ELSEVIER

Mathematics and Computers in Simulation 61 (2003) 509–516

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

MPI solver for 3D elasticity problems

Ivan Lirkov*

Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Bl. 25A, 1113 Sofia, Bulgaria

Abstract

The numerical solution of 3D linear elasticity equations is considered. The problem is described by a coupled system of second-order elliptic partial differential equations. This system is discretized by trilinear parallelepipedal finite elements.

The preconditioned conjugate gradient iterative method is used for solving of the large-scale linear algebraic systems arising after the finite element method (FEM) discretization of the problem. Displacement decomposition technique is applied at the first step to construct a preconditioner using the decoupled block-diagonal part of the original matrix. Then circulant block-factorization is used for preconditioning of the obtained block-diagonal matrix. Both techniques, displacement decomposition and circulant block-factorization, are highly parallelizable.

A parallel algorithm is invented for the proposed preconditioner. The theoretical analysis of the execution time shows that the algorithm is highly efficient for coarse-grain parallel computer systems.

A portable MPI parallel FEM code is developed. Numerical tests for real-life engineering problems of the geomechanics in geosciences on a number of modern parallel computers are presented. The reported speed-up and parallel efficiency well illustrate the parallel features of the proposed method and its implementation.

MSC: 65F10; 68W10; 74B05; 74B20; 74S05

Keywords: Parallel algorithms; PCG method; Preconditioner; Circulant matrix; Elasticity problem

1. Introduction

This work concerns new efficient parallel algorithms and the related program software for solving the elasticity problem of the geomechanics in geosciences. Typical application problems include the simulations of the foundation of engineering constructions (which transfer and distribute the total loading into the bed soil) and the multi-layer media with strongly varying material characteristics. Here, the spatial framework of the construction produces a composed stressed–strained state in active interaction

* Tel.: +359-29796629; fax: +359-2722349.
E-mail address: ivan@parallel.bas.bg (I. Lirkov).
URL: http://parallel.bas.bg/∼ivan/

0378-4754/02/$ – see front matter © 2002 IMACS. Published by Elsevier Science B.V. All rights reserved.
PII: S0378-4754(02)00104-0

zones. A modern design of cost-efficient construction with a sufficient guaranteed reliability requires to determine the parameters of this stressed–strained state.

These application problems are modeled mathematically as 3D nonlinear elasticity problem described by a system of nonlinear partial differential equations. A finite element (or finite difference) discretization reduces the partial differential equation problem to a system of linear/nonlinear equations. To make a reliable prediction of the construction safety, which is sensitive to soil deformations, a very accurate model and a large system of sparse linear equations is required. In the real-life applications, the system can be very large containing up to several millions of unknowns. Hence, these problems have to be solved by robust and efficient parallel iterative methods (see [1]) on a powerful multiprocessor machine.

Note that the numerical solution of linear systems is fundamental in the elasticity problem. In fact, nonlinear equations generated from the discretization of the nonlinear elasticity problem have to be solved by an iterative procedure, in which a system of linear equations has to be solved in every step of iteration. Solving these linear systems is usually very time-consuming (costing up to 90% of the total solution time). Hence, developing fast algorithms for solving linear equations becomes the most important and fundamental issue. A highly efficient iterative method for solving linear systems significantly speed up the simulation processes of real application problems. An efficient iterative solver should not only have a fast convergence rate but also a high parallel efficiency. Moreover, the resulting program should be efficiently implemented on modern shared-memory, distributed memory, and shared–distributed memory parallel computers.

2. Elasticity problems

The foundations of engineering constructions transfer and distribute the total loading into the bed soil. For simplicity, we mainly study the 3D linear elasticity problem based on the following *two basic assumptions*: (1) the displacements are small, and (2) the material properties are isotropic.

The mathematical formulation of the 3D elasticity problem is described as follows. Let $\underline{u} = (u_1, u_2, u_3)^{\mathrm{T}}$ be the displacement vector and $\underline{p}$ the volume force vector. Here T denotes the transpose of a vector or a matrix. Let us denote the matrices D, G and H by

$$D = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 & 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{pmatrix}, \qquad G = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{2\partial x_2} & \frac{\partial}{2\partial x_1} & 0 \\ 0 & \frac{\partial}{2\partial x_3} & \frac{\partial}{2\partial x_2} \\ \frac{\partial}{2\partial x_3} & 0 & \frac{\partial}{2\partial x_1} \end{pmatrix},$$

$$H = \begin{pmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-2\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-2\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-2\nu) \end{pmatrix}.$$

Then the strain vector $\epsilon = (\epsilon_{11}, \epsilon_{22}, \epsilon_{33}, \epsilon_{12}, \epsilon_{23}, \epsilon_{31})^{\mathrm{T}}$ and the stress vector $\sigma = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{31})^{\mathrm{T}}$ are determined by

$$\epsilon = G u \quad \text{and} \quad \sigma = E^* H \epsilon, \tag{1}$$

where $E^* = E/(1+\nu)(1-2\nu)$. Here, $\nu = \nu(\epsilon)$ and $E = E(\epsilon)$ are, respectively the Poisson ratio and the deformation module.

With the earlier notation, the 3D elasticity problem in the stressed–strained state, on a computational domain Ω, can be described by a coupled system of 3D equations, which can be written in the form

$$\begin{cases} D\sigma = -\underline{p} & \text{in } \Omega \\ u = u_{\mathrm{D}} & \text{on } \Gamma_{\mathrm{D}} \\ \sum_{i=1}^{3} \sigma_{ij} n_i = \sigma_{Nj} & \text{on } \Gamma_{\mathrm{N}}, \quad j = 1, 2, 3, \end{cases} \tag{2}$$

where Γ_{D} and Γ_{N} are the parts of the boundary of Ω with respectively Dirichlet and Neumann boundary conditions; and u_{D} and σ_{N} are respectively the given displacement and stress vectors on the boundaries Γ_{D} and Γ_{N}. Here we set $\sigma_{ji} = \sigma_{ij}$ for $i < j$.

If the Poisson ratio and the deformation module are nonlinear functions, the relations (1) represent the nonlinear nature of the generalized Hooke's law. Here the generalized Hooke's law is specified by the following additional assumption: the Poisson ratio $\nu \in (0, 1/2)$ is a constant for a given material (soil layer or constructive element). Obviously, this means that the coefficients in the boundary value problem (2) are piece-wise continuous with jumps through the inner boundaries between the different soil layers as well as between the soil and the construction elements.

With a linearization, the nonlinear equations given in (2) can be simplified to a system of three linear differential equations, which is often referred to as the Lamé equations.

Denote Sobolev spaces $[H_E^1(\Omega)]^3 = \{v \in [H^1(\Omega)]^3 : v|_{\Gamma_{\mathrm{D}}} = u_{\mathrm{D}}\}$ and $[H_0^1(\Omega)]^3 = \{v \in [H^1(\Omega)]^3 : v|_{\Gamma_{\mathrm{D}}} = 0\}$. The variational formulation of the Lamé equations is given as follows:

$$\text{find } u \in [H_E^1(\Omega)]^3 \quad \text{such that} \quad \forall v \in [H_0^1(\Omega)]^3$$

$$\int_{\Omega} \left[\lambda \operatorname{div} u \operatorname{div} v + 2\mu \sum_{i=1}^{3} \sum_{j=1}^{3} \epsilon_{ij}(u) \epsilon_{ij}(v) \right] \mathrm{d}\Omega = -\int_{\Omega} \underline{p}^{\mathrm{T}} v \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{N}}} \sigma_{\mathrm{N}}^{\mathrm{T}} v \, \mathrm{d}\Gamma,$$

where λ and μ are the Lamé positive constants. Here $\operatorname{div} u$ is the divergence of the vector u. The relations between the elasticity modulus E, ν and the material parameters λ, μ are

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

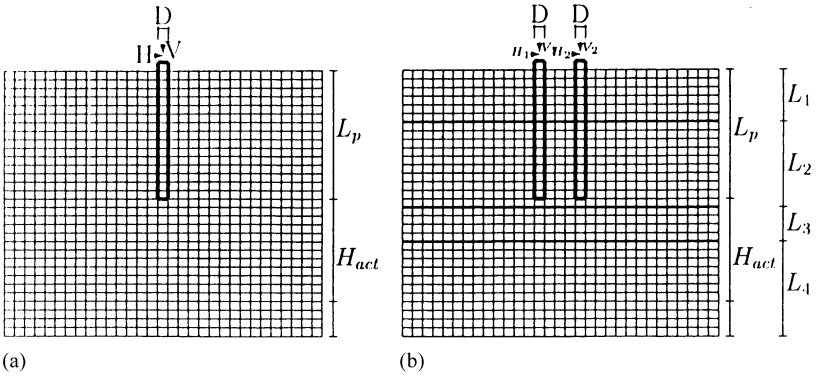

Fig. 1. Benchmark problems: (a) Problem 1: Cross-section of the computational domain Ω. $E_{\text{soil}} = 10\,\text{MPa}$, $\nu_{\text{soil}} = 0.3$, $E_{\text{pile}} = 31500\,\text{MPa}$, $\nu_{\text{pile}} = 0.2$; (b) Problem 2: Cross-section of the computational domain Ω. $E_{L_1} = 5.2\,\text{MPa}$, $\nu_{L_1} = 0.4$, $E_{L_2} = 9.4\,\text{MPa}$, $\nu_{L_2} = 0.35$, $E_{L_3} = 14.0\,\text{MPa}$, $\nu_{L_3} = 0.25$, $E_{L_4} = 21.4\,\text{MPa}$, $\nu_{L_4} = 0.2$.

We restrict our considerations to the case $\Omega = [0, x_1^{\max}] \times [0, x_2^{\max}] \times [0, x_3^{\max}]$, where the boundary conditions on each of the sides of Ω are of a fixed type. The benchmark problems from [4] are used in the reported numerical tests. These benchmarks represent the model of a single pile in a homogeneous sandy clay soil layer (see Fig. 1(a)) and two piles in an inhomogeneous sandy clay soil (Fig. 1(b)). A uniform grid is used with n_1, n_2 and n_3 grid points along the coordinate directions. Then the stiffness matrix K can be written in a 3×3 block form where the blocks K_{ij} are sparse block-tridiagonal matrices of a size $n_1 n_2 n_3$.

3. DD CBF preconditioning

First, let us recall that a $m \times m$ circulant matrix C has the form $(C_{k,j}) = (c_{(j-k) \bmod m})$. Each circulant matrix can be factorized as $C = F \Lambda F^*$, where Λ is a diagonal matrix of the eigenvalues of C, and F is the Fourier matrix $F = (1/\sqrt{m})\{e^{2\pi(jk/m)\mathrm{i}}\}_{0 \le j,k \le m-1}$. Here i stands for the imaginary unit.

3.1. A displacement decomposition-based preconditioner

There are a lot of works dealing with preconditioning of iterative solution methods for the FEM elasticity systems. In [2] Axelsson and Gustafsson construct their preconditioners based on the point-ILU factorization of the displacement decoupled block-diagonal part of the original matrix. This approach is known as *displacement decomposition* (see, e.g. [3]).

To define the displacement decomposition preconditioner M_{DD} of the matrix K, we introduce the auxiliary Laplace equation $-u_{x_1x_1} - u_{x_2x_2} - u_{x_3x_3} = f$, with boundary conditions corresponding to the considered coupled elasticity problem. Let us primary assume, that this Laplace equation is discretized by the same brick finite elements as the original problem, and let K_0 be the obtained stiffness matrix.

The following estimate based on Korn's inequality gives the theoretical background of the displacement decomposition methods [2]:

$$\kappa(M_{\mathrm{DD}}^{-1}K) = \mathcal{O}\left(\frac{1}{1-2\nu_{\max}}\right), \tag{3}$$

where $\nu_{\max} = \max_{\Omega}\nu$ and

$$M_{\mathrm{DD}} = \mathrm{diag}(K_0, K_0, K_0). \tag{4}$$

The next step in our construction is to substitute in (4), K_0 by A_0, where A_0 stands for the Laplace stiffness matrix corresponding to linear finite elements or, which is equivalent in the case under consideration, to a seven-point finite difference stencil. This step is motivated by the more simple/sparse structure of A_0 as well as by the spectral equivalence

$$\kappa(A_0^{-1}K_0) = \mathcal{O}(1). \tag{5}$$

3.2. Circulant block factorization

The CBF preconditioning technique (see [5]) incorporates the circulant approximation into the framework of the LU block factorization. It was recently analyzed for the model Dirichlet boundary value problem

$$-(a(x_1,x_2,x_3)u_{x_1})_{x_1} - (b(x_1,x_2,x_3)u_{x_2})_{x_2} - (c(x_1,x_2,x_3)u_{x_3})_{x_3} = f(x_1,x_2,x_3)$$

in $\Omega = [0, x_1^{\max}] \times [0, x_2^{\max}] \times [0, x_3^{\max}]$. Let us assume (as in Section 2) that Ω is discretized by a uniform grid with n_1, n_2 and n_3 grid points along the coordinate directions, and that a standard (for such a problem) seven-point finite difference (FEM) approximation is used. The related stiffness matrix $A^{(d)}$ can be written in the block-form $A^{(d)} = \mathrm{tridiag}(-A_{i,i-1}^{(d)}, A_{i,i}^{(d)}, -A_{i,i+1}^{(d)})$, $i = 1, 2, \ldots, n_1$, where $A_{i,i}^{(d)}$ is a block-tridiagonal matrix corresponding to the ith x_1-plane, and the off-diagonal blocks are diagonal matrices. Now, CBF preconditioner M_{CBF} is defined as follows:

$$M_{\mathrm{CBF}} = \mathrm{tridiag}(-C_{i,i-1}, C_{i,i}, -C_{i,i+1}), \qquad i = 1, 2, \ldots, n_1. \tag{6}$$

Here $C_{i,j} = \mathrm{BC}(A_{i,j}^{(d)})$ is a block-circulant approximation of the corresponding block $A_{i,j}^{(d)}$. The relative condition number of the CBF preconditioner for the model 3D problem is analyzed using the technique from [6] and the following estimate is derived:

$$\kappa(M_{\mathrm{CBF}}^{-1}A_0) \leq 2\max(n_2, n_3) + 2\sqrt{2}. \tag{7}$$

Now, let us denote by M_0 the CBF preconditioner for A_0, the matrix introduced in the previous subsection. At the last step of our construction we substitute in (4), K_0 by M_0, and get the DD CBF preconditioner defined by:

$$M_{\mathrm{DD\,CBF}} = \mathrm{diag}(M_0, M_0, M_0).$$

The estimate of the condition number of the preconditioned matrix

$$\kappa(M_{\mathrm{DD\,CBF}}^{-1}K) = \mathcal{O}\left(\frac{n_{\max}}{1-2\nu_{\max}}\right), \qquad \text{where } n_{\max} = \max(n_1, n_2, n_3) \tag{8}$$

follows straightforwardly from (3), (5) and (7).

Remark 1. We have observed in the performed numerical tests that a diagonal scaling of K improves the convergence rate of the iterative method in the case of problems with jumping coefficients.

4. Parallel tests of the DD CBF preconditioning FEM code

In this section, we report the results of the experiments executed on four parallel systems. We report here the elapsed time T_p on p processors, the speed-up $S_p = T_1/T_p$, and the parallel efficiency $E_p = S_p/p$. The benchmark problems were already described in Section 2. We used discretizations with $n_1 = n_2 = n_3 = n$ where $n = 32, 64$. The sizes of the discrete problems are 3.32^3 and 3.64^3.

The developed parallel code has been implemented in C and the parallelization has been facilitated using the MPI [7,8] library. In all cases, the manufacturer provided MPI kernels have been used. No machine-dependent optimization has been applied to the code itself. Instead, in all cases, the most aggressive optimization options of the compiler have been turned on. Times have been collected using the MPI provided timer. In all cases, we report the best results from multiple runs in interactive and batch modes (for varying workloads).

In Table 1, we present results of experiments executed on two shared memory machines: SUN Ultra-Enterprise Symmetric Multiprocessor with eight Ultra-SPARC 168 MHz processors and 1GB main

Table 1
Parallel time (in seconds), speed-up and parallel efficiency on shared memory machines

n	p	SUN Ultra-Enterprise 3000			Origin 2000		
		T_p	S_p	E_p	T_p	S_p	E_p
Problem 1							
32	1	383.35			74.45		
	2	186.77	2.05	1.026	47.19	1.58	0.789
	4	95.11	4.03	1.008	28.34	2.63	0.657
	8	50.12	7.65	0.956	13.89	5.36	0.670
	16				8.31	8.96	0.560
64	1	3753.39			1453.42		
	2	1824.00	2.06	1.029	700.95	2.07	1.037
	4	918.31	4.09	1.022	381.35	3.81	0.953
	8	472.62	7.94	0.993	207.05	7.02	0.877
	16				133.36	10.90	0.681
Problem 2							
32	1	1680.15			335.02		
	2	809.64	2.08	1.038	201.65	1.66	0.831
	4	400.39	4.20	1.049	110.73	3.03	0.756
	8	204.28	8.22	1.028	53.80	6.23	0.778
	16				27.53	12.17	0.761
64	1	20900.10			7932.28		
	2	10401.40	2.01	1.005	3643.01	2.18	1.089
	4	5197.65	4.02	1.005	2164.27	3.67	0.916
	8	2623.48	7.97	0.996	1069.57	7.42	0.927
	16				585.54	13.56	0.847

memory; and SGI Origin 2000 with 64 R10000 300 MHz processors and 64GB main memory. As expected, the parallel efficiency increases with the size of the discrete problems. The parallel efficiency is >50% which confirms our general expectations. There exist at least two reasons for the reported high efficiency: (a) the network parameters *start-up time* and *time for transferring of single word* are relatively small for the multiprocessor machines; (b) there is also some overlapping between the computations and the communications in the algorithm. Moreover, the super-linear speed-up can be seen in some of the runs. This effect has a relatively simple explanation. When the number of processors increases, the size of data per processor decreases. Thus the stronger *memory locality* increases the role of the cache memories.

Table 2 shows results obtained on a clusters consisting of: four dual processor PowerPC with G4 450 MHz processors and 512MB memory per node (Black Lab); and 10 dual processor DS20 AlphaServers with Compaq Alpha EV6 500 MHz processors and 768MB memory per node. The memory on one PowerPC is sufficient only for the discretization with $32 \times 32 \times 32$ grid points. For the larger problem, we report the parallel efficiency related to the results on two processors. The Alpha processors are the fastest from the reported machines. We suppose that the network on this cluster is not fast enough and this affects the performance of the parallel algorithm. Nevertheless, we point your attention that the presented results are very promising for $n = 64$ and $p = 2, 4$.

Table 2
Parallel time (in seconds), speed-up and parallel efficiency on clusters

n	p	Black Lab cluster			Alpha cluster		
		T_p	S_p	E_p	T_p	S_p	E_p
Problem 1							
32	1	125.99			57.03		
	2	67.39	1.87	0.935	46.69	1.22	0.611
	4	39.58	3.18	0.795	36.29	1.57	0.393
	8	23.92	5.27	0.658	18.17	3.14	0.392
	16				15.85	3.60	0.225
64	1	5528.39			1201.64		
	2	1190.63			817.75	1.47	0.735
	4	651.55		0.914	549.14	2.19	0.547
	8	362.58		0.821	534.86	2.25	0.281
	16				227.41	5.28	0.330
Problem 2							
32	1	541.30			245.09		
	2	295.71	1.82	0.912	200.97	1.22	0.610
	4	170.60	3.52	0.880	157.45	1.56	0.389
	8	101.50	4.79	0.598	77.71	3.15	0.394
	16				71.59	3.42	0.214
64	1	27090.40			6109.09		
	2	6141.29			4192.16	1.46	0.729
	4	3318.85		0.925	2847.01	2.15	0.536
	8	1861.66		0.825	2778.42	2.20	0.275
	16				1189.52	5.14	0.321

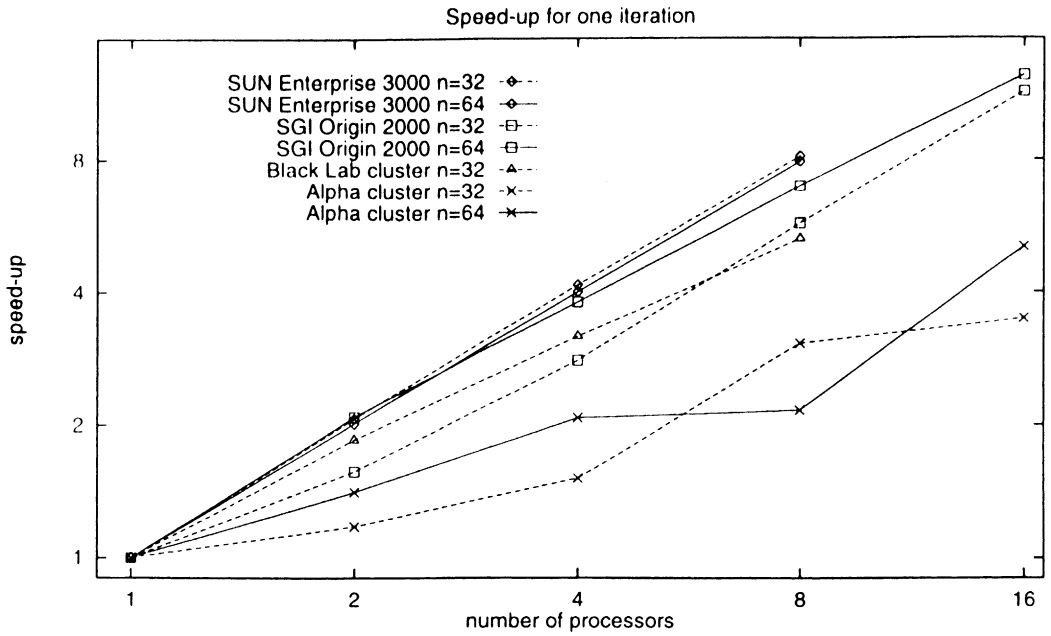

Fig. 2. Speed-up for one iteration.

Finally, Fig. 2 shows parallel speed-up for execution of one iteration on different parallel systems.

Acknowledgements

This research has been supported by the Bulgarian NSF grant I-1001/2000 and by Center of Excellence BIS-21 grant ICA1-2000-70016.

References

[1] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.
[2] O. Axelsson, I. Gustafsson, Iterative methods for the solution of the Navier equations of elasticity, Comp. Meth. Appl. Mech. Eng. 15 (1978) 241–258.
[3] R. Blaheta, Displacement decomposition—incomplete factorization preconditioning techniques for linear elasticity problems, Num. Lin. Alg. Appl. 1 (1994) 107–128.
[4] A. Georgiev, A. Baltov, S. Margenov, Hipergeos benchmark problems related to bridge engineering applications, PROJECT REPORT HG CP 94-0820-MOST-4.
[5] I. Lirkov, S. Margenov, P.S. Vassilevski, Circulant block-factorization preconditioners for elliptic problems, Computing 53 (1) (1994) 59–74.
[6] I. Lirkov, S. Margenov, L. Zikatanov, Circulant block-factorization preconditioning of anisotropic elliptic problems, Computing 58 (3) (1997) 245–258.
[7] M. Snir, St. Otto, St. Huss-Lederman, D. Walker, J. Dongara, MPI: The Complete Reference, Scientific and Engineering Computation Series, 3rd Edition, MIT Press, Cambridge, MA, USA, 1997.
[8] D. Walker, J. Dongara, MPI: a standard message passing interface, Supercomputer 63 (1996) 56–68.