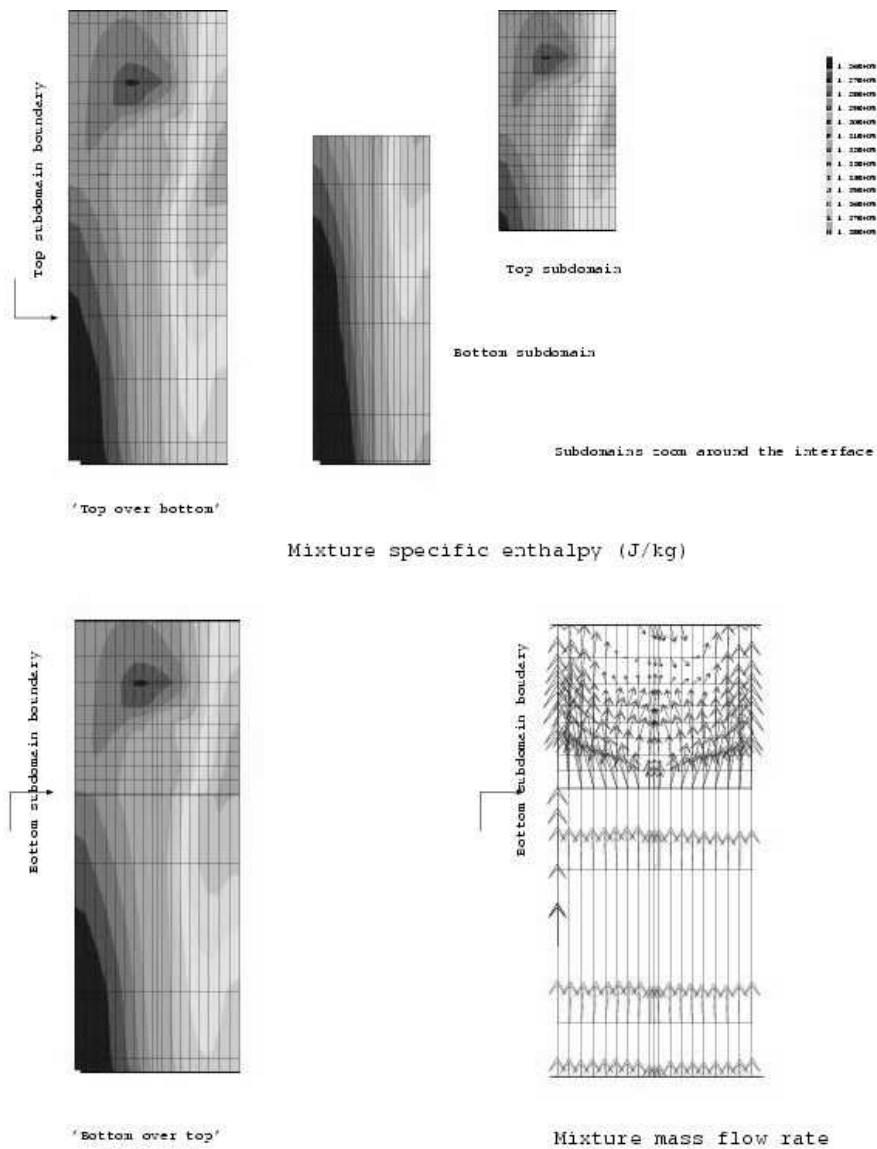


Fig. 5. Two overlapping subdomains partitioning meshing

8.1 The Local Defect Correction method

8.1.1 Overview

We want to reach a steady state flow simulation by a transient computation. Consequently, there are time terms in the balance equations. The used semi implicit approach leads to a splitting of the equation terms into an operator part (left-hand side, implicit) and a source part (right-hand side, explicit). Actually, the main purpose of this time scheme is to overcome the non linear feature (relaxation process). Moreover, the source terms in the balance equations are dependent of the unknown variables.

In the introduced LDC application, we do not take into account the time terms, focusing only on the steady state equations to solve. Hence, we prefer take into account equations with the following form : $T(u) = 0$.

The goal of the LDC method is to locally correct the discretized operator, says T_i , involved in the numerical resolution of a discretized equation on a given mesh $M_i : T_i(u_i) = 0$. For a fixed discretization order, the error comes here from the space discretization (space step).

Basically, we deal with two discretizations of the operator T . A coarse one (or standard one, T_1) in the whole computation domain, says D_1 , and a fine one, T_2 , in the zoom region (improvement in the space step), says D_2 . Generally, the coarse discretization allows an easy numerical resolution, but may introduce large errors in some particular parts of the computation domain. At the opposite, the numerical resolution is enhanced with the fine discretization, but more expensive. Hence, we only may compute a part of the whole domain. The strategy is to solve the local problem with the best discretization and locally correct the standard expression of the operator in the whole domain. To do this, an iterative process is applied.

For an detailed presentation of the LDC method, see the Hackbusch's original paper (Hackbusch, 1984). In the following section, we present the modified LDC method used in the GENEPI software, including the non linear feature as well as possible.

8.1.2 The application of the LDC method in GENEPI

Let M_1 be the mesh associated with the standard space discretization of the global domain D_1 and M_2 the mesh associated with the local problem D_2 (finer discretization). And let B_1 be the boundary of M_1 and B_2 be the boundary of M_2 . The local mesh and the global mesh are embedded forming a hierarchical grid structure. The internal boundary is given by $B_2 \setminus B_1 \cap B_2$. Let A be the coarse nodes included in the zoom region of the M_1 mesh and \hat{A} be the internal nodes of A .

The iterative process is the following. Let χ be the characteristic function of \hat{A} and u_1^0 the initial approximation of the solution on the mesh M_1 .

- start with $f_1^0 = 0$,
- given f_1^i , $i = 0, 1, 2, \dots$
 - do some smoothing iterations for the global problem $T_1(u_1^i) = f_1^i$ on M_1 ,
 - compute the internal boundary values on M_2 , by interpolation for instance,
 - do some smoothing iterations for the local problem $T_2(u_2^i) = 0$ on M_2 ,
 - compute the restriction of u_2^i (called ru_2^i) on nodes of A (in M_1),
 - define the next right-hand side by :

$$f_1^{i+1} = \chi T_1(ru_2^i).$$

We add the LDC terms to the right-hand part of the secondary flow energy balance equation and momentum balance equation. No correction is performed on the pressure equation of the Chorin algorithm (the LDC term is kept constant during a time step). Here, u_j^i stands for the couple (H, Q) for the energy balance equation and for (\vec{G}, P) for the momentum balance equation.

Basically this algorithm may be viewed like a classical LDC iterations loop in which we do not solve exactly the local and global problems, but only perform few iterations (nc) of one iterative method: the usual numerical techniques of Genepi, namely outer iterations for the non linear terms (transient computation with a semi-implicit approach) and a Conjugated Gradient type method for the linear system solving. It can be view in terms of classical outer iterations (i.e. time iterations) with periodic updates of the defect correction values for the global problem and of the boundary values for the local one.

We have presented the multiplicative version (sequential one) of the algorithm, but we have also developed an additive version (parallel one). In all cases, one slave Genepi task runs for the global domain computation and several others for the zoom subdomains computations (one by subdomain).

Data are sent from the zoom subdomain to the global one for the purpose of the filling of the defect correction values. Following the type of the balance equation, the sent data are specific enthalpy, mass flux, volume thermal source or pressure. Then, by mean of canonical restrictions (for variables) or volume weighted restrictions (for thermal sources and pressures), we restrict these data on all nodes of the global domain mesh included in the zoom region A (except at the Dirichlet nodes). The local defect corrections values are computed for the nodes of the global mesh included in the correction region, \hat{A} . These nodes, included in A, are distant of the internal boundary nodes by more than one or two elements. The extension of this region is managed by the code users.

8.1.3 *The correction term for the energy balance equation.*

The correction term is $T_1(ru_2^i)$ where ru_2^i stands for the restriction on the global domain of the mixture specific enthalpy (noted : Hr), of the mixture pressure (noted : Pr) and of the mixture mass flux (noted : \vec{Gr}) initially computed on the zoom subdomain. The counterpart of the symbolic operator $T_1(ru_2^i)$ is the finite element formulation of the following term (in case of the drift flux hypothesis), on the mesh of the global domain :

$$\begin{aligned}
& \int \overline{dX} \varphi^i [\beta (\overline{Gr} \cdot \overline{\nabla}) Hr + \text{div}(\beta x_r (1 - x_r) \rho_r L_r \overline{v_{Rr}})] \\
& \quad - \int \overline{dX} \varphi^i \beta Q_r \\
& \quad + \int \overline{dX} [(\overline{\nabla} \varphi^i) (\beta \chi_T \overline{\nabla} Hr)] \tag{9}
\end{aligned}$$

with $L_r(Pr)$: latent heat, $x_r(Hr, Pr)$: static quality, $\rho_r(Hr, Pr)$: mixture density, $\overline{v_{Rr}}(Hr, \overline{Gr}, Pr)$: relative velocity, Q_r : source term including the restricted volume thermal source, the boundary thermal flux and, possibly, the pressure gradient terms. If exists, this last term is a function of the restricted variables (Hr, \overline{Gr}, Pr).

Concerning the non linear feature, the weakness lies in the values of χ_T , computed with the variables taken at the current outer iteration (i.e. time iteration), and of the volume thermal source (use of the restricted volume thermal source).

8.1.4 The correction term for the momentum balance equation.

The correction term $T_1(ru2_1^i)$, with the same previously defined restricted variables, is the finite element formulation of the following expression (in case of the drift flux hypothesis) :

$$\begin{aligned}
& \int \overline{dX} \varphi^i [\beta (\overline{Gr} \cdot \overline{\nabla}) \overline{v\hat{r}} + \text{div}(\beta x_r (1 - x_r) \rho_r \overline{v_{Rr}} \otimes \overline{v_{Rr}})] \\
& \quad - \int \overline{dX} \varphi^i [\beta \rho_r \overline{g} - \beta \bar{\Lambda}_r \overline{Gr}] - \int \overline{dX} \overline{\nabla} (\beta \varphi^i) Pr \\
& \quad + \int \overline{dX} \overline{\nabla} \varphi^i [\beta \mu_T (\overline{\nabla} \overline{v\hat{r}} + \overline{\nabla}^t \overline{v\hat{r}})] \\
& \quad \quad \quad + BC \tag{10}
\end{aligned}$$

with the same notations than above, $\bar{\Lambda}_r(Hr, \overline{Gr}, Pr)$ is the total two-phase friction tensor, $\overline{v\hat{r}} = \overline{Gr} / \rho_r$ and BC is the boundary stress.

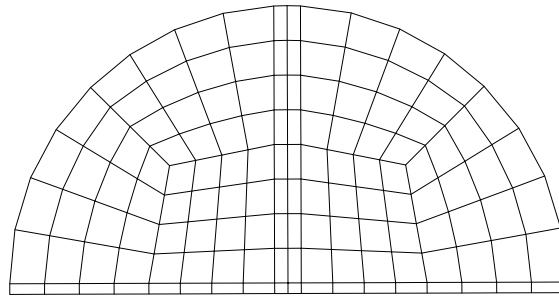
The weakness concerning the non linear feature lies only in the values of μ_T , computed with the current iteration variables and not with the restricted ones.

9 NUMERICAL TESTS

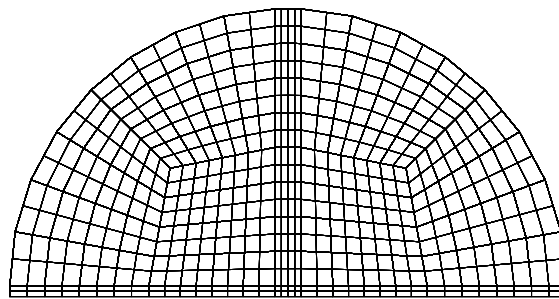
9.1 Tests definition

We choose to compute a zoom simulation of the U-tube bundle region, above the last tube support plate, of the CEA CLOTAIRE mock-up previously presented (DDM test cases). We know the existence of fluid vortices located in the bundle wake.

We want compare the LDC zoom results (denoted LDC) with post-processing (or chained) Zoom computations, denoted Post. By post-processing, we mean a computation in two independent and successive steps : first, the computation of the global domain and, second, the computation of the zoom subdomain. We also want compare LDC zoom results with global domain simulations at low and high space resolutions (respectively denoted Ref0 and Ref1).



130 cells



520 cells

Fig. 6. Coarse and fine meshes (sections)

9.2 LDC zoom and meshing

Concerning the global domain, we use two meshes. The first one, denoted the global coarse mesh, is composed of 5330 cells. There are 780 hexahedral elements in the zoom region. The second one is denoted the global fine mesh (each cell of the coarse is split in eight), see Fig. 6. This fine mesh is made of 42640 cells.

The mesh concerning the zoom subdomain is made of 6240 cells (it is a part of the global domain fine mesh). It is a slice of the riser with walls, incoming and out-coming areas, see Fig. 7.

9.3 Numerical features

The physical models and the numerical parameters for the internal iterative method (here, Preconditioned Conjugated Gradient) are identical for all the simulations. In particular, the stationary criteria is $5.10^{-3}s^{-1}$ for the relative difference error.

The boundary conditions for the global domain computations are the usual ones. Concerning the zoom computations, there are specific boundary conditions for the exits. Due to the presence of vortices across this section, we use either Dirichlet conditions for the specific enthalpy and the mass flux (inner nodes of the exit surface), either Adaptive Dirichlet Neumann (ADN) ones, see Quarteroni (1990).

For the LDC computations, the momentum and the energy balance equations on the global domain and on the zoom subdomain are coupled. Concerning the primary flow energy balance equation, the zoom inlet primary flow temperature and the zoom outlet thermal flux are coupled with the global domain (as in DDM). The numerical parameters for the LDC computations are : coupling period (nc) = 20 outer iterations and relaxation parameter = 0.5. The number of corrected nodes in the global computation are 750 nodes (but, we have done computations with several numbers of corrected nodes).

9.4 Results

We perform the coupled simulations with two Genepi tasks using a bi-processors computer (Sun Enterprise 2, model 2002, 200 MHz). The performances concerning the CPU time and the memory are listed in Table 3.

**Here: Table 3 **

We can do some comments. The memories used in the LDC case and Chained case are similar. They deeply differ of the memory needed in the case of Ref1. It is the same for the CPU time comparison. Moreover, with parallel computations, the LDC method leads to lower CPU time. Hence, it is possible to increase locally the accuracy of the computation (see below) without too much increase the cost.

Concerning the CPU time spends for the coupling, the LDC iterations on the global domain or the updates of the boundary condition on the zoom domain cost about 10% of the total CPU time.

Fig. 8 plots the convergence history of the mass flux variable for the global domain LDC computations and the reference computations (Ref0 and Ref1).

About 25 correction steps were applied. Beyond 200 outer iterations, the correction values are stabilized. At the end of the computation, the discrete L2 norm of the energy correction is about 3.79 kW. To compare, the exchanged power in the U-tube bundle region is about 70 kW.

Fig. 9 plots the convergence history of the zoom subdomain variables. Clearly, the computation convergence is strongly affected by the refreshments of the coupled boundary values. In fact, the goal of the relaxation parameter (here 0.5) is to partially damp these oscillations (a lower value not increases deeply the damping). Some additional tests showed that the decrease of the coupling period (for instance, $nc = 5$) leads to a better convergence history with identical physical results. However, the CPU times are increased.

The figure Fig. 10 shows the correction fields of the LDC computation for,

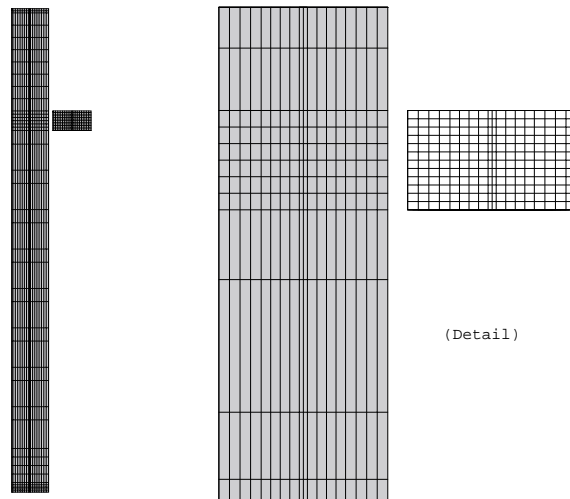


Fig. 7. Coarse and fine meshes for the zoom area

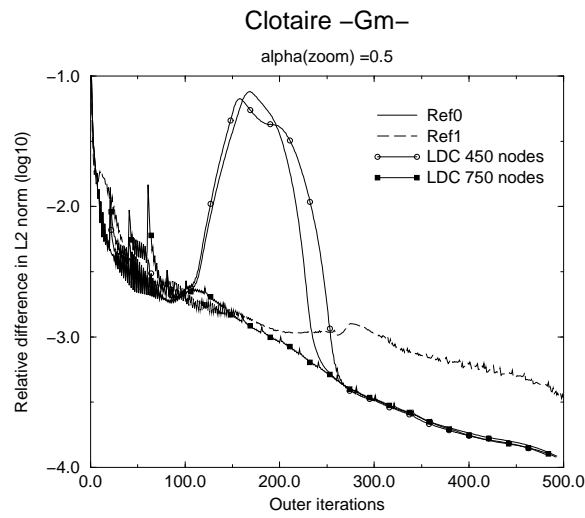


Fig. 8. Mass flux convergence history, global domain

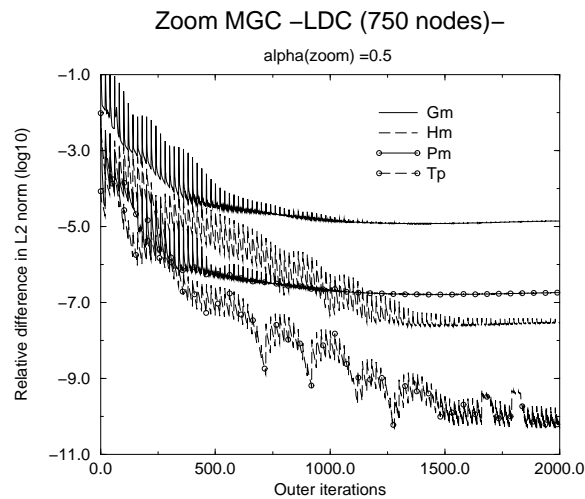


Fig. 9. Variable convergence history, zoom domain

respectively, the energy equation and the momentum equation (cut at 5 cm inside the mock-up). The energy correction terms are located at the frontier of the U-tube bundle and the momentum correction terms outside the anti-vibration bars.

We have performed sequential computations for the same test case. Results are similar, but obviously the CPU time significantly increases.

9.5 Physical features

Concerning some global quantities, like the exchanged power, the steam mass flow rate, ..., all computations give similar results. For instance, the exchanged

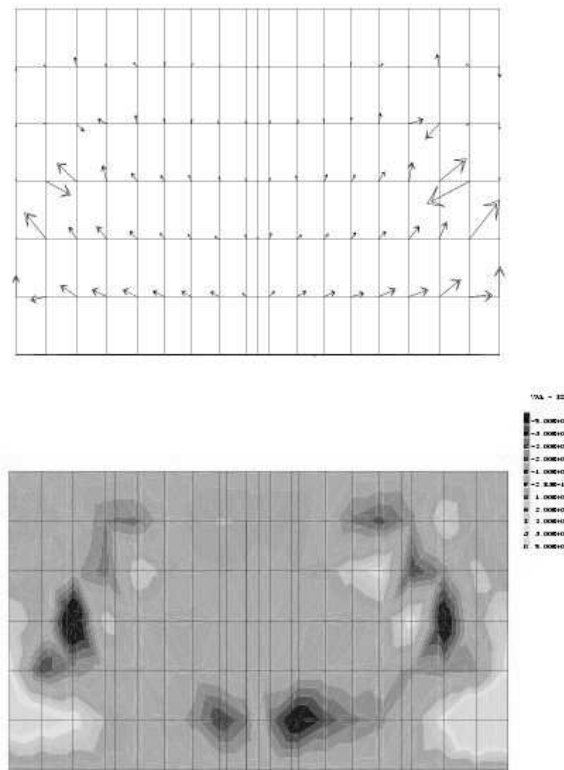


Fig. 10. LDC corrections

power, in the LDC computation, is slightly overestimated (about 0.4% in comparison with the Ref1 case).

In fact the improvement obtained with the LDC method is particularly in the local variation of the variables in the global domain. On Fig. 11, we show the enthalpy distribution along some vertical lines in the cold leg for the coarse and fine meshes computations alone and for the LDC computation. In particular, the iso-values of the specific enthalpy inside the correction region are closer to the Ref1 results than the Ref0 ones, even if some under-estimations appear. The enthalpy profiles computed with the LDC on zoom or global domain are very close and both present the same variations than those observed on the Ref1 results. Obviously, the better are the coarse grid results in the entry of the zoom, the better are the zoom predictions.

We have performed computations with Adaptive Dirichlet Neumann boundary conditions for the same test case once. Again, the results are very similar.

10 CONCLUSION

Finally, we have proved the efficiency of an algorithm based on an additive version of the Dirichlet-Neumann method in the simulation of SG two-phase

Clotaire Cold Leg (LDC / ADN)

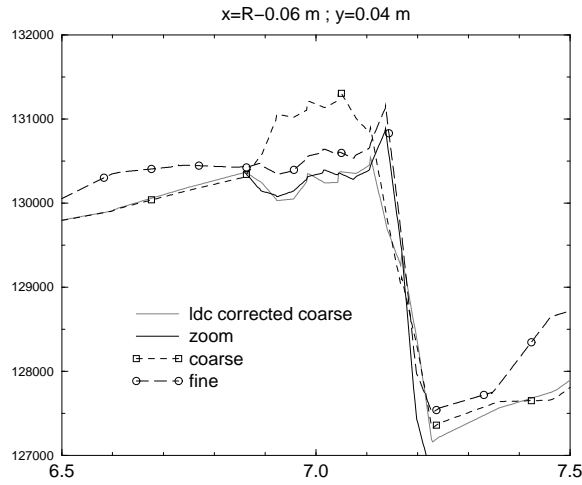


Fig. 11. Specific enthalpy in cold leg

flows. We take advantage of parallelism and preconditioning. It works with overlapping or nonoverlapping partition and the global meshing can be non conforming. The tests performed in industrial context conclude to the save of a large amount of time iteration number and CPU time. Accordingly, the usual mesh cells number ($\approx 30,000$) can be easily increased to 100,000 on a workstation network. This work is completed with an adapted method to face an eventual vortex presence at subdomain interfaces, like the Adaptive Dirichlet Neumann method (Quarteroni, 1990), and with DDM computation of primary fluid energy balance. We must outline the interest of code-linker like the CEA one (ISAS) to parallelize tasks using a network, without code re-engineering. However, this choice involves limitations on parallel computation capacities. To illustrate this point, the Amdahl's law allows the estimation of the code parallel part and of the maximal reachable speed-up. We found values around 70 to 90 % and 4 to 8, respectively (depending of the problem size). If we compare with massively parallel machine capacities, this speed-up value is low, but is enough for the planned applications.

Moreover, the implementation is versatile enough to extend to equations coupling or to local adaptive zooming methods for flow simulation around obstacle. Concerning this last point, we have proved the efficiency of an additive version of a zooming algorithm based on the LDC method in the simulation of SG two-phase flows. The LDC computation allows the enhancement of the software prediction in the correction region :

- the behavior of the variables is between the result of the computations with coarse and fine mesh,
- the coherence between the zoom subdomain results and the global domain ones is enhanced.

The LDC computation cost (zoom and global domains) is gently higher than a post chained computation and, obviously, very cheaper than the global domain one with the fine mesh.

This work consists in a first attempt to bring modern numerical tools in the context of industrial simulations of SG two-phase flows. Future improvements are planned. In particular, we can mention the development of a version of the Flux Interface Correction method (FIC) for the zoom computations, see Angot et al. (1992) or Angot et al. (1993), and FAS methods for multigrid preconditioning (Brandt, 1977).

Acknowledgements

We would like to thank the Framatome Company for their financial support.

References

- Angot, P., Caltagirone, J. P., Khadra, K., 1992. Une méthode adaptative de raffinement local: la correction du flux à l'interface. *C. R. Acad. Sci. Paris, Sér. I* 315, 739–745.
- Angot, P., Caltagirone, J. P., Khadra, K., 1993. A comparison of locally adaptive multigrid methods: L.D.C., F.A.C. and F.I.C. In: NASA Conf. Publ.CP-3224. Vol. 1. pp. 275–292.
- Brandt, A., 1977. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 31 (138), 333–390.
- Campan, J., Bouchter, J., April 1988. Steam generator experiment for advanced computer code qualification : CLOTAIRE international program. In: Third International Topical Meeting on nuclear power plant thermohydraulics and operations. Seoul.
- De Gramont, T., Toumi, I., 1996. ISAS user's guide. Tech. Rep. SERMA/LETR/96/1955, CEA, Saclay, France.
- Grandotto, M., Bernard, M., Gaillard, J., Cheissoux, J., De Langre, E., November 1989. A 3d finite element analysis for solving two phase flow problems in PWR steam generators. In: 7th International Conference on Finite Element Methods in Flow Problems. Huntsville, Alabama, USA.
- Grandotto, M., Oby, P., 1996. Calculs des écoulements diphasiques dans les échangeurs par une méthode aux éléments finis. *Revue Européenne des Eléments Finis* 5 (1), 53–74.
- Gresho, P., Chan, S.J. Lee, R., Upson, C., 1984. A modified finite element method for solving the time dependent, incompressible Navier Stokes equations (Part 1: Theory). *Int. J. Num. Methods in Fluids* 4, 557–598.
- Gresho, P., Chan, S., 1990. On the theory of semi implicit projection methods for viscous incompressible flow and its implementation via finite element method that also introduces a nearly consistent matrix. *International Journal for Numerical Methods in Fluids* 11 (5), 587–659.
- Gresho, P., Lee, R., Sani, R., 1978. Advection-Dominated Flows with emphasis on the consequences of mass lumping. *Finite Elements in Fluids* 3, 335–350.
- Hackbusch, W., 1984. Local Defect Correction Method and Domain Decomposition Techniques. Vol. 5 of *Computing Suppl.* Springer-Verlag (Wien), pp. 89–113.
- Hughes, E., Chen, F., 1977. Transient three dimensional thermalhydraulic analysis of homogeneous two phase flows in heat exchangers. In: AICHE National Heat Transfer Conference, PWR Steam Generators.
- Le Tallec, P., 1994. Domain decomposition methods in computational mechanics. Vol. 1 of *Computational Mechanics advances.* North-Holland, Elsevier Science, pp. 121–220.
- Oby, P., Cheissoux, J., Grandotto, M., Gaillard, J., De Langre, E., Bernard, M., November 1990. An advanced steam generators design 3d code. In: ASME Winter Annual Meeting. Dallas, Texas, USA.
- Quarteroni, A., 1990. Domain decomposition method for the numerical solu-

tion of partial differential equations. Tech. Rep. UMSI90/246, Supercomputer Institute, University of Minnesota.

List of Figures

1	French nuclear steam generator (scheme)	2
2	Three subdomains partition meshing	7
3	Convergence history of the variables for the three subdomains partition	8
4	Convergence history of the boundary values (mass flux) for the three subdomains partition	9
5	Two overlapping subdomains partition meshing	11
6	Coarse and fine meshes (sections)	15
7	Coarse and fine meshes for the zoom area	17
8	Mass flux convergence history, global domain	18
9	Variable convergence history, zoom domain	18
10	LDC corrections	19
11	Specific enthalpy in cold leg	20

List of Tables

1	Efficiency of parallel DD computation for three subdomains partition.	26
2	Accuracy of parallel DD computation for three subdomains partition.	26
3	Efficiency of the parallel LDC zoom computation.	26

Table 1

Efficiency of parallel DD computation for three subdomains partition.

Domain	# iterations	CPU (s)	$\Delta t \cdot 10^2$ (s)	\diamond
Top	681	12655	0.7	2.1
Middle	622	9133	2.1	2.3
Bottom	550	7416	1.5	4.5
Full domain	1660	64698	0.75	1.1

 \diamond Final time/time need to flow the domain

Table 2

Accuracy of parallel DD computation for three subdomains partition.

Subdomain	\vec{G} (%)	H (%)	P (%)
Top	0.7	0.06	0.02
Middle	3.8	0.11	0.01
Bottom	2.7	0.08	0.04

Table 3

Efficiency of the parallel LDC zoom computation.

Simulation name	# iterations	CPU (s)	Memory (Mo)
Ref0	493	1460	20
Ref1	1079	25460	280
Ref0 + Post	618	3180	60
LDC	492	2100	76