Annual Report of
ADVENTURE Project
ADV-99-1(1999)

AN IMPLEMENTATION OF BALANCING DOMAIN DECOMPOSITION METHOD ON STATIC ELASTIC STRESS ANALYSES

Ryuji SHIOYA

Department of Intelligent Machinery and Systems
Graduate School of Engineering, Kyushu University
6-10-1, Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
e-mail: shioya@mech.kyushu-u.ac.jp, URL: <http://cm.mech.kyushu-u.ac.jp/~shioya>

Key Words: Parallel Finite Element Method, Domain Decomposition Method, Balancing Domain Decomposition, Massively Parallel Processors

1 INTRODUCTION

The iterative Domain Decomposition Method(DDM) is one of the most effective parallel methods for large scale problems due to its excellent parallelism and suitability for various kinds of parallel computers such as massively parallel processors and workstation/PC clusters [G.Yagawa and R.Shioya, 1994], [R.Shioya and G.Yagawa, 1998]. As the iterative DDM satisfies continuity among subdomains through iterative calculations such as the Conjugate Gradient(CG) method, it is indispensable to reduce the number of iterations with a preconditioning technique especially for large problems.

The Neumann-Neumann algorithm(NN) is known to be an efficient domain decomposition preconditioner with unstructured subdomains for an iterative solution of finite element discretization of difficult problems with strongly discontinuous coefficients [Y.H.De Roeck and P.Le Tallec, 1991]. However, this algorithm suffers from the need to solve in each iteration an inconsistent singular problem for every subdomain, and its convergence deteriorates with the increasing number of subdomains due to the lack of a coarse problem to propagate the error globally.

The Balancing Domain Decomposition method(BDD) based on NN introduced by Mandel [J.Mandel, 1993] shows that the equilibrium conditions for the singular problems on subdomains lead to the simple and natural construction of a coarse problem. The construction is purely algebraic and applies also to systems. In this report, an implementation of BDD to static elastic stress analyses is presented and some numerical experiments are performed.

In the following section, NN and BDD algorithms are presented, and the construction of the preconditioner of BDD for the Schur complement matrix is described. The application of the present implementation for the stress analyses and estimation of efficiency of BDD are described in section 3.

2 BALANCING DOMAIN DECOMPOSITION

2.1 Interface Problem

Consider a system of linear algebraic equations

$$Ku = f \tag{1}$$

arising from a finite element discretization of a linear, elliptic, self-adjoint boundary value problem on domain Ω. The matrix K is the stiffness matrix and assumed to be an $m \times m$, symmetric positive definite matrix.

The domain Ω is split into non-overlapping subdomains $\Omega^{(1)}, \ldots, \Omega^{(k)}$, each of which is the union of certain elements. Let $u^{(i)}$ be the vector of degrees of freedom corresponding to all elements in subdomain $\Omega^{(i)}$, and let $N^{(i)}$ denote the 0-1 matrix that maps the degrees of freedom $u^{(i)}$ into global degrees of freedom; then

$$u^{(i)} = N^{(i)T} u \tag{2}$$

and

$$K = \sum_{i=1}^{k} N^{(i)} K^{(i)} N^{(i)T} \tag{3}$$

where $K^{(i)}$ is the local stiffness matrix corresponding to subdomain $\Omega^{(i)}$. Each $u^{(i)}$ is split into degrees of freedom $u_B^{(i)}$ that correspond to the interface of the subdomain $\Omega^{(i)}$ with other subdomains, and the remaining degrees of freedom $u_I^{(i)}$, which are associated with the subdomain $\Omega^{(i)}$ only. The subdomain stiffness matrices and the 0-1 matrices $N^{(i)}$ are then split accordingly and the system (1) is,

$$\begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)T} & K_{BB}^{(i)} \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_B^{(i)} \end{pmatrix} = \begin{pmatrix} f_I^{(i)} \\ f_B^{(i)} \end{pmatrix} \tag{4}$$

and

$$N^{(i)} = (N_I^{(i)}, N_B^{(i)}) \tag{5}$$

After eliminating $u_I^{(i)}$, the system (1) becomes

$$Su_B = g \tag{6}$$

where S is the assembly of the Schur complements,

$$S = \sum_{i=1}^{k} N_B^{(i)} S^{(i)} N_B^{(i)T}, \qquad S^{(i)} = K_{BB}^{(i)} - K_{IB}^{(i)T} K_{II}^{(i)-1} K_{IB}^{(i)} \tag{7}$$

We assume that the subdomain matrices $K^{(i)}$ are symmetric and positive semidefinite, with the submatrices $K_{II}^{(i)}$ nonsingular. Then the Schur complements $S^{(i)}$ are also positive semidefinite.

A large number of domain decomposition (or substructuring) methods consist of solving the reduced system (6) iteratively [J.H.Bramble et al., 1989], [M.Dryja and O.B.Widlund, 1990]. Since S is symmetric positive definite, the preconditioned conjugate gradient method [P.Concus et al., 1976], [G.H.Golub and C.F.Van Loan, 1989] is the standard choice for iterative methods. This method requires at each step the solution of an auxiliary problem,

$$Mz = r \tag{8}$$

with a symmetric positive definite matrix M, called a preconditioner.

The BDD preconditioner is a variation of the following preconditioner due to De Roeck and Le Tallec [Y.H.De Roeck and P.Le Tallec, 1991], based on an earlier work for the case of two subdomains [J.F.Bourgat et al., 1989] and a closely related method of Glowinski and Wheeler for mixed problems [R.Glowinski and M.F.Wheeler, 1988]. It is also called the Neumann-Neumann preconditioner(NN) because of an interpretation in terms of boundary conditions on interfaces of the subdomains [M.Dryja and O.B.Widlund, 1990]. NN algorithm is described in the next section.

2.2 Neumann-Neumann Preconditioning algorithm

The method uses a collection of matrices $D^{(i)}$ that form a decomposition of unity,

$$\sum_{i=1}^{k} N^{(i)} D^{(i)} N^{(i)T} = I \tag{9}$$

The simplest choice for $D^{(i)}$ is the diagonal matrix with diagonal elements equal to the reciprocal of the number of subdomains with which the degrees of freedom is associated [Y.H.De Roeck and P.Le Tallec, 1991].

Given r, compute $M^{-1}r$ as follows.

- **Step 1:** Distribute r to the subdomains $\Omega^{(i)}$

$$r^{(i)} = D^{(i)T} N_B^{(i)T} r, \qquad i = 1, \ldots, k \tag{10}$$

- **Step 2:** Solve the local problems for all subdomains

$$S^{(i)} z^{(i)} = r^{(i)}, \qquad i = 1, \ldots, k \tag{11}$$

- **Step 3** Average the local solutions on the interfaces

$$z = \sum_{i=1}^{k} N_B^{(i)} D^{(i)} z^{(i)} \tag{12}$$

Unfortunately, $K^{(i)}$ and thus $S^{(i)}$ are typically singular. In this case, De Roeck and Le Tallec [Y.H.De Roeck and P.Le Tallec, 1991] suggest modifying the Gaussian elimination algorithm for solving (11) to avoid zero pivots and thus solve (11) only approximately. A different approximate solution was proposed by Dryja and Widlund [M.Dryja and O.B.Widlund, 1990].

Another drawback of NN is the lack of a mechanism to exchange information between all subdomains in the preconditioning step and thus to prevent the growth of the condition number with the number of subdomains. Indeed, it was observed experimentally that the condition number of NN algorithm deteriorates with the number of subdomains, and that 16 subdomains is a practical limit [P.Le Tallec et al., 1991]. In other domain decomposition methods, the needed propagation of error has been accomplished by solving a coarse problem with few degrees of freedom per subdomain in each iteration [J.H.Bramble et al., 1986], [J.H.Bramble et al., 1989], [J.Mandel, 1990], [B.F.Smith, 1991], [B.F.Smith, 1992]. Such methods are similar to multi-grid methods and especially to two-level methods such as in [O.Axelsson and I.Gustafsson, 1983], [C.Farhat, 1989] and [M.Kocvara and J.Mandel, 1987].

Several such coarse problems for the NN algorithm have been suggested in [M.Dryja and O.B.Widlund, 1990], based on earlier coarse problems in different contexts. A related method was introduced and tested in a parallel environment by Farhat and Roux [C.Farhat and F.X.Roux, 1991]. Their method enforces continuity on subdomain interfaces by Lagrange multipliers and uses a coarse problem to obtain consistency of the singular problems associated with subdomains.

Unlike the coarse problems in [M.Dryja and O.B.Widlund, 1990], no part of the BDD algorithm needs to know what the faces, edges, or vertices of the subdomains are. The subdomains can be completely unstructured, and the algorithm is formulated in purely algebraic terms. The BDD algorithm is described in the next section.

2.3 Balancing Domain Decomposition algorithm

Given r, compute $M^{-1}r$ as follows.

- **Step 1:** Solving the auxiliary problem for unknown vectors λ_i

$$Z^{(i)T} D^{(i)T} N^{(i)T} (r - S \sum_{j=1}^{k} N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)}) = 0, \qquad i = 1, \ldots, k \tag{13}$$

- **Step 2:** Balance the original residual r using $\lambda^{(j)}$

$$s = r - S\sum_{j=1}^{k} N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)}, \qquad s^{(i)} = D^{(i)T} N_B^{(i)T} s, \quad i = 1, \ldots, k \tag{14}$$

- **Step 3:** Find any solution $u^{(i)}$ for each of the local problems

$$S^{(i)} u^{(i)} = s^{(i)}, \qquad i = 1, \ldots, k \tag{15}$$

- **Step 4:** Balance the residual by solving the auxiliary problem for $\mu^{(i)}$

$$Z^{(i)T} D^{(i)T} N^{(i)T} (r - S\sum_{j=1}^{k} N_B^{(j)} D^{(j)} (u^{(j)} + Z^{(j)} \mu^{(j)})) = 0, \qquad i = 1, \ldots, k \tag{16}$$

- **Step 4:** Average the result on the interfaces

$$z = \sum_{i=1}^{k} N_B^{(i)} D^{(i)} (u^{(i)} + Z^{(i)} \mu^{(i)}) \tag{17}$$

where n_i : dimension of $K_{IB}^{(i)}$, $m_i : 0 \leq m_i \leq n_i$ and Z_i be $n_i \times m_i$ matrices of full column rank such that

$$\text{Null } S_i \subset \text{ Range } Z_i, \qquad i = 1, \ldots, k \tag{18}$$

2.4 Z_i for elastic problem

Using BDD, we should determine an efficient Z_i which satisfies (18). For an elastic problem, Null S_i in (18) can be considered to correspond to the degrees of freedom of rigid displacement. Let $v(x)$ be the vector of rigid displacement; then

$$v(x) = a + b \times x \tag{19}$$

where $a = (a_1, a_2, a_3)$ and $b = (b_1, b_2, b_3)$ are arbitrary vectors and $x = (x_1, x_2, x_3)$ is the coordinates of a point. Let V_i be the space of the interface degrees of freedom for the subdomain Ω_i, $n_i = \dim V_i$, then at the point $P(x_1, x_2, x_3)$ on the interface of the subdomain $\Omega^{(i)}$, (19) can be written as:

$$v(x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{20}$$

Now let $Z_i^{(P)}$ on P be defined as:

$$Z_i^{(P)} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & x_3 & -x_2 \\ 0 & 1 & 0 & -x_3 & 0 & x_1 \\ 0 & 0 & 1 & x_2 & -x_1 & 0 \end{pmatrix} \tag{21}$$

and Z_i be defined by assembling $Z_i^{(P)}$,

$$Z_i \equiv \sum_{P \in \Gamma_i} B_i^{(P)} Z_i^{(P)} B_i^{(P)T} \tag{22}$$

where $B_i^{(P)}$ is the 0-1 matrix that maps the degrees of freedom P into global degrees of freedom of the interface of the subdomain $\Omega^{(i)}$. Now, with arbitrary $\lambda = (a_1, a_2, a_3, b_1, b_2, b_3)^T$, (19) becomes

$$v(P) = Z_i \lambda \tag{23}$$

and it satisfies (18). Numerical experiments with this Z_i, are discussed in the following section.

3 NUMERICAL EXPERIMENTS

The current system was applied to the finite element analyses of two types of models, i.e. the cubic and beam structures shown in Figures 1 and 3, respectively. These models are expressed by a 10-noded tetrahedron elements.

For the cubic model, to estimate the relation between the number of iterations and subdomains, this structure is divided into 8, 27 and 64 subdomains, each subdomain including six elements. The size of models are 48, 162 and 384 elements and 125, 343 and 729 nodes, respectively. Using these three sizes of models, DDM without any preconditioner and BDD are performed.

The degrees of freedom on the interface of the subdomain, which affects the number of iterations for CG, and the number of iterations until convergence are shown in Table 1. Comparing with the increasing number of iterations for the number of subdomains in case of DDM, that of BDD increases very little. The force imbalance measures within the inter subdomain measure (residual value) are shown in Figure 2 in contrast to the number of CG iterations.

For the beam model, this structure is modeled by 7,222 elements, 13,327 nodes, the total degrees of freedom is 39,981 and the whole structure is divided into 148 subdomains. The degrees of freedom of the interface of the subdomain is 14,484. For this model, normal DDM without any preconditioner, DDM with diagonal preconditioner and BDD are performed. Here, diagonal preconditioner is defined by assembling of diagonal elements of Schur complements $S^{(i)}$ of (7).

To estimate the effect of the preconditioner, two types of boundary conditions are tested, i.e., model 1: with only Dirichlet boundary condition type and model 2: with Dirichlet and Neumann boundary condition types. These problems were solved by one

processor, Ultra Sparc 1/200MHz with 512 MBytes memory and calculation times, the number of iterations and memory usage of models 1 and 2 are shown in Tables 2 and 3, respectively.

As shown in the tables, although BDD requires more memory than other methods, reducing the number of iterations and speeding up the calculation time are achieved for both cases.

The force imbalance measure within the inter subdomain measure (residual value) of model 1 and model 2 are shown in Figures 4 and 5 in contrast to the number of CG iterations.

4 CONCLUSIONS

The finite element system based on the DDM with preconditioner using BDD was developed in the current study. This system can be applied to simple and small size of structural analyses and effective performances were obtained. To apply for a larger model like over ten million DOF problem, parallelising of the system and reducing memory usage of BDD are required.

REFERENCE

O.Axelsson and I.Gustafsson, Preconditioning and two-level multigrid methods of arbitrary degree of approximation, *Math. Comp.* **40** (1983) 219-242.

J.F.Bourgat, R.Glowinski, P.Le Tallec and M.Vidrascu, Variational formulation and algorithm for trace operator in domain decomposition calculations, *in Domain Decomposition Methods, SIAM, Philadelphia, PA* (1989)

J.H.Bramble, J.E.Pasciak and A.H.Schatz, The construction of preconditioners for elliptic problems by substructuring, I, *Math. Comp.* **47** (1986) 103-134

J.H.Bramble, J.E.Pasciak and A.H.Schatz, The construction of preconditioners for elliptic problems by substructuring, *Math. Comp.* **IV 53** (1989) 1-24.

P.Concus, G.H.Golub and D.P.O'Leary, A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations, *in Sparse Matrix Computations, Academic Press* (1976) 309-332.

Y.H.De Roeck and P.Le Tallec, Analysis and test of a local domain decomposition preconditioner, *in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA* (1991)

M.Dryja and O.B.Widlund, Towards a unified theory of domain decomposition algorithms for elliptic problems, *in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA* (1990)

C.Farhat, A multigrid-like semi-iterative algorithm for the massively parallel solution of large scale finite element systems, *in Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods, SIAM, Philadelphia, PA* (1989) 171-180.

C.Farhat and F.X.Roux, An unconventional domain decomposition method for an efficient

parallel solution of large-scale finite element systems, *SIAM J. Sci. Stat. Comput.* **13** (1992) 379-396.
R.Glowinski and M.F.Wheeler, Domain decomposition and mixed finite element methods for elliptic problems, *in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA* (1988)
G.H.Golub and C.F.Van Loan, Matrix Computations, *John Hopkins University Press, second ed.* (1989).
M.Kocvara and J.Mandel, A multigrid method for three-dimensional elasticity and algebraic convergence estimates, *Appl. Math. Comput.* **23** (1987) 121-135.
P.Le Tallec, Y.H.De Roeck and M.Vidrascu, Domain decomposition methods for large linearly elliptic three dimensional problems, *J. of Computational and Applied Mathematics* **34** (1991)
J.Mandel, Two-level domain decomposition preconditioning for the p-version finite element method in three dimensions, *Int. J. Numer. Methods Engrg.* **29** (1990) 1095-1108.
J.Mandel, Balancing Domain Decomposition, *Communications on Numerical Methods in Engineering* **9** (1993) 233-241.
R.Shioya and G.Yagawa, Parallel Finite Elements of Ten-Million DOFs Based on Domain Decomposition Method *WCCM IV, Computational Mechanics -New Trends and Applications-* **VII, Section 5, 11** (1998) 1-12.
B.F.Smith, A domain decomposition algorithm for elliptic problems in three dimensions, *Numer. Math.* **60** (1991) 219-234.
B.F.Smith, An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems, *SIAM J. Sci. Stat. Comput.* **13** (1992) 364-378.
G.Yagawa and R.Shioya, Parallel finite elements on a massively parallel computer with domain decomposition, *Computing systems in Engineering* **4(4-6)** (1994) 495-503.

Table 1: Number of Iterations of DDM & BDD for Cubic Model

Num of Dom	D.O.F. of Interface	DDM	BDD
8	376	129	18
27	1,588	236	22
64	4,104	332	25

Table 2: Number of Iterations and Calculation Costs(Model 1)

	Num of Iter.	cpu time[hour]	Memory [MByte]
DDM	3,517	43.0	4.4
SCALING	1,756	21.5	4.5
BDD	505	12.6	145.0

Table 3: Number of Iterations and Calculation Costs(Model 2)

	Num of Iter.	cpu time[hour]	Memory [MByte]
DDM	2,163	26.4	4.4
SCALING	984	12.1	4.5
BDD	215	5.4	152.0

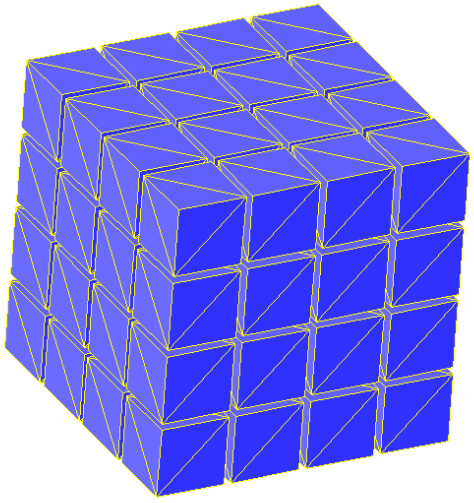

Figure 1: Mesh of Box model

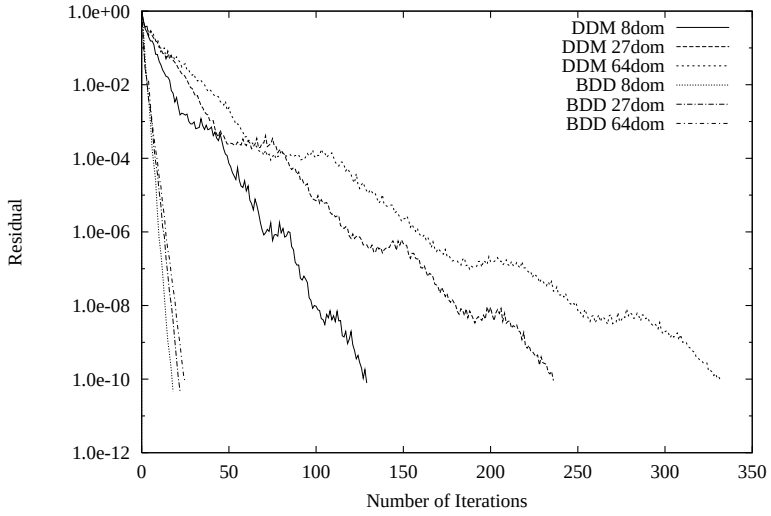

Figure 2: Residual vs number of iterations

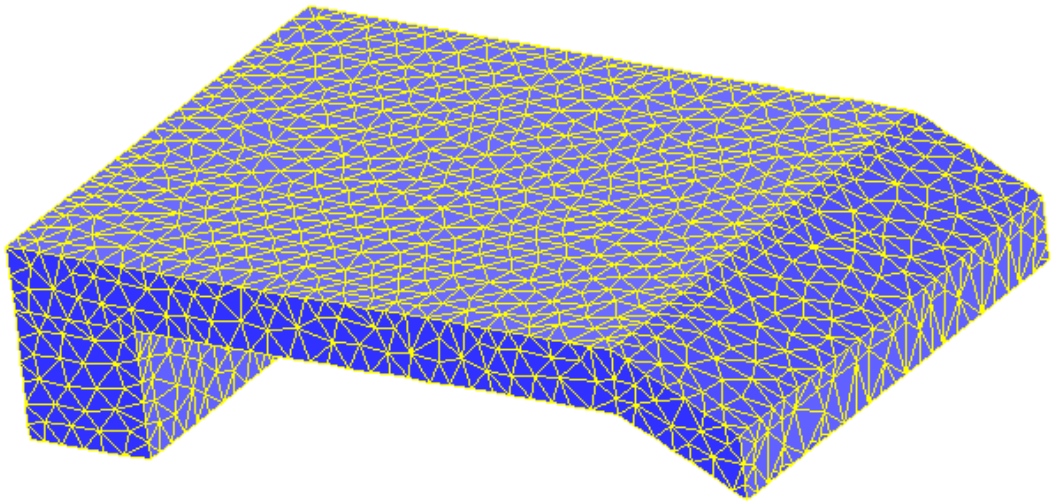

Figure 3: Mesh of Beam model

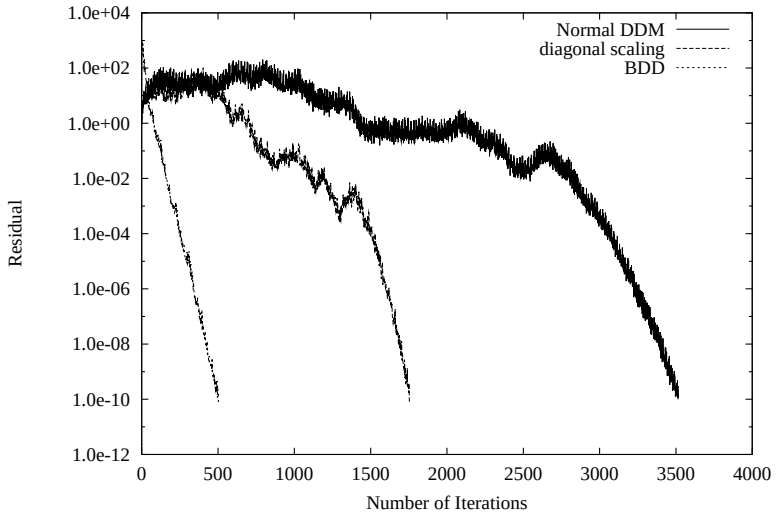

Figure 4: Residual vs number of iterations

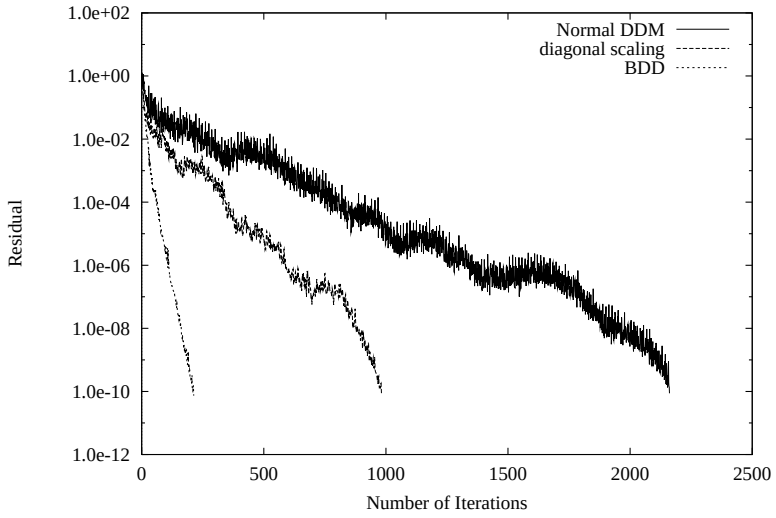

Figure 5: Residual vs number of iterations