INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 247–260 (1999)

AN EFFICIENT PARALLEL ALGORITHM FOR THREE-DIMENSIONAL ANALYSIS OF SUBSIDENCE ABOVE GAS RESERVOIRS

B.A. SCHREFLER[a,*], X. WANG[b], V.A. SALOMONI[a] AND G. ZUCCOLO[a]

[a] *Department of Structural and Transportation Engineering, Faculty of Engineering, via F. Marzolo, 9-35131 Padova, Italy*
[b] *Research Institute of Engineering Mechanics, Dalian University of Technology, Dalian 116023, China*

SUMMARY

In this paper an efficient parallel algorithm to solve a three-dimensional problem of subsidence above exploited gas reservoirs is presented. The parallel program is developed on a cluster of workstations. The parallel virtual machine (PVM) system is used to handle communications among networked workstations. The method has advantages such as numbering of the finite element mesh in an arbitrary manner, simple programming organization, smaller core requirements and computation times. An implementation of this parallel method on workstations is discussed, the speed-up and efficiency of this method being demonstrated by a numerical example. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: parallel algorithm; frontal solution; subsidence due to fluid withdrawal; deforming porous media

1. INTRODUCTION

Surface subsidence due to extraction of underground fluids (water, hydrocarbons) plays an important role in reservoir engineering. In recent years a great deal of attention has been directed towards this phenomenon, also because it affects historical cities, like Venice and Ravenna in Italy [1–5]. Subsidence analyses are computationally intensive because they involve problems of regional scale and very long time spans, e.g. in the case of the Groningen gas field, subsidence predictions for the year 2050 have been made from the year 1973 on.

Moreover, there may be several reservoirs at different levels distributed over a large area with possible interaction. A typical case is the upper Adriatic region, where the depth of the location of the pools ranges between 900 and 4000 m and the horizontal area involved is about 19000 km^2. In addition, the different pools are not put in production at the same time, which complicates the situation further. Thus, in general, a subsidence analysis is usually very time consuming, also because of the number of interacting fields (variables) involved. Hence, even if the problem is a three-dimensional one, the major part of studies is still mainly limited to one- or two-dimensional analyses. However, lower-dimensional models cannot always simulate these problems correctly. Thus, it seems necessary to construct and solve three-dimensional models. One such model exists [6], but it is rather expensive for gas reservoirs, where a more convenient approach can be used [7].

* Correspondence to: Department of Structural and Transportation Engineering, Faculty of Engineering, via F. Marzolo, 9-35131 Padova, Italy.

Recent developments in parallel computing has enabled rapid linear, non-linear and dynamic analysis of complex structures [8–12]. This same computational power may be utilized for efficient three-dimensional subsidence analyses. In this paper, a multi-level frontal parallel algorithm has been developed for such an analysis. The given domain of the problem is discretized into multi-level substructures. High-level substructures, named *super-elements*, are composed of lower level substructures. Multi-wavefronts are first used to assemble and eliminate variables in the lowest level substructures. The contributions for super-elements are obtained when every wavefront comes to the boundary of each substructure. Then, multi-frontal procedures go up to higher level super-elements, and so on up to the highest level super-element. Finally, the global interface equations are solved. Once the values corresponding to the highest level super-element have been obtained, the solutions of the super-elements or substructures at other levels may be obtained through back-substitution routines level by level. In every super-element or substructure at the same level, multi-frontal procedures are independent and can be carried out in parallel. A similar concept was developed also by Geng *et al.* [12].

The basic formulation of the problem is briefly recalled, then the details of the parallel computing strategy employed for its implementation on a cluster of workstations are given. A numerical example is presented, with speed-up resulting from the parallel strategy.

2. THE PROBLEM OF SUBSIDENCE ABOVE GAS RESERVOIRS

The particular subsidence problem solved here is first briefly summed up as follows [5,7]. It is supposed that there are several gas reservoirs at different levels, and some of the reservoirs have an edge aquifer. It is further assumed that in each respective domain there is only one fluid: water in the aquifers and gas in the reservoirs. The gas, upon exploitation of the reservoirs, may be substituted by encroaching water, which comes from the edge aquifers and from possible leaky aquitards. Capillary effects due to the simultaneous presence of gas and water in the reservoirs are hence neglected. They may play a significant role [13] and will be introduced in the future. With these assumptions, we have the following balance equations [7], where the chosen macroscopic field variables are displacement **u** and water pressure p_w.

Linear momentum balance equation for the mixture solid + water or solid + gas

$$\nabla^\mathrm{T}\sigma + \rho\mathbf{g} = 0, \tag{1}$$

where σ is the total stress vector, **g** is the acceleration of gravity and ρ is the averaged density of the mixture:

$$\rho = (1-\phi)\rho_\mathrm{s} + \phi\rho_\pi, \tag{2}$$

where $\pi = \mathrm{w}$ (water) or g (gas), s = solid, ϕ being the porosity.

Flow conservation equation for the aquifers and aquitards

$$-\nabla^\mathrm{T}\left\{\frac{\mathbf{k}}{\mu}\nabla(p_\mathrm{w} + \rho_\mathrm{w}gh)\right\} + \left(\mathbf{m}^T - \frac{\mathbf{m}^\mathrm{T}\mathbf{D}_\mathrm{T}}{3K_\mathrm{s}}\right)\frac{\partial\varepsilon}{\partial t} + \left[\frac{(1-\phi)}{K_\mathrm{s}} + \frac{\phi}{K_\mathrm{w}} - \frac{1}{(3K_\mathrm{s})^2}\mathbf{m}^\mathrm{T}\mathbf{D}_\mathrm{T}\mathbf{m}\right]\frac{\partial p_\mathrm{w}}{\partial t} = 0, \tag{3}$$

where **m** is a vector with components equal to unity for the normal stress components and zero for the shear stress components, K_w is the bulk modulus, K_s is the averaged bulk modulus of the solid grains, **k** is the absolute permeability matrix of the medium, $\mathbf{D}_\mathrm{T}$ is the tangent matrix, ε is the total strain of the skeleton, and μ is the dynamic viscosity of the water.

Instead of writing a similar mass balance equation, as Equation (3), for the gaseous phase, we consider its integral form for the whole reservoir volume. This is valid if the reservoir volume is small compared with the analysed cross-section and its thickness is small compared with the depth of burial. The conservation equation hence assumes the form of a

Material balance equation for the reservoir

$$GB_{gi} = (G - G_p)B_g + W_e - W_pB_w, \tag{4}$$

where G is the initial free gas in place, B_g is the gas formation volume factor, G_p is the cumulative gas production, W_p is the cumulative water production and W_e is the influx from the adjacent aquifer and from leaky aquitards; the index i denotes initial conditions. In incremental form, the above equation can be written as

$$(G - G_p)\Delta B_g + \Delta W_e = \Delta G_pB_g + B_w\Delta W_p + \Delta B_w W_p. \tag{5}$$

The formation volume factor for natural gas [14] is obtained from the state of gas equation as

$$B_g = \frac{p_{sc}}{T_{sc}}\frac{ZT}{P} \tag{6}$$

and

$$\Delta B_g = \frac{p_{sc}}{T_{sc}}\left(\Delta Z\frac{T}{P} - \frac{ZT}{P^2}\Delta P + \frac{p_{sc}}{T_{sc}}Z\Delta T\right), \tag{7}$$

where $Z(P, T)$ is the compressibility factor, T the absolute temperature and sc stands for surface conditions.

Equations (4) and (6) yield at each time step gas pressure p_g, when gas and water inflow are known. This gas pressure is applied in the reservoir volume as a boundary condition and the whole subsiding domain is then analysed using the fully coupled Equations (1)–(3). Equations (4)–(7) are necessary when the gas production is given as an input. Sometimes the gas pressure decline in time is given either from *in situ* measurements or from separate flow analyses. In that case, these pressures can be used directly as boundary conditions in the reservoir. It is customary to assume that the pressures in the reservoir or large parts of it are constant in space.

A generalized Galerkin procedure [5,15] is introduced to discretize the governing equations (1)–(3), yielding the following system of equations

$$\begin{aligned} &\mathbf{c}_{uu}\dot{\mathbf{u}} + \mathbf{c}_{uw}\dot{\mathbf{p}}_w + \mathbf{f}_u = \mathbf{0},\\ &\mathbf{c}_{wu}\dot{\mathbf{u}} + \mathbf{c}_{ww}\dot{\mathbf{p}}_w + \mathbf{k}_{ww}\mathbf{p}_w + \mathbf{f}_w = \mathbf{0}, \end{aligned} \tag{8}$$

All matrices are listed in Appendix B.

Discretization in the time domain is accomplished through approximating $\mathbf{u}$, $\mathbf{p}_w$ with a linear variation within each time step Δt

$$\{\mathbf{u}, \mathbf{p}_w\} = \{1 - \alpha, \alpha\}\begin{bmatrix} \mathbf{u}^t & \mathbf{p}^t \\ \mathbf{u}^{t+\Delta t} & \mathbf{p}^{t+\Delta t} \end{bmatrix}, \tag{9}$$

where

$$\alpha = (t - t_n)/\Delta t_n. \tag{10}$$

The point collocation method at the $(n+1)$th time step yields the following matrix equation

$$\mathbf{K}\mathbf{X}_{n+1} = \mathbf{F}, \tag{11}$$

where

$$\mathbf{K} = \begin{bmatrix} \mathbf{c}_{\mathrm{uu}} & \mathbf{c}_{\mathrm{uw}} \\ \mathbf{c}_{\mathrm{wu}} & \mathbf{c}_{\mathrm{ww}} + \alpha\Delta t\mathbf{k}_{\mathrm{ww}} \end{bmatrix}, \tag{12}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{c}_{\mathrm{uu}} & \mathbf{c}_{\mathrm{uw}} \\ \mathbf{c}_{\mathrm{wu}} & \mathbf{c}_{\mathrm{ww}} - (1-\alpha)\Delta t\mathbf{k}_{\mathrm{ww}} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p}_{\mathrm{w}} \end{Bmatrix}_n + \Delta t_n \begin{Bmatrix} \mathbf{f}_{\mathrm{u}} \\ \mathbf{f}_{\mathrm{w}} \end{Bmatrix}_n \tag{13}$$

and $\mathbf{X}^T = \{\mathbf{u}, \mathbf{p}_{\mathrm{w}}\}$. It has to be reminded that this system of equations has to be solved iteratively together with Equations (4) and (6) or (5) and (7) if the (cumulative) gas production is given as input.

3. SUBSTRUCTURING TECHNIQUE

Substructuring techniques have been efficiently used in the finite element analysis [16]. They may be described as 'divide-and-conquer' algorithms since their objective is to divide the larger problem into a series of smaller subproblems. The basic formulation of substructuring method is briefly recalled here.

Consider a domain, Ω, that is subdivided into m subregions (substructures) without overlapping boundaries $\Omega = \cup_{j=1}^{m} \Omega_j$.

For each substructure, Equation (11) takes the following matrix form

$$\begin{bmatrix} \mathbf{kk}^i_{mm}((\mathbf{x}_m)_{n+1}) & \mathbf{kk}_{mi}((\mathbf{x}_i)_{n+1}) \\ \mathbf{kk}_{im}((\mathbf{x}_m)_{n+1}) & \mathbf{kk}_{ii}((\mathbf{x}_i)_{n+1}) \end{bmatrix} \begin{Bmatrix} (\mathbf{x}_m)_{n+1} \\ (\mathbf{x}_i)_{n+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{ff}^i(\mathbf{x}_m)_n \\ \mathbf{ff}(\mathbf{x}_i)_n \end{Bmatrix}, \tag{14}$$

where $\mathbf{kk}^i_{mm}$ indicates the part of $\mathbf{kk}_{mm}$ referring to the interface nodes belonging to substructure i. They can be obtained by assembling the element matrix equations.

The elimination of the internal nodes can be carried out in every substructure. Application of static condensation to Equation (14) results in the following:

$$\begin{bmatrix} \mathbf{I} & \mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{im}((\mathbf{x}_m)_{n+1}) \\ 0 & \mathbf{kk}^i_{mm}((\mathbf{x}_m)_{n+1}) - \mathbf{kk}_{mi}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{im}((\mathbf{x}_m)_{n+1})_{im} \end{bmatrix} \begin{Bmatrix} (\mathbf{x}_i)_{n+1} \\ (\mathbf{x}_m)_{n+1} \end{Bmatrix}$$
$$= \begin{Bmatrix} \mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{ff}_i((\mathbf{x}_i)_n) \\ \mathbf{ff}^i_m((\mathbf{x}_m)_n) - \mathbf{kk}_{mi}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{ff}_i((\mathbf{x}_i)_n) \end{Bmatrix}. \tag{15}$$

The global interface equations may be assembled by using the coefficients corresponding to interface variables in each substructure

$$\left[\sum_{i=1}^{n} \mathbf{kk}^i_{mm}((\mathbf{x}_m)_{n+1}) - \sum_{i=1}^{n} \mathbf{kk}_{mi}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{im}((\mathbf{x}_m)_{n+1})\right]\{(\mathbf{x}_m)_{n+1}\}$$
$$= \sum_{i=1}^{n} \mathbf{ff}^i_m((\mathbf{x}_m)_n) - \sum_{i=1}^{n} \mathbf{kk}_{mi}((\mathbf{x}_i)_{n+1})\mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})\mathbf{ff}_i((\mathbf{x}_i)_n). \tag{16}$$

The interface values $(\mathbf{x}_m)_{n+1}$ are obtained by solving Equation (16). These are then introduced into the first of Equations (15) to obtain the values $(\mathbf{x}_i)_{n+1}$ within each substructure

$$((\mathbf{x}_i)_{n+1}) = \mathbf{kk}_{ii}^{-1}((\mathbf{x}_i)_{n+1})[\mathbf{ff}_i(\mathbf{x}_i)_n - \mathbf{kk}_{im}((\mathbf{x}_m)_{n+1})((\mathbf{x}_m)_{n+1})]. \tag{17}$$

Again, Equations (4) and (6) yield the boundary conditions at the beginning of each time step.

4. MULTI-FRONTAL ALGORITHM

Solving Equations (15)–(17) directly consumes huge computing times. A multi-frontal method can efficiently carry out the above mentioned static condensation in each substructure. It is not necessary to solve directly the inverse of the matrix $\mathbf{kk}_{ii}$ and form directly the coupling matrices $\mathbf{kk}_{im}$ and $\mathbf{kk}_{mi}$ associated with internal and interface nodes.

First of all, a very brief sketch of the operations performed in the program may be helpful.

The frontal solution method of Irons [17] recognizes the basic property of the finite element method that a typical node *ik* contributes a non-zero entry *jk* in the matrix **KK** only when the node *jk* belongs to one of the elements that contains the node *ik*. Irons has made full use of this characteristic feature by controlling the order of elimination according to the sequence as against the node numbering in band schemes. The wavefront never exceeds the bandwidth and is usually smaller. The frontal process alternates between accumulation of element coefficients (assembly) and elimination. Only the coefficients of matrix **KK** associated with the 'active' variables need to be readily available in the core. A variable is said to become 'active' on its first appearance, and is eliminated immediately after its 'last' appearance.

Consider for this purpose, the two-dimensional four-element ten-noded mesh shown in Figure 1. The solving region is subdivided into two substructures. As an example, it is supposed that each node has only one degree of freedom, and all the equations are stored in the order of ascending node number. A nodal number array is constructed by tracing elimination of degrees-of-freedom of nodes. All degree-of-freedoms of the interface nodes are set with positive signs, and degree-of-freedoms corresponding to fully summed nodes are always set negative. The matrices **A** and **B** are defined as the operating matrices of frontal routine in which the element matrices are to enter. For the left substructure, after assembling of element 1, the states of the matrices **A** and **B** are as follows:

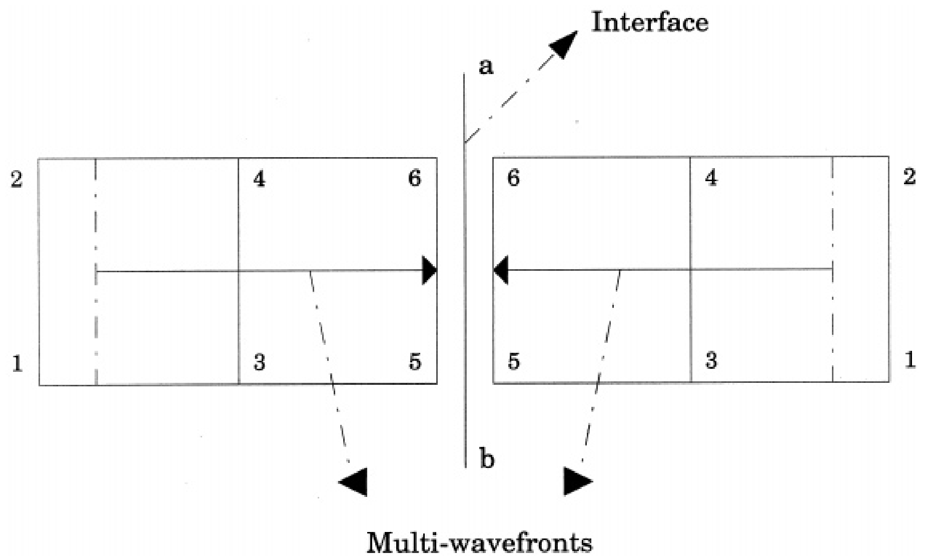

Figure 1. Multi-frontal processes.

$$\mathbf{A}^0 = \begin{bmatrix} a_{41}^1 & a_{42}^1 & a_{43}^1 & a_{44}^1 \\ a_{31}^1 & a_{32}^1 & a_{33}^1 & a_{34}^1 \\ a_{21}^1 & a_{22}^1 & a_{23}^1 & a_{24}^1 \\ a_{11}^1 & a_{12}^1 & a_{13}^1 & a_{14}^1 \end{bmatrix}, \qquad \mathbf{B}^0 = \begin{Bmatrix} b_4^1 \\ b_3^1 \\ b_2^1 \\ b_1^1 \end{Bmatrix}, \tag{18}$$

where a_{ij} and b_i are the coefficients and the right-hand-sides of the element equations, and superscripts 0 and 1 denote operating stage and the element number respectively.

At stage 0, because the contributions to variables x_1 and x_2 are completed (their rows and columns are fully summed), x_1 and x_2 may be eliminated. Eliminating x_1 yields

$$\mathbf{A}^1 = \begin{bmatrix} (a_{42}^1)^1 & (a_{43}^1)^1 & (a_{44}^1)^1 \\ (a_{32}^1)^1 & (a_{33}^1)^1 & (a_{34}^1)^1 \\ (a_{22}^1)^1 & (a_{23}^1)^1 & (a_{24}^1)^1 \end{bmatrix}, \qquad \mathbf{B}^1 = \begin{Bmatrix} (b_4^1)^1 \\ (b_3^1)^1 \\ (b_2^1)^1 \end{Bmatrix}, \tag{19}$$

where

$$(a_{ij}^1)^1 = a_{ij}^1 - \frac{a_{i1}^1 a_{1i}^1}{a_{11}^1}, \qquad (b_i)^1 = b_i^1 - \frac{a_{i1}^1 b_1^1}{a_{11}^1} \quad (i, j = 2, 3, 4). \tag{20}$$

Similarly, x_2 is eliminated

$$\mathbf{A}^2 = \begin{bmatrix} (a_{43}^1)^2 & (a_{44}^1)^2 \\ (a_{33}^1)^2 & (a_{34}^1)^2 \end{bmatrix}, \qquad \mathbf{B}^2 = \begin{Bmatrix} (b_4^1)^2 \\ (b_3^1)^2 \end{Bmatrix}, \tag{21}$$

where

$$(a_{ij}^1)^2 = (a_{ij}^1)^1 - \frac{(a_{i2}^1)^1 (a_{2i}^1)^1}{(a_{22}^1)}, \qquad (b_i^1)^2 = (b_i^1)^1 - \frac{(a_{i2}^1)(b_2^1)^1}{(a_{22}^1)^1}. \tag{22}$$

In the above elimination process, the pivotal rows are normalized by division with pivots and removed into another array for back-substitution. In this stage, x_3 and x_4 are active variables.

Suppose now that the equations from element 2 are assembled, then the operating matrices **A** and **B** become

$$\mathbf{A}^3 = \begin{bmatrix} a_{63}^2 & a_{64}^2 & a_{65}^2 & a_{66}^2 \\ a_{53}^2 & a_{54}^2 & a_{55}^2 & a_{56}^2 \\ (a_{43}^1)^2 + a_{43}^2 & (a_{44}^1)^2 + a_{44}^2 & a_{45}^2 & a_{46}^2 \\ (a_{33}^1)^2 + a_{33}^2 & (a_{34}^1)^2 + a_{34}^2 & a_{35}^2 & a_{36}^2 \end{bmatrix}, \qquad \mathbf{B}^3 = \begin{Bmatrix} b_6^2 \\ b_5^2 \\ (b_4^1)^2 + b_4^2 \\ (b_3^1)^2 + b_3^1 \end{Bmatrix}. \tag{23}$$

After assembling element 2, x_3 and x_4 may be eliminated from the operating matrices, while x_5 and x_6 become active variables. Because of nodes 5 and 6 are the interface nodes and their degrees-of-freedom have been set positive, the frontal routine stops. The static condensation of Equation (15) has been accomplished. The contributions to the coefficient matrix and right-hand-side vector of the interface equation (16) can be directly obtained from the arrays **A** and **B**. A similar procedure can be used in the right substructure. Finally, the interface equation (16) can be formed by summing the contributions of both substructures.

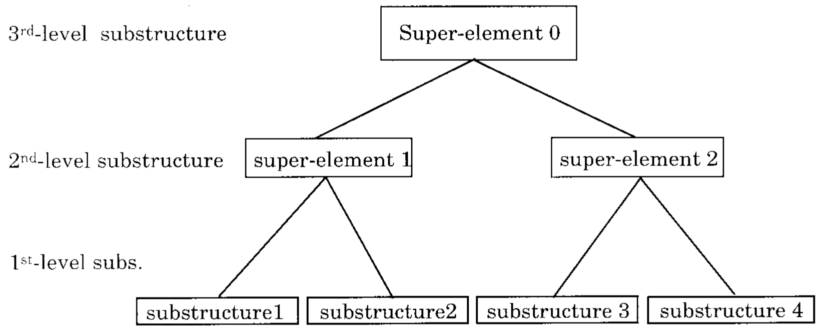

Figure 2. Organization of a multi-level structure.

5. MULTI-LEVEL FRONTAL SCHEME

A multi-level frontal scheme can be used for multi-level substructural analysis [18,19]. The given domain of the problem is discretized into multi-level substructures. A substructure at higher-level, called super-substructure, can be composed of the substructures that are at the lower level. Multi-frontal procedures are first used to assemble and eliminate at the lowest level substructures. The contributions to the equations of higher level substructures are obtained once every wavefront comes to the boundary of its own substructure. Then multi-frontal procedures go up to the substructures at a higher level, and so on up to the equations of the highest level substructure, i.e. global interface equations. Finally, the interface equations are solved to obtain the values of variables of the highest level structure. Once the equations corresponding to the super-element at the highest level have been solved, the solutions of other substructures may be obtained through back-substitution routines level by level.

In order to show how to construct the multi-level structure, the example of Figure 1 will again be used. However, every element of this example is here treated as a substructure that may contain many elements; the entire region is hence divided into four substructures. Super-elements 1 and 2 can be composed of two substructures on both sides of line *ab* respectively. Finally, super-elements 1 and 2 compose super-element 0 at the highest level. Figure 2 shows the constructing method of such a tree-level substructure.

The solution of the example described above by using a multi-frontal scheme can be reached through the multi-frontal procedures of Section 3 to do static condensations in the first level substructures, and form the equations of super-elements 1 and 2 respectively. Then multi-frontal procedures go up to the second level, finish static condensations of each super-element and assemble super-element 0. When the values of the nodes corresponding to the super-element 0 are obtained, the solutions for every substructure level can be obtained by using a back-substitution routine of multi-frontal procedure level by level.

6. PARALLEL COMPUTING ORGANIZATION

Parallel computing for the problem described in the previous sections is implemented on a cluster of workstations through the parallel virtual machine (PVM) system [9,10].

The PVM system is composed of a set of user-interface primitives and supporting software that together enable concurrent computing on loosely coupled networks of processing elements. PVM may be implemented on a hardware base consisting of a mixture of machine architectures, including single CPU systems, vector machines and multi-processors. These computing elements may be interconnected by one or more networks, which may themselves be different. Applications access these processing elements via a standard interface, namely a set of well-defined primitives that are embedded in procedural host languages. They are composed of one or more components that are subtasks at moderately large level of granularity. During execution, multiple instances of each component may be initiated.

The PVM user interface is strongly typed; a support for operating in a heterogeneous environment is provided in the form of special constructs that selectively perform machine-dependent data conversions where necessary. Inter-instance communication constructs include those for the exchange of data structures as well as high-level primitives, such as broadcast, barrier synchronization, mutual exclusion, global extreme and rendezvous.

Application programs view the PVM system as a general and flexible parallel computing resource that supports a message passing model of computation. PVM supports three general parallel programming models: tree computations, crowd computations and hybrid computation, in which crowd computing is the most common model for PVM applications. In this paper, the master–slave (or host–node) model in crowd computing is adopted. The master program is responsible for process spawning, initialization, simple computation, collection and display of results. The slave programs perform the actual computation involved.

Programming a PVM-like workstation network is, for the most parts, similar to programming a sequential computer, except that special attention has to be paid to the operations requiring communication among the processors. Most computations in the proposed algorithm can be performed at the substructure or super-element level. Thus, a natural parallel implementation of the method comprises a mapping of one (or more) substructure(s) (or super-element) onto each processor of workstation networks. The multi-level frontal parallel method is described in Figure 3. In general, the number of the substructures and super-elements will reduce, with increasing level, in the multi-level substructures [20]. Thus, it is better to balance the work of each processor on the basis of the number of substructures and super-elements.

7. APPLICATION

As an example, the subsidence in a small portion of the Northern Adriatic Basin is investigated, where four gas reservoirs are sited at different depths in the subsoil and are exploited with different production histories. The situation requires a three-dimensional analysis. In the Northern Adriatic Basin, several gas fields exist that are actually exploited or are planned to be exploited. Hence, material properties have been extensively investigated and are available. The studied region covers an area of 40×40 km^2 and has a depth of 1300 m; it is discretized by $7 \times 7 \times 12$ 20-noded isoparametric elements (with a total amount of 588 elements and 3056 nodes). Free flux on the horizontal and vertical boundaries of the investigated area is assumed. The stratigraphy and some material parameters are shown in Figure 4 and Table I [7,21], k_i being the permeabilities along direction i, α the compressibility of the grains and γ the specific unit weight of the solid ($\gamma = 0$ means that subsidence problems for natural consolidation of the soil are neglected in the computations). Some data have been obtained through analysis of *master-logs* at our disposal, which are representative of the investigated area. The location of the pools is shown in Figure 5 (horizontal projection); in particular, the division into substructures is indicated: the two-level substructure analysis is characterised by two substructures, each one with 1528 elements, whereas the three-level analysis by four substructures with 147 elements.

```
Master program:

Step 1.  Enrol the program in PVM
Step 2.  Set number of slaves to spawn, initialize the instances of slave program,
         broadcast subregion data to all slave programs
Step 3   Time-step cycle
Step 4.  Multi-level frontal procedure
         (1) Receive the condensed interface matrices and load vectors of super-
         elements from (n-1)-level substructure
         (2) Assemble and solve global interface equations (17) to get $X_m$
         (3) Send $X_m$ to every super-element at (n-1)-level structure
         (4) Receive $X_i$ from super-elements or structure
Step 5.  Check time-step and send time-step information to slave programs, if
         time-step was not finished then go to step 3 else go to step 6
Step 6.  Exit PVM and stop

Slave program:

Step 1.  Enrol the program in PVM
Step 2.  Receive super-elements and substructure data from master program
Step 3.  Muti-level frontal procedure
         (1) Static condensation: assemble and eliminate in super-elements or
         substructures to get the condensed interface matrices and load vector of
         the higher level super-element
         (2) Send condensed interface matrix and load vector to the slave
         program (or master program) corresponding to the higher level super-
         element
         (3) Back-substitution: from master program (or slave program)
         corresponding to the higher level super-element
         (4) Send $X_i$ to slave program corresponding to the lower super-element
         (or substructure)
Step 4.  Receive time-step information, if time-step is not finished then go to
         step 3
         else go to step 5
Step 5.  Exit PVM and stop
```

Figure 3. Steps used by a cluster of workstations.

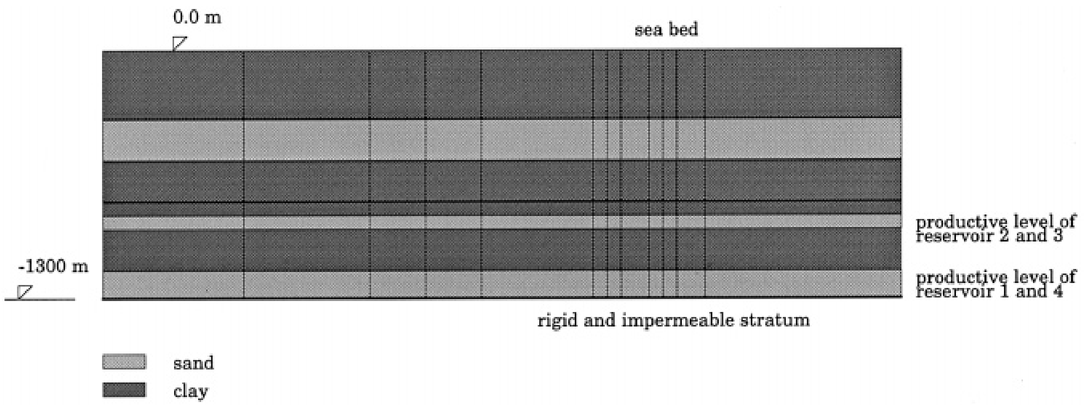
0.0 m
sea bed
-1300 m
productive level of reservoir 2 and 3
productive level of reservoir 1 and 4
rigid and impermeable stratum
sand
clay

Figure 4. The stratigraphy scheme (vertical section A–A).

Table I. Material properties

Layer	**E** (MPa)	ν	$\mathbf{k}_x$ (m day^{-1})	$\mathbf{k}_y = \mathbf{k}_z$ (m day^{-1})	α	γ
1	224	0.39	0.865E−04	0.865E−04	0.10E−12	0.0
2	898	0.15	0.9752	0.9752	0.10E−12	0.0
3	555	0.37	0.865E−04	0.865E−04	0.10E−12	0.0
4	322	0.38	0.865E−04	0.865E−04	0.10E−12	0.0
5	1140	0.17	0.7985	0.7985	0.10E−12	0.0
6	1000	0.37	0.865E−04	0.865E−04	0.10E−12	0.0
7	1130	0.17	0.2208	0.2208	0.10E−12	0.0

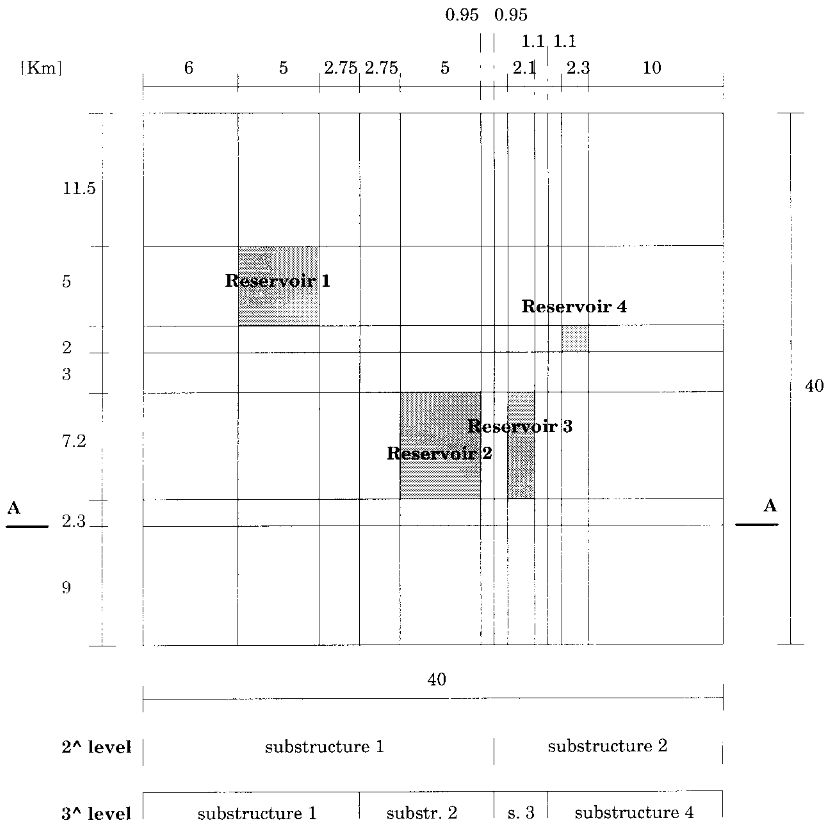

Figure 5. Location of the reservoirs (horizontal projection) and division into substructures.

The exploitation points (wells) are assumed to be equally distributed above each reservoir so as to allow for the assumption of a constant drop of pressure inside it. The pressure histories, shown in Figure 6, obtained from previous reservoir simulators, are applied as boundary conditions to the nodes of each pool. A computationally more expensive alternative would be to apply the outflow given from the production schedule (if available), as explained in Section 2. The results for a three-level substructure analysis in terms of surface subsidence above each reservoir are shown in Figure 7; the effect of interaction among the different reservoirs can be seen from the shifting in time of the maximum value of subsidence as compared with the minimum of reservoir pressure. This phenomenon is also to be partly ascribed to the presence of thick clay layers confining the pools. Figure 8 shows the surface settlement at the time of maximum subsidence, along an ideal broken line passing above the central point of each reservoir.

Table II gives the computing time, speed-up and efficiency of the considered example, showing that problems like this can be run efficiently on a cluster of workstations. The speed-up is measured by speed-up $= T_s/T_p$, where T_s is the solution time of one processor (CPU) and T_p is the parallel computation time using p processors (CPUs) of the workstations. The efficiency is measured by speed-up$/p$, where p is the above mentioned number of processors.

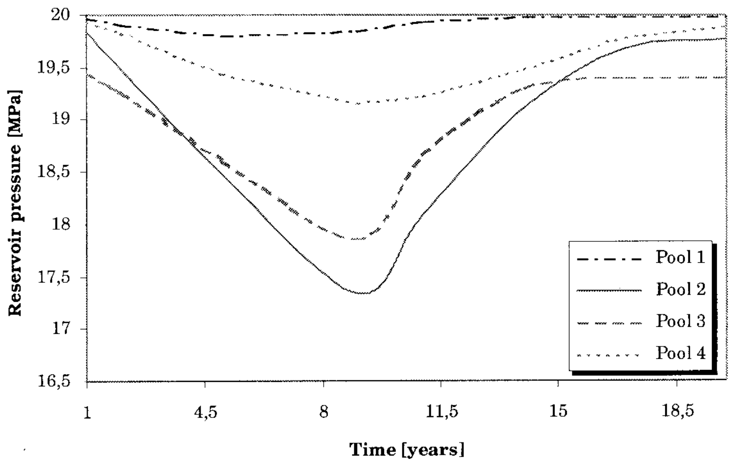

Figure 6. History of the reservoirs pressures.

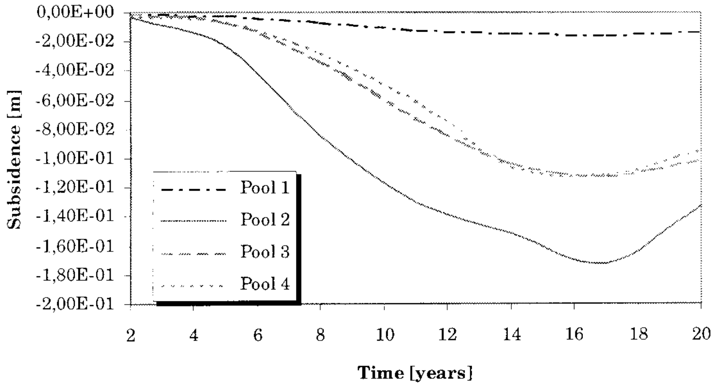

Figure 7. History of surface subsidence above the reservoirs.

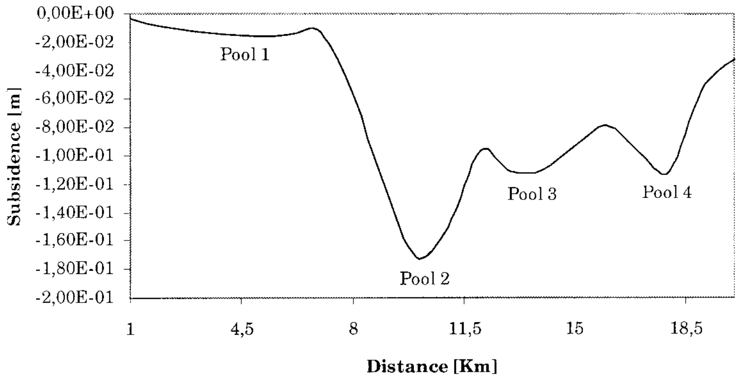

Figure 8. Land subsidence.

Table II. Speedup and efficiency

Number of processors	CPU time (s)	Speed-up	Efficiency (%)
1	47625.542	—	—
2 (2-level substructure)	24640.012	1.93	0.965
4 (3-level substructure)	16535.65	2.88	0.72

8. CONCLUSIONS

In this paper a multi-level frontal parallel algorithm has been presented for a three-dimensional analysis of subsidence above gas reservoirs. The main operations of the method are performed by means of multi-level frontal processes at a super-element level in parallel. Multi-frontal processes in the multi-level substructures can increase utilization ratio of parallel processors. The given domain is partitioned into multi-level substructures according to the following rules: (1) the global interface problem should be minimized, i.e. the super-element in the highest level must consist of minimum nodes connectivity to reduce the magnitude of the global interface equations and communication overheads in parallel computation; (2) lower super-elements should have roughly an equal amount of computational load to ensure that all the processors finish their work at about the same time.

The proposed parallel implementation of the method allocates one super-element to each processor of a cluster of workstations and handles the communications among the networked workstations by using the popular parallel software package PVM. The application shows that the multi-level frontal parallel algorithm for three-dimensional subsidence analysis results in considerable savings in computer time. Workstations have communication speed slower than usual parallel computers; this may more or less reduce parallel computing efficiency. However, they are still capable of performing parallel three-dimensional subsidence analyses above gas reservoirs effectively. Although this algorithm is developed for workstations, it is also suitable for other parallel computers.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support for this work from the International Civil Engineering Services (ICES) S.p.A. of Padova, Italy, and the National Natural Science Foundation of China. The computations were performed on a cluster of workstations at the Department of Structural and Transportation Engineering of the University of Padova.

APPENDIX A. NOMENCLATURE

$\mathbf{B}$	strain matrix relating strain and displacement
$\mathbf{D}_T$	tangent matrix
$\mathbf{g}$	acceleration of gravity (m s^{-2})
$\mathbf{k}$	absolute permeability matrix
K_w	bulk modulus of liquid phase
K_s	bulk modulus of solid phase
$\mathbf{m}$	vector with unit values for the normal stress components and zero for the shear stress components
p_w	liquid water pressure
t	time (s)
$\boldsymbol{t}_f$	traction imposed on boundary
$\mathbf{u}$	displacement vector of solid matrix (m)
$\mathbf{b}$	body force vector
Greek letters	
α	Biot's constant
ε	strain vector
ε_0	all other strains not directly associated with stress change
φ	porosity (pore volume/total volume)
ρ	effective density of porous medium (kg m^{-3})
ρ_w	liquid phase density (kg m^{-3})
ρ_s	solid phase density (kg m^{-3})
σ	total stress vector
Δt	time step (s)

APPENDIX B

The matrices contained in the discretized governing Equation (8) are

$$\mathbf{c}_{uu} = -\int_\Omega \mathbf{B}^T\mathbf{D}_T\mathbf{B}\,\mathrm{d}\Omega,$$

$$\mathbf{c}_{uw} = \int_\Omega \left(\mathbf{B}^T\mathbf{m}\mathbf{N} - \frac{1}{3K_s}\mathbf{B}^T\mathbf{D}_T\mathbf{m}\mathbf{N}\right)\mathrm{d}\Omega,$$

$$\dot{\mathbf{f}}_u = \int_\Omega \left(\mathbf{N}^T\frac{\mathrm{d}\mathbf{b}}{\mathrm{d}t} + \mathbf{B}^T\mathbf{D}_T\frac{\mathrm{d}\varepsilon_0}{\mathrm{d}t}\right)\mathrm{d}\Omega + \int_\Gamma \mathbf{N}^T\frac{\mathrm{d}\mathbf{t}_f}{\mathrm{d}t}\,\mathrm{d}\Gamma,$$

$$\mathbf{c}_{\mathrm{wu}} = \int_\Omega \left(\mathbf{N}^T \mathbf{m}^T \mathbf{B} - \frac{1}{3K_{\mathrm{s}}} \mathbf{N}^T \mathbf{D}_T \mathbf{m}^T \mathbf{B} \right) \mathrm{d}\Omega,$$

$$\mathbf{c}_{\mathrm{ww}} = \int_\Omega \mathbf{N}^T \left[\frac{1-\phi}{K_{\mathrm{s}}} + \frac{\phi}{K_{\mathrm{w}}} - \frac{1}{(3K_{\mathrm{s}})^2} \mathbf{m}^T \mathbf{D}_{\mathrm{T}} \mathbf{m} \right] \mathbf{N}\, \mathrm{d}\Omega,$$

$$\mathbf{k}_{\mathrm{ww}} = \int_\Omega (\nabla \mathbf{N})^T \frac{\mathbf{k}}{\mu} \nabla \mathbf{N}\, \mathrm{d}\Omega,$$

$$\mathbf{f}_{\mathrm{w}} = \int_\Omega (\nabla \mathbf{N})^T \frac{\mathbf{k}}{\mu} \nabla \rho g h\, \mathrm{d}\Omega + \int_\Gamma \mathbf{N}^T q\, \mathrm{d}\Gamma.$$

REFERENCES

1. M. Zambon, 'Abbassamenti anormali del suolo nel Ravennate', *La Bonifica*, **1–2**, 58–64 (1968).
2. L. Carbognin, P. Gatto, G. Mozzi and G. Gambolati, 'Land subsidence of Ravenna and its similarities with the Venice case', in S.K. Saxena (ed.), *Evaluation and Prediction of Subsidence*, ASCE, New York, 1978, pp. 254–266.
3. Ministero Dell' Agricoltura e Delle Foreste, 'l'Abbassamento del suolo della zone litorale Ravennate', *Concessione Studio D. M. 9-11-71*, No. 481, 1976.
4. T. Ippolito, 'La subsidenza di Ravenna', *Le Sci.*, **99**, 104–111 (1976).
5. B.A. Schrefler, R.W. Lewis and C.E. Majorana, 'Subsidence above volumetric and waterdrive gas reservoirs', *Int. J. Numer. Methods Fluids*, **1**, 101–115 (1981).
6. R.W. Lewis and Y. Sukirman, 'A coupled implicit model for deforming porous media', in *Proc. 2nd APCOM, Sydney, Australia*, S. Valliappan, V.A. Pulmano, V. Murti (eds.), Balkema, Rotterdam, 1993, pp. 573–578.
7. R.W. Lewis and B.A. Schrefler, *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*, Wiley, Chichester, 1998.
8. H. Adeli (ed.), 'Parallel and distributed processing in structural engineering', in *Proceedings of a Session Sponsored by the Structural Division of ASCE in Conjunction with the ASCE National Convention*, Nashville, TN, 11 May, 1988.
9. V. Sunderam, 'PVM: a framework for parallel distributed computing', *Concurr. Pract. Exp.*, **3**, 315–319 (1990).
10. G.A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM 3 User's Guide and Reference Manual*, Oak Ridge National Laboratory, ORNL/TM-12187, 1994.
11. B.H.V. Topping and A.I. Khan, 'Parallel computations for structural analysis, re-analysis and optimization', Presented at the NATO: DFG Advanced Study Institute, Optimization of Large Structural Systems, Berchtesgaden, Germany, 23 September–4 October, 1991.
12. P. Geng, J.T. Oden and R. A. van de Geijn, 'A parallel multi-frontal algorithm and its implementation', *Comput. Methods Appl. Mech. Eng.*, **149**, 289–301 (1997).
13. L. Simoni, V. Salomoni and B.A. Schrefler, 'Elastoplastic subsidence models with and without capillary effects', *Comput. Methods Appl. Mech. Eng.*, **171**, 491–502 (1999).
14. B.C. Craft and M.F. Hawkins, *Applied Petroleum Reservoir Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1969.
15. B.A. Schrefler, 'FE in environmental engineering: coupled thermo-hydromechanical processes in porous media including pollutant transport', *Arch. Comput. Methods Eng.*, **2**, 1–54 (1995).
16. J.S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw-Hill, New York, 1986.
17. B.M. Irons, 'A frontal solution program for finite element analysis', *Int. J. Numer. Methods Eng.*, **2**, 5–32 (1970).
18. P. Hood, 'Frontal solution program for unsymmetric matrices', *Int. J. Numer. Methods Eng.*, **10**, 379–399 (1976).
19. M.F. Yeo, 'A more efficient front solution: allocating assembly locations by longevity consideration', *Int. J. Numer. Methods Eng.*, **7**, 570–573 (1973).
20. X. Wang, D. Gawin and B.A. Schrefler, 'A parallel algorithm for thermo-hydromechanical analysis of deforming porous media', *Comput. Mech.*, **19**, 94–104 (1996).
21. AGIP, 'Progetto Alto Adriatico—Studio di Impatto Ambientale', 1996.