Mathematica Balkanica

Implementation of the additive overlapping domain decomposition method to 3-D elasticity problems [1]

Krassimir Georgiev

Presented by S. Margenov

We consider the additive overlapping domain decomposition method for 3-D linear elasticity problems. Due to the possibilities to model complicated structures the finite element method (FEM) is an extremely successful method for the problems of linear elasticity. The implementation requires the solution of very large, sparse, positive definite linear systems of equations.

In this paper we report on some algorithms and numerical experiments concerning the implementation of the additive overlapping domain decomposition method to problems arising from the 3-D Elasticity Theory.

1. Introduction

The domain decomposition (DD) methods are very convenient in order to handle complex geometries, nonsmooth solutions, differential equations which have different coefficients in different subregions of the domain.

In 1936, Sobolev [9] showed that the Schwarz alternating algorithm converges for the linear elasticity equations. Recent work in this area were done by Bjorstad and Hvidsten [2] (Neumann - Dirichlet algorithm), De Roeck [8] (iterative substructuring type algorithm), Huges and Ferencz [6], and Tezduyar and Lion [10] (element - by - element preconditioning on large structural problems), etc.

In this paper we report on some algorithms and numerical experiments concerning the implementation of the *additive overlapping domain decomposition* (DD) method to problems arising from the 3-D Elasticity Theory.

The paper is organized as follows. Section 2. is devoted to a description of the problem under consideration. In Section 3. we focus on the discretization and the solution method where three-linear finite elements and preconditioned conjugate gradient method with an additive overlapping domain decomposition preconditioner are included. A brief description of the benchmark problem according to the research plan of the HIPERGEOS Project COP–94 00820 is given in Section 4. Section 5. is the main section in this paper. It is devoted to some numerical experiments which have been done using the benchmark problem and a different choice of the discrete problem parameters: *meshsize of the discretization, number of the subdomains used in the domain decomposition algorithm, size of the overlap and stoping criteria.* The results obtained show a good behaviour of the numerical procedure in consideration - domain decomposition preconditioner in preconditioned conjugate gradient method and *block–size reduction BILU* algorithm for solving the local (in each subdomain) problems. Small numbers of overlapping subdomains are used at this stage of the investigations and therefore a coarse grid correction has not been constructed. The results are compared with the previous results obtained for the same benchmark problem.

2. The elasticity problem

Let B be an elastic body occupying bounded polyhedral domain $\Omega \subset \mathbf{R}^3$, impose a Dirichlet and Neumann boundary conditions on $\Gamma = \partial\Omega$ (fig.1). The domain consists of two different parts: *soil* – Ω_1 and *concrete element* – Ω_2. We denote by $\mathbf{u} = [u_1, u_2, u_3]^T$ the vector of the displacements, by $\underline{\sigma} = (\sigma_{ij})(\mathbf{u})$ the stress tensor and by $\underline{\epsilon}(\mathbf{u}) = (\epsilon_{ij}(\mathbf{u}))$ the strain tensor (the deformation associated with $\mathbf{u}$).

We consider elasticity problems under the following general assumptions:

- the displacements are small;

- the material properties are isotropic.

The material constants depend on the modulus of elasticity (Young's modulus) - E and on the contraction ratio (Poisson ratio) - ν of the elastic materials of the body B.
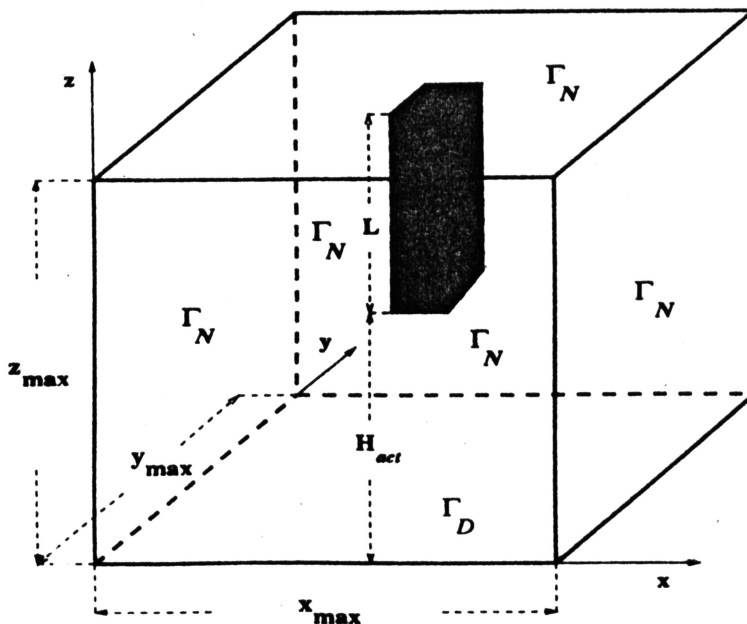
Figure 1. *The domain under consideration*

Let us note with D the elasticity matrix

$$D = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix},$$

where the positive λ and μ (material constants) can be expressed as:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{1+\nu}.$$

Thus,

$$\underline{\sigma} = D\underline{\varepsilon} \quad \text{or} \quad \underline{\varepsilon} = D^{-1}\underline{\sigma}.$$

If we introduce the matrix A as:

$$A = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{pmatrix},$$

then $\epsilon = A\mathbf{u}$.

The following constitute relation (Hooke's law)

$$\sigma_{ij} = \lambda\, div\mathbf{u}\, \delta_{ij} + \mu\, \epsilon_{ij}(\mathbf{u}), \quad i, j = 1, 2, 3.$$

and the equilibrium equations:

$$(L\,\mathbf{u})_i \equiv -\sum_{j=1}^{3} \frac{\partial \sigma_{ij}(\mathbf{u})}{\partial x_j} = f_i, \quad i = 1, 2, 3$$

hold, where $\mathbf{f} = [f_1, f_2, f_3]^T$ is the vector of body forces.

3. The numerical problem and the solution method

The numerical problem

The stressed–strained state of the elastic body under consideration (see Section 1) can be described by a coupled elliptic system of three partial differential equations and the corresponding boundary conditions as follows [7]:

$$\begin{aligned} A^T \underline{\sigma} + \mathbf{f} &= 0 & \forall\, (x, y, z) \in \Omega \\ \mathbf{u} &= \mathbf{u}_D & \forall\, (x, y, z) \in \Gamma_D \\ \sum_{i=1}^{3} \sigma_{ij} n_i &= u_{N_j} & \forall\, (x, y, z) \in \Gamma_N, \quad j = 1, 2, 3, \end{aligned}$$

where Γ_D and Γ_N are the parts of the boundary Γ with Dirichlet and Neumann boundary conditions and $\mathbf{u}_D$ and $\mathbf{u}_N$ are the given boundary data.

The variational formulation of the above determined elasticity problem is:

Find: $\mathbf{u} \in (H^1(\Omega))^3$, such that

$$a_\Omega(\mathbf{u}, \mathbf{v}) \equiv \int_\Omega (\lambda\, div\mathbf{u}\, div v + \mu \sum_{i,j=1}^{3} \epsilon_{ij}(\mathbf{u})\, \epsilon_{ij}(\mathbf{v}))\, dx = F(\mathbf{v}), \quad \forall \mathbf{v} \in (H^1(\Omega))^3,$$

where

$$F(\mathbf{v}) = \sum_{i=1}^{3} \left(\int_{\Omega} f_i v_i dx + \int_{\Gamma_N} u_{N_i} v_i d\Gamma \right).$$

The bilinear form $a_\Omega(\mathbf{u}, \mathbf{v})$ is a second order, symmetric and coercive bilinear form.

The solution method

The finite element method with three-linear parallelepipedal finite elements is used for the numerical solution of the problem (see [1], for example).

Let $V^h(\Omega) \subset (H^1(\Omega))^3$ be the spaces of continuous piecewise linear functions, on the triangulation. Then, the discrete variational problem is:

Find: $\mathbf{u}_h \in V^h(\Omega)$, such that

$$a_\Omega(\mathbf{u}_h, \mathbf{v}_h) = F(\mathbf{v}_h), \quad \forall\, \mathbf{v}_h \in V^h(\Omega).$$

We will use the standard nodal basis functions Φ_i for the space V^h. The algorithm leads to the linear system of equations according to the unknown nodal displacements

(0.1) $$\mathbf{U} = \mathbf{F},$$

where $\mathbf{U} = [U_i]^T$ is the vector of unknowns $- i = 1, 2, \ldots, n$, n is the number of all nodes in Ω, the right-hand side is the vector of components $F_i = \mathbf{F}(\Phi_i)$, and K is the stiffness matrix with entries $K_{ij} = a_\Omega(\Phi_i, \Phi_j)$.

To solve the linear system of equations (1) the following preconditioned conjugate gradient procedure is used.

<u>PCGM – procedure:</u>

- $\mathbf{U}_0$ – *an initial arbitrary choice*;

- $\mathbf{r}_0 = \mathbf{F} - K\mathbf{U}_0$;

for $k = 0, 1, \ldots$ *until convergence*

- $M\mathbf{z}_k = \mathbf{r}_k$;

- $\beta_k = \dfrac{(\mathbf{r}_k, \mathbf{z}_k)}{(\mathbf{r}_{k-1}, \mathbf{z}_{k-1})}, \qquad \beta_0 = 0;$

- $\mathbf{p}_k = \mathbf{z}_k + \beta_k \mathbf{p}_{k-1};$

- $$\alpha_k \;=\; \frac{(\mathbf{r}_k, \mathbf{z}_k)}{(A\mathbf{p}_k, \mathbf{p}_k)};$$

- $$\mathbf{U}_{k+1} \;=\; \mathbf{U}_k + \alpha_k \mathbf{p}_k;$$

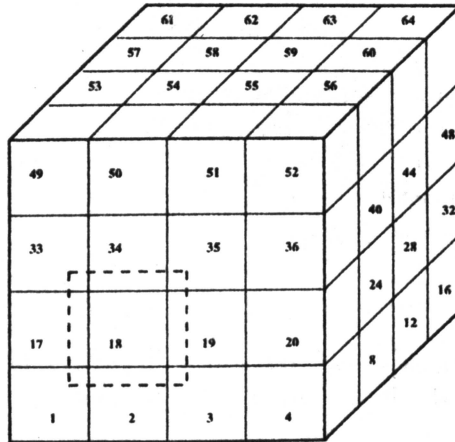- $$\mathbf{r}_{k+1} \;=\; \mathbf{r}_k - \alpha_k A\mathbf{p}_k;$$

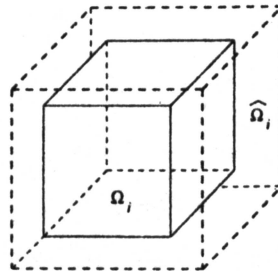Figure 2. *Partioning of the spatial domain* Ω $(4 \times 4 \times 4)$; $N = 64$.

Figure 3. *Subdomain* Ω_i *and extended subdomain* $\widehat{\Omega}_i$

The preconditioner M is constructed via the *additive overlapping domain decomposition method* as follows.

The domain Ω is divided into non-overlapping substructures $\Omega_i, i = 1, 2, ..., N$ (Fig.2) with diameters of the order of H while the finite elements have characteristic diameters of order h.

We extend each subdomain Ω_i to a larger region $\widehat{\Omega}_i$ (Fig.3). We assume that the overlap is generous, assuming that the distance between the boundaries $\partial\Omega_i$ and $\partial\widehat{\Omega}_i$ is bounded from below from a fixed fraction of H_i, the diameter of Ω_i.

We also assume that $\partial\widehat{\Omega}_i$ does **not cut** through any finite element. We make the same construction for the substructures that are next to the boundary of the domain Ω except that we cut off the part of $\widehat{\Omega}_i$ that is outside of Ω.

Let K_i be a local stiffness matrix according to the subdomain $\widehat{\Omega}_i$. We define restriction maps R_i and extension maps R_i^T, as follows. Let the global stiffness matrix K is of order n and denote with $\widehat{n}_i$ the number of the nodes with respect to $\widehat{\Omega}_i$. For each subdomain $\widehat{\Omega}_i$ the matrix R_i is $(n \times \widehat{n}_i)$ restriction matrix with entries 1's and 0's, taking into account the indexes of the nodes belonging to $\widehat{\Omega}_i$. This matrix restricts a vector $\mathbf{x}$ of length n to a vector $R_i\mathbf{x}$ of length $\widehat{n}_i$. Thus, the local stiffness matrix $\widehat{K}_i$ corresponding to the subdomain $\widehat{\Omega}_i$ is

$$\widehat{K}_i = R_i K R_i^T.$$

In the case of many subdomains, the additive Schwarz algorithm is a generalization of the well known two-subdomain case with a preconditioner

$$M_{add}^{-1} = \sum_{i=1}^{N} R_i^T \, \widehat{K}_i^{-1} \, R_i.$$

As the matrix–vector products $R_i^T K_i^{-1} R_i \, \mathbf{y}$ can be computed in parallel for the different subdomains, this leads to a coarse grain parallelism. Taking into account the definition of the restriction matrix R_i (and the extension or interpolation matrix R_i^T) we may conclude that it is not necessary to store these matrices.

So defined preconditioner M_{add} is a straightforward generalization of the standard block-Jacobi preconditioner. The convergence rate of this preconditioner deteriorates when the number of subdomains increases.

The algorithm for solving the system $M\mathbf{z}_k = \mathbf{r}_k$ with $M = M_{add}$ is as follows.

- $\widehat{\mathbf{r}}_k^i$ — vector of a length $\widehat{n}_i$; entries are equal to those of $\mathbf{r}_k$
  for the components corresponding to the meshpoints in
  $\widehat{\Omega}_i$ and 0 elsewhere;

- $\widehat{\mathbf{z}}_k^i$: $\widehat{K}_i \widehat{\mathbf{z}}_k^i = \widehat{\mathbf{r}}_k^i$;

- $\mathbf{z}_k^i$ — vector of a length n; an extension of $\widehat{\mathbf{z}}_k^i$
  with 0 entries outside $\widehat{\Omega}_i$;

- $\mathbf{z}_k = \sum_{i=1}^{N} \mathbf{z}_k^i$

The coefficient matrices of the systems of linear algebraic equations

$$\widehat{K}_i \widehat{\mathbf{z}}_k^i = \widehat{\mathbf{r}}_k^i$$

are symmetric positive definite block matrices and the preconditioned conjugate gradient method with block–size reduction block–ILU preconditioner proposed by Chan and Vassilevski [4] (see also: [3] and [5]) is used to solve the systems in each subdomain.

4. The benchmark problem

The benchmark problem imposes some basic requirements to the HIPER-GEOS code related to the bridge engineering applications. A filling pile element under the assumption that the technology guarantees the ideal contact between the pile and the surrounding soil media is considered.

The computational domain is the parallelepiped

$$\Omega = [0, x_{max}] \times [0, y_{max}] \times [0, z_{max}].$$

Boundary conditions of Dirichlet and Neumann type on the different parts of the boundary of the domain are given as follows:

- Homogeneous Dirichlet boundary conditions on the *bottom side* of the parallelepiped $z = 0$, i.e the displacements are equal to zero;

- Homogeneous Neumann boundary conditions on the *vertical sides* of the parallelepiped .

$$x = 0, \quad x = x_{max}, \quad y = 0, \quad y = y_{max},$$

i.e the stresses are equal to zero;

- Nonhomogenuous Neumann boundary conditions on the *top side* of the parallelepiped $z = z_{max}$, where the load from the upper construction is applied.

Benchmark: Single pile in a homogeneous sandy clay soil layer is considered. The computational domain Ω is determined by

$$x_{max} = y_{max} = 12m; \quad z_{max} = 27m.$$

The length of the pile and the size of the active zone under the top of the pile are respectively $L = 15m$ and $H_{act} = 12m$. The mechanical characteristics of the pile and of the soil layer are:

(a) *pile:* $E = 31500\,MPa; \quad \nu = 0.2;$

(b) *soil:* $E = 10\,MPa; \quad \nu = 0.3.$

An uniformly load distribution on the cross section of the pile is assumed.

5. Numerical experiments

This section is devoted to some numerical experiments which have been done using the benchmark problem under consideration and a different choice of the discrete problem parameters: *meshsize of the discretization, number of the subdomains used in the domain decomposition algorithm, size of the overlap and the stopping criteria.* All the tests were run on SUN SPARC station 20.

The results from the numerical experiments have been done are presented in Tables 1–7. In all the tests $h_x = h_y = 0.6m$, while the meshsize parameter in z–direction is different: $h_z = 1m$ in Tables 1–5, and $h_z = 0.5m$ in Tables 6–7.

In Table 1. a comperisson of the *user time* between two subdomains and four subdomains domain decomposition of Ω (see Fig. 4 and Fig.5) is shown. All the times in all the tests are in minutes.

overlap	2 subdomains	4 subdomains
h_z	59.57	76.34
$2h_z$	57.28	85.12

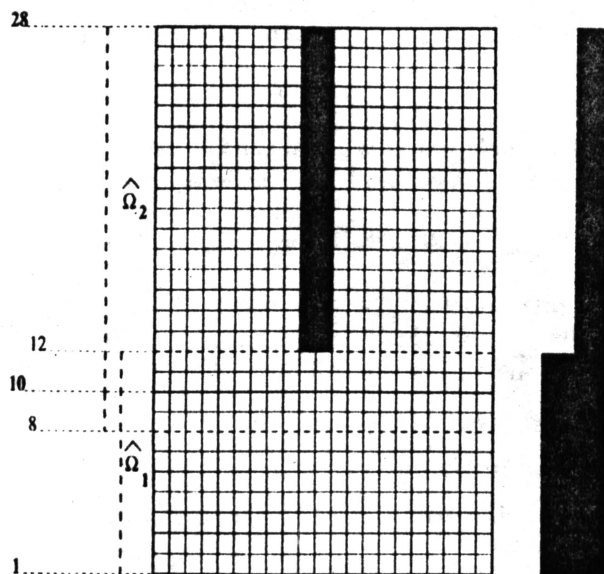Table 1. *User time in minutes; $h_x = h_y = 0.6m$; $h_z = 1m$*

Figure 4. *The two subdomain case. The overlap is 2h.*

Table 2. and Table 3. are devoted to show some dependences of the most important components of the numerical algorithm on the size of the overlapping when the meshsize parameters are: $h_x = h_y = 0.6m; h_z = 1m$.

The user time, the number of iterations in the PCG algorithm with the corresponding domain decomposition preconditioner, *average number of iterations in the inner solver* (the preconditioned conjugate gradient method with block–size reduction block-ILU preconditioner) and *the average reduction factors* for the two solvers can be found in these tables. The results for two–subdomain case show an improving of the components, including the user time when the size of the overlap increases. The number of DD iterations only slightly decreases with the increasing of the overlap size as has to be expected.

There are two test runs for four–subdomains case: with h and $2h$ overlap. The results obtained show that nevertheless the number of DD iterations slightly decreases the user time increases in 12%.

The Table 6. shows the dependences of the most important components of the numerical algorithm on the size of the overlapping when the meshsize parameters are $h_x = h_y = 0.6m; h_z = 0.5m$. One can find the same

dependences as at the previous discretization.
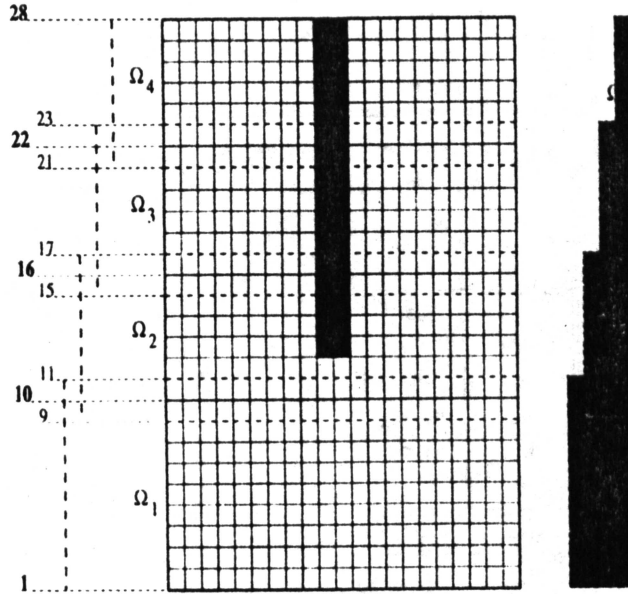
Figure 5. *The four subdomain case. The overlap is h.*

Over-lap	User time in *min*	No. of iter. in pcg-dd	Average No. of iter. in PCG (inner)		*arf* in pcg-dd	*arf* in PCG (inner)	
			Ω_1	Ω_2		Ω_1	Ω_2
h_z	59.57	6	3.2	9.7	0.248	0.46	0.59
$2h_z$	57.28	5	2.8	9.4	0.167	0.32	0.61
$3h_z$	56.46	4	2.7	10	0.108	0.29	0.58
$4h_z$	50.41	4	1.5	5.8	0.096	0.13	0.39

Table 2. *Two subdomain case:* $h_x = h_y = 0.6m$; $h_z = 1m$

Over-lap	User time in min	No. of iter. in pcg-dd	Average No. of iter. in PCG (inner)			
			Ω_1	Ω_2	Ω_3	Ω_4
h_z	76.3	13	3.8	5.1	4.7	7.5
$2h_z$	85.1	10	4.7	6.3	6.9	10

Table 3 a. *Four subdomain case:* $h_x = h_y = 0.6m$; $h_z = 1m$
User time and number of iterations

Over-lap	arf in pcg-dd	arf in PCG (inner)			
		Ω_1	Ω_2	Ω_3	Ω_4
h_z	0.535	0.41	0.44	0.43	0.53
$2h_z$	0.453	046	0.53	0.49	0.65

Table 3 b. *Four subdomain case:* $h_x = h_y = 0.6m$; $h_z = 1m$
Average reduction factor

The results presented up to now are obtained when the stopping criteria tolerances in *the outer* (DD) iterative procedure and *the inner* (PCG in each subdomain) iterative procedure are equal.

Tables 4., 5. and 7. show some results when the tolerance in the inner iterative procedure is square root of the tolerance in DD iterative procedure. Tables 4. and 5. are devoted the cases when $h_x = h_y = 0.6m$; $h_z = 1m$, while the Table 7. presents the results when $h_x = h_y = 0.6m$; $h_z = 0.5m$.

Overlap	$\varepsilon_{in} = \varepsilon_{out}$			$\varepsilon_{in} = \sqrt{\varepsilon_{out}}$		
	User time	No. of iter.	arf	User time	No. of iter.	arf
1	59.57	6	0.248	73.27	41	0.814
2	57.28	5	0.167	43.11	13	0.525
3	56.46	4	0.108	45.21	13	0.523
4	50.41	4	0.096	81.26	39	0.805

Table 4. *Two subdomain case:* $h_x = h_y = 0.6m$; $h_z = 1m$
$\varepsilon_{in} = \varepsilon_{out}$ versus $\varepsilon_{in} = \sqrt{\varepsilon_{out}}$

One can see that in the case of two subdomains and overlap of h_z or $2h_z$ nevertheles the number of DD iterations increases significantly, the user time decreases in 25% and 20% correspondly. At the same time in the case of four subdomains the user time increases in 3% when the overlap is h and decreases in 9% when the overlap is $2h$.

Overlap	$\varepsilon_{in} = \varepsilon_{out}$			$\varepsilon_{in} = \sqrt{\varepsilon_{out}}$		
	User time	No. of iter.	arf	User time	No. of iter.	arf
1	76.34	13	0.535	78.47	34	0.798
2	85.12	10	0.453	77.59	28	0.750

Table 5. *Four subdomain case:* $h_x = h_y = 0.6m$; $h_z = 1m$
$\varepsilon_{in} = \varepsilon_{out}$ versus $\varepsilon_{in} = \sqrt{\varepsilon_{out}}$

Over-lap	User time in in min	No. of iter. in pcg-dd	Average No. of iter. PCG (inner)		arf in pcg-dd	arf in PCG (inner)	
			Ω_1	Ω_2		Ω_1	Ω_2
1	204.42	11	5.36	14	0.489	0.49	0.68
2	169.34	9	5.11	12	0.411	0.47	0.65

Table 6. *Two subdomain case:* $h_x = h_y = 0.6m$; $h_z = 0.5m$

It can be seen in Table 7. that in the case of two subdomains and $h_x = h_y = 0.6m$; $h_z = 0.5m$ the user time decreases in 23% when the overlap is h and in 14% if the overlap is $2h$.

Overlap	$\varepsilon_{in} = \varepsilon_{out}$			$\varepsilon_{in} = \sqrt{\varepsilon_{out}}$		
	User time	No. of iter.	arf	User time	No. of iter.	arf
1	204.42	11	0.489	157.34	26	0.749
2	169.34	9	0.411	145.29	23	0.716

Table 7. *Two subdomain case:* $h_x = h_y = 0.6m$; $h_z = 0.5m$
$\varepsilon_{in} = \varepsilon_{out}$ versus $\varepsilon_{in} = \sqrt{\varepsilon_{out}}$

6. Summary

An implementation of the additive overlapping domain decomposition method to 3-D elasticity problems on sequential computers is presented. The results obtained are for filling pile element under the assumption of ideal contact between the pile and the surrounding soil.

The numerical results of the test examples are in an agreament with the theoretically expected.

REFERENCES

[1] O. A x e l s s o n, V. A. B a r k e r. Finite element solution of boundary value problems: theory and applications, *Orlando: Academic Press*, 1983

[2] P. B j o r s t a d, A. H v i d s t e n. Iterative methods for substructured elasticity problems in structural analysis, *Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. Golub, G. Meurant, J. Périaux, Eds., SIAM, Philadelphia, PA, 1988

[3] T. C h a n, S. M a r g e n o v, P. V a s s i l e v s k i. Performance of Block-ILU factorization preconditioning techniques for linear elasticity problems, *SIAM J. Sci. Comput.*

[4] T. C h a n, P. V a s s i l e v s k i. A framework for block-ILU factorizations using block–size reduction, *Math. Comp.*

[5] T. C h a n, P. V a s s i l e v s k i. Convergence analysis of block–ILU factorization algorithms based on block–size reduction, *UCLA CAM Report, Math. Comp.*

[6] T. H u g h e s, R. F e r e n c z. Fully vectorized EBE preconditioners for nonlinear solid mechanics: applications to large-scale three-dimensional continuum, shell and contact/impact problems, *Proc. of the First Int. Symp. on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. Golub, G. Meurant, J. Périaux, Eds., SIAM, Philadelphia, PA, 1988

[7] J. L. M e e k. Computer methods in structural analysis, Fong @ Sons Printers, 1991

[8] Y. De R o e c k. A local preconditioner in a domain decomposition method, *Tech. Rep., Centre Europée de Recherche et de Formation Avancée en Calcul Scientifique*, Toulouse, France, 1989

[9] S. S o b o l e v. L'Algorithme de Schwarz dans la Théorie de l'Elasticité, *Comptes Rendus (Doklady) de l'Académie des Sciences de l'URSS*, IV, **1212-1220**, 1936

[10] T. T e z d u y a r, J. L i o u. Element-by-element and implicit-explicit finite element formulations for computational fluid dynamics, *Proc. of the First Int. Symposium on Domain Decomposition Methods for Partial Differential Equations*, R. Glowinski, G. Golub, G. Meurant, J. Périaux, Eds., SIAM, Philadelphia, PA, 1988

Central Laboratory for Parallel Processing   Received 07.07.1995
Bulgarian Academy of Sciences
Acad. G. Bonchev str, Bl. 25-A,
1113 Sofia,
BULGARIA