Comparing Quasi-Newton Methods for Solving Sparse Interface Problems

C.-H. Lai
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract

We examine some quasi-Newton methods for interface problems which arise from domain decomposition methods. These interface problems are usually sparse systems of linear or non-linear equations. We are particularly interested to apply these methods to systems of linear equations where we are not able or willing to calculate the Jacobian matrices. We also examine the suitability of implementing these algorithms on coarse-grained parallel computers.

1991 Mathematics Subject Classification : 39B12, 65-04, 65B99, 65F10
1991 CR Categories : G.1.0, G.1.5

Keywords & Phrases : Quasi-Newton methods, domain decomposition, interface problems, coarse-grained parallel computers.

Note : This research was done during a visit to CWI supported by an ERCIM Fellowship. The current address of the author is School of Mathematics, Statistics, and Computing, University of Greenwich, Wellington Street, Woolwich, London SE18 6PF, U.K. email : c.h.lai@greenwich.ac.uk

Report NM-R9303
ISSN 0169-0388
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

1. Introduction

Some quasi-Newton methods for interface problems which arise from domain decomposition methods are compared. Such interface problems are obtained by applying a non-overlapped domain decomposition technique to given differential problems. The original problem is then reduced to an interface problem defined along the non-overlapped interfaces. The interface problem is described by a defect equation [5] specially designed to couple the interface in between two non-overlapped subdomains. Our technique is to find numerical solutions to the defect equation $\mathbf{D}(\boldsymbol{\lambda}) = 0$ where $\mathbf{D} : R^s \rightarrow R^s$ is the interface problem. In most cases, it is possible to know the sparsity pattern of the Jacobian matrix. However, we are not able to calculate the Jacobian matrix analytically. Furthermore, we are not willing to find a finite difference approximation to the Jacobian matrix, because the computational work is expensive. Our motivation is to study a class of quasi-Newton methods which would allow efficient computation of a sequence of approximations to the Jacobian matrix and hence an efficient algorithm for the numerical solutions of interface problems. The approach is different from most of the current domain decomposition approaches [3] where the subdomain solutions are treated as a preconditioner to be used in a conjugate gradient algorithm applied to the entire discretised problem. Some related work in applying a fixed point iteration scheme for interface problems can be found in Ref. [5].

The organisation of this report is as follows. First, we briefly describe the formulation of the interface problem to a given differential problem. We also comment on the advantages of solving interface problems in different physical situations. Second, we describe a number of quasi-Newton methods, in particular when these methods are applied to linear systems and we also present the convergence properties of these methods. Third, we examine the suitability for parallel implementation of the algorithms on coarse-grained parallel computers with shared or distributed memory. In particular, we are interested to know the minimum number of subdomains such that the parallel implementation of the algorithms can be regarded as efficient. Finally, we present some numerical results to a 1-D convection-diffusion problem and our conclusions and future research direction.

2. Formulation of the Interface Problem

We confine our description to the interface problem of the following two-point boundary value problem,

$$-\frac{d^2\phi}{dx^2} + c(x)\frac{d\phi}{dx} + r(x)\phi = 0, \quad \in \quad \Omega = \{x | a < x < b\} \tag{1}$$

subject to Dirichlet boundary conditions $\phi(a) = \phi_a$ and $\phi(b) = \phi_b$ where ϕ_a and ϕ_b are given constants. Under the conditions $0 < \mathcal{C}_l \leq c(x) \leq \mathcal{C}_u$ and $0 \leq \mathcal{R}_l \leq r(x) \leq \mathcal{R}_u$, ϕ is unique and $\phi \in C^2(\Omega)$. The domain Ω is split into $s+1$ nonoverlapped subdomains, Ω_k, $k=1,2,\cdots,s+1$, such that

$$\Omega = \{\cup_{k=1}^{s+1}\Omega_k\} \cup \{\cup_{k=1}^{s}\Gamma_k\} \tag{2}$$

where $\Omega_k = \{x|x_{k-1} < x < x_k\}$, $\partial\Omega_k = \{x_{k-1}, x_k\}$, $\Gamma_k = x_k$. Each of the subdomains has the following related two-point boundary value problem,

$$-\frac{d^2u_k}{dx^2} + c(x)\frac{du_k}{dx} + r(x)u_k = 0, \quad \in \quad \Omega_k \tag{3}$$

subject to boundary conditions $u_k(x_{k-1}) = \lambda_{k-1}$ and $u_k(x_k) = \lambda_k$, and $u_1(x_0) = \phi_a$ and $u_{s+1}(x_{s+1}) = \phi_b$ where $x_0 = a$, $x_{s+1} = b$. Let $u_k = u_k(x;\boldsymbol{\lambda})$ denote the solution of (3) in Ω_k, where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \ \cdots \ \lambda_s] \in \Omega_D \subset R^s$. In order to obtain unique values of $\phi'(x_k)$, $k=1,2,\cdots,s$, we define the defect $\mathbf{D} : \Omega_D \subset R^s \rightarrow R^s$ as

$$\mathbf{D}(\boldsymbol{\lambda}) = [D_k(\boldsymbol{\lambda})] \equiv [\frac{\partial}{\partial x}u_k(x_k;\boldsymbol{\lambda}) - \frac{\partial}{\partial x}u_{k+1}(x_k;\boldsymbol{\lambda})] \tag{4}$$

and require to solve the defect equation $\mathbf{D}(\boldsymbol{\lambda}) = 0$. The continuity of the function ϕ across the interfaces is implicit in (3). The defect equation represents the reduced interface problem and guarantees the continuity of ϕ' across the interfaces. It can be easily seen that $\mathbf{D} \in C^1(\Omega_D)$. In the two subdomain case, the defect equation is a scalar equation involving one interface and thus only one unknown. In the multidimensional case, the Jacobian matrix $J(\boldsymbol{\lambda}) = \mathbf{D}'(\boldsymbol{\lambda})$ is a nonsymmetric tridiagonal matrix [5]. If $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ is a root of $\mathbf{D}(\boldsymbol{\lambda}) = 0$, then the function

$$\phi(x) = \begin{cases} \lambda_{k-1}^* & x = x_{k-1} \\ u_k(x;\boldsymbol{\lambda}^*) & x_{k-1} < x < x_k, \\ \lambda_k^* & x = x_k \end{cases} \qquad k = 1,2,\cdots,s+1 \tag{5}$$

where $\lambda_0^* = \phi_a$ and $\lambda_{s+1}^* = \phi_b$, is a solution of (1).

The coupling of subdomain solutions is incorporated into a defect equation which involves unknowns along the subdomain interfaces. This is particularly important in the coupling of different mathematical models in different subdomains such as viscous and inviscid coupling in aerodynamic applications. Once the mathematical interface coupling is defined, the defect equation can be easily set up and contributions to the defect equation from different subdomains can be separately computed which ensures highly parallel computing tasks.

3. Quasi-Newton Methods

In this section we consider various quasi-Newton methods for the solution of the interface problem governed by the defect equation $\mathbf{D}(\boldsymbol{\lambda}) = 0$. Let

$\mathbf{D} : \Omega_D \subset R^s \to R^s$ where Ω_D is an open set, $\mathbf{D} \in C^1(\Omega_D)$, $\mathbf{D}(\boldsymbol{\lambda}^*) = 0$, $J(\boldsymbol{\lambda}^*)$ nonsingular, and, for all $\boldsymbol{\lambda} \in \Omega_D$

$$||J(\boldsymbol{\lambda}) - J(\boldsymbol{\lambda}^*)|| \leq L||\boldsymbol{\lambda} - \boldsymbol{\lambda}^*||^p \tag{6}$$

for some norm $||.||$, and some L, $p > 0$. The general quasi-Newton method for the solution of $\mathbf{D}(\boldsymbol{\lambda}) = 0$ is given by

$$\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} - \alpha_n^{-1}\mathbf{D}(\boldsymbol{\lambda}^{(n)}) \tag{7}$$

where $\alpha_n \in \mathbf{L}(R^s)$, and $\mathbf{L}(R^s)$ being a finite dimensional linear space. We have the following convergence result.

Theorem 3.1 (Dennis and Moré [1]) Suppose $\boldsymbol{\lambda}^*$ satisfies the above assumptions. Then given $r \in (0,1)$, there exists $\varepsilon = \varepsilon(r)$, $\delta = \delta(r)$ such that if $||\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^*|| \leq \varepsilon$ and $||\alpha_n - J(\boldsymbol{\lambda}^*)|| \leq \delta$ for all $n = 0, 1, 2, \ldots$, the sequence generated by (7) is well defined, converges to $\boldsymbol{\lambda}^*$ and satisfies

$$||\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^*|| \leq r||\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^*||$$

for all $n = 0, 1, 2, \ldots$.

It is also proved in [1] that $\{\boldsymbol{\lambda}^{(n)}\}$ converges Q-superlinearly to $\boldsymbol{\lambda}^*$ if and only if

$$\lim_{n\to\infty} \frac{||(\alpha_n - J(\boldsymbol{\lambda}^*))(\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)})||}{||\boldsymbol{\lambda}^{(n+1} - \boldsymbol{\lambda}^{(n)}||} = 0$$

One classical technique of choosing α_n is called Broyden's update. Let $Q_{u,v} = \{\hat{\alpha} \in \mathbf{L}(R^s), \mathbf{u}, \mathbf{v} \in R^s : \hat{\alpha}\mathbf{u} = \mathbf{v}\}$. Then Broyden's update is obtained as the solution to the minimisation problem [2]

$$\min\{||\hat{\alpha} - \alpha_n||_F : \hat{\alpha} \in Q_{\mathbf{s}_n,\mathbf{y}_n}\} \tag{8}$$

where $\mathbf{s}_n = \boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)}$, $\mathbf{y}_n = \mathbf{D}(\boldsymbol{\lambda}^{(n+1)}) - \mathbf{D}(\boldsymbol{\lambda}^{(n)})$, $||.||_F$ is the Frobenius norm. The solution of (8) is given by

$$\alpha_{n+1} = \alpha_n + \frac{\mathbf{D}(\boldsymbol{\lambda}^{(n+1)})\mathbf{s}_n^T}{< \mathbf{s}_n, \mathbf{s}_n >} \tag{9}$$

Suppose $\hat{W}$ and W are nonsingular matrices in $\mathbf{L}(R^s)$, then a weighted update can also be obtained as the solution to the minimisation problem [2]

$$\min\{||\hat{W}(\hat{\alpha} - \alpha_n)W||_F : \hat{\alpha} \in Q_{\mathbf{s}_n,\mathbf{y}_n}\} \tag{10}$$

and is given by

$$\alpha_{n+1} = \alpha_n + \frac{\mathbf{D}(\boldsymbol{\lambda}^{(n+1)})\mathbf{v}_n^T}{< \mathbf{v}_n, \mathbf{s}_n >} \tag{11}$$

where $\mathbf{v}_n = W^{-T}W^{-1}\mathbf{s}_n$. We are interested to apply the above techniques, as well as some other extensions as described below, to a linear system

$\mathbf{D}(\boldsymbol{\lambda}) \equiv J\boldsymbol{\lambda} - \mathbf{b} = 0$ where J is an $s \times s$ matrix. For some interface problems, the sparsity structure of the Jacobian matrices is known, therefore one should employ the so-called Schubert's update rather than Broyden's update. The difference between Broyden's method and Schubert's method is that the former does not take care of the sparsity of the Jacobian matrix while the latter preserves the sparseness structure of the Jacobian matrix. Any algorithm listed below with its name begining with Schubert has its sparseness structure of the Jacobian matrix being preserved.

Algorithm 3.1 Modified Newton's Method [5]. Given $\boldsymbol{\lambda}^{(0)}$ and α_0, compute $\boldsymbol{\lambda}^{(1)}$ using (7). Then evaluate $J(\boldsymbol{\lambda}^{(1)})$ by means of a finite difference approximation. Finally use (7) to compute $\boldsymbol{\lambda}^{(n+1)}$ by choosing $\alpha_n = J(\boldsymbol{\lambda}^{(1)})$, $n = 1, 2, \ldots$.

α_0 is chosen as a diagonal matrix or in such a way that its sparseness is the same as that of the Jacobian matrix. Also a constant α_n is chosen because it is expensive to evaluate the Jacobian matrix. The convergence of this method is guaranteed by Theorem 3.1.

Algorithm 3.2 Broyden's Method [2]. Given $\boldsymbol{\lambda}^{(0)}$ and α_0, compute $\boldsymbol{\lambda}^{(n+1)}$ using (7) and α_{n+1} using (9), for $n = 0, 1, \ldots$.

For simplicity α_0 is chosen as a diagonal matrix. However the subsequent generated matrices, α_n, are full matrices. The inverses of these matrices are expensive and there is no guarantee that these matrices are nonsingular. The convergence properties of this method are given by Theorem 3.2.

Algorithm 3.3 Schubert's Method [7]. Perform the same steps as that given in Algorithm 3.2 but the successive updates α_{n+1} are made in such a way that the sparseness structure is preserved.

For simplicity α_0 is chosen as a diagonal matrix. As in the previous case, the sequence of matrices generated by (7) is not necessarily nonsingular. The convergence properties of this method are also given by Theorem 3.2.

In a recent paper published by Kim and Tewarson [4], they proposed a weighted mean of (9) and (11) with $\mathbf{v}_n = -\alpha_n^T \mathbf{D}(\boldsymbol{\lambda}^{(n)})$, i.e.

$$\alpha_{n+1} = \alpha_n + (1-\mu)\frac{\mathbf{D}(\boldsymbol{\lambda}^{(n)})\mathbf{s}_n^T}{<\mathbf{s}_n, \mathbf{s}_n>} + \mu\frac{\mathbf{D}(\boldsymbol{\lambda}^{(n)})\mathbf{v}_n^T}{<\mathbf{v}_n, \mathbf{s}_n>} \tag{12}$$

where μ is chosen to satisfy

$$\|\frac{\mathbf{D}(\boldsymbol{\lambda}^{(n)})\mathbf{s}_n^T}{<\mathbf{s}_n, \mathbf{s}_n>}\|_F = \|\mu\frac{\mathbf{D}(\boldsymbol{\lambda}^{(n)})\mathbf{v}_n^T}{<\mathbf{v}_n, \mathbf{s}_n>}\|_F$$

from which μ is obtained as

$$\mu = \frac{< -\alpha_n^T \mathbf{D}(\boldsymbol{\lambda}^{(n)}), \mathbf{s}_n >^2}{<\mathbf{s}_n, \mathbf{s}_n>< -\alpha_n^T \mathbf{D}(\boldsymbol{\lambda}^{(n)}), -\alpha_n^T \mathbf{D}(\boldsymbol{\lambda}^{(n)}) >} \tag{13}$$

We implemented the weighted update of Kim and Tewarson by means of Schubert's approach, i.e. the sparsity of the Jacobian matrix is preserved.

Algorithm 3.4 Schubert - Kim Method. Given $\boldsymbol{\lambda}^{(0)}$ and α_0, compute $\boldsymbol{\lambda}^{(n+1)}$ using (7) and α_{n+1} using (12), for $n = 0, 1, \ldots$.

It is worth to note that when $\mu = 1$, the method actually fails to converge for some interface problems, in particular the 1-D convection diffusion problem described later. Convergence results shown in [4] do not take care of the sparseness structure of the Jacobian matrix. However a slight modification of the proof will lead to the same result as that given in Theorem 3.2.

All methods described so far do not take care of the nonsingularity of the matrix α_{n+1}. Following the well known determinant property

$$\det(I + \mathbf{u}\mathbf{v}^T) = 1 + < \mathbf{u}, \mathbf{v} >$$

for any $\mathbf{u}$ and $\mathbf{v}$ in R^s, one can deduce that if α_n is nonsingular then, α_{n+1} is nonsingular if and only if $< \mathbf{s}_n, \alpha_n^{-1}\mathbf{y}_n > \neq 0$. Powell defined a modification to Broyden's method given by [8]

$$\alpha_{n+1} = \alpha_n + \theta_n \frac{\mathbf{D}(\boldsymbol{\lambda}^{(n+1)})\mathbf{s}_n^T}{< \mathbf{s}_n, \mathbf{s}_n >} \tag{14}$$

where θ_n is chosen so that α_{n+1} is nonsingular. In other words, we require

$$|\det \alpha_{n+1}| \geq \eta |\det \alpha_n|, \qquad |\theta_n - 1| \leq \eta \tag{15}$$

for any given $\eta \in (0, 1)$. Using the above determinant property, one can easily deduce the following relation

$$\theta_n = \begin{cases} 1, & |\gamma_n| \geq \eta \\ \frac{1 - \text{sign}\,(\gamma_n)\eta}{1 - \gamma_n}, & |\gamma_n| < \eta \end{cases} \tag{16}$$

where $\gamma_n = < \mathbf{s}_n, \alpha_n^{-1}\mathbf{y}_n > / < \mathbf{s}_n, \mathbf{s}_n >$, and $\text{sign}\,(0) = 1$. We implement, in Algorithm 3.5, the above modification by means of Schubert's approach.

Algorithm 3.5 Schubert - Powell Method [7]. Given $\boldsymbol{\lambda}^{(0)}$ and α_0, compute $\boldsymbol{\lambda}^{(n+1)}$ using (7) and α_{n+1} using (14) and (16) for $n = 0, 1, \ldots$.

The convergence properties [7] of this algorithm is described by Theorem 3.2. In addition to the results given by the theorem, the algorithm is also globally convergent [8].

Theorem 3.2 (Lam [6], Marwil [7], Moré and Trangenstein [8]) Suppose $\mathbf{D}$ satifies the assumption given in (6) with $p = 1$, and consider Algorithms 3.2, 3.3, 3.4, 3.5. Then given $r \in (0, 1)$, there exists $\varepsilon = \varepsilon(r)$, $\delta = \delta(r)$ such that if $||\boldsymbol{\lambda}^{(0)} - \boldsymbol{\lambda}^*|| \leq \varepsilon$, and $||\alpha_n - J(\boldsymbol{\lambda}^*)|| \leq \delta$ for all $n = 0, 1, 2, \ldots,$ the sequences $\{\boldsymbol{\lambda}^{(n)}\}$ generated by Algorithms 3.2, 3.3, 3.4, 3.5 are well defined, converge to $\boldsymbol{\lambda}^*$ and satisfy

$$||\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^*|| \leq r||\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^*||$$

for all $n = 0, 1, \ldots$. Moreover, the convergence rate is Q-superlinear, i.e.

$$\lim_{n\to\infty} \frac{||(\alpha_n - J(\boldsymbol{\lambda}^*))(\boldsymbol{\lambda}^{(n+1)} - \boldsymbol{\lambda}^{(n)}||}{||\boldsymbol{\lambda}^{(n+1} - \boldsymbol{\lambda}^{(n)}||} = 0$$

Finally, we describe an algorithm based on a sequence of adaptive parameters [5]. Here the technique for a scalar defect equation is essentially an optimal one-point iteration method where α_n is obtained by setting $\mathbf{G}' = 0$, where $\mathbf{G} = \boldsymbol{\lambda}^{(n)} - \alpha_n^{-1}\mathbf{D}(\boldsymbol{\lambda}^{(n)})$. This adaptive parameter α_n is equivalent to the scalar ϵ-algorithm [5]. An adaptive α for the extension [5] to s-dimensional problems is

$$\alpha_{n+1} = \alpha_n \frac{\|\mathbf{D}(\boldsymbol{\lambda}^{(n+1)}) - \mathbf{D}(\boldsymbol{\lambda}^{(n)})\|}{\|\mathbf{D}(\boldsymbol{\lambda}^{(n)})\|} \tag{17}$$

Algorithm 3.6 Adaptive α [5]. Given $\boldsymbol{\lambda}^{(0)}$ and α_0, compute $\boldsymbol{\lambda}^{(n+1)}$ using (7) and α_{n+1} using (17), for $n = 0, 1, \ldots$.

Since the adaptive α technique is equivalent to choosing a diagonal matrix with constant diagonal elements, the proof of convergence proceeds in the same way as that given in Ref. [6]

4. Parallel Performance Monitor

Suppose a finite difference scheme is applied to each of the subdomains, then each subproblem solver involves the solution of a sparse system of lower dimension compared with the system of the entire problem with the same sparsity structure. Nevertheless the number of iterations, n_{it}, obtained by using an exact subproblem solver is similar to that obtained by using a discretised subproblem solver [5].

Let M denote the total number of nodes in the entire computational domain. One work unit is defined as the computational work required to solve the sparse system with $M - 2$ unknowns. For 1-D steady problems, solving the entire computational problem requires one work unit. For 1-D time dependent problems, solving the entire computational problem requires n_t work units where n_t is the number of time steps. Let the entire computational domain be divided into $s + 1$ subdomains, M_k be the number of nodes in the k-th subdomain, $k = 1, 2, \cdots, s + 1$ and n_{it} be the number of updates in order to obtain a converged solution $\boldsymbol{\lambda}^{(n_{it})}$ along the interfaces by using a quasi-Newton scheme. Suppose there is a set of $s + 1$ concurrent processors and that the connectivity is the same as the layout of the subdomains. Since most of the computational work is devoted to the solutions of subproblems, then it is possible to estimate the parallel computing time by taking the sum of the maximum work unit in any iteration involved in the subproblem solutions, i.e.

$$\tau = \sum_{n=1}^{n_{it}} \max_{1\leq k\leq s+1} \{\frac{M_k - 2}{M - 2}\} \tag{18}$$

For 1-D steady problem, a quasi-Newton algorithm described previously is considered as an efficient parallel algorithm provided $\tau < 1$. Assuming that $n_{it} = C(s+1)^q$, $q \geq 0$, $C > 0$ and that the interfaces are evenly distributed, then

$$\tau = \frac{C(M-2-s)(s+1)^q}{(s+1)(M-2)} \tag{19}$$

Proposition 4.1 Suppose the number of iterations, n_{it}, of applying a quasi-Newton iteration scheme to an iterface problem for a steady 1-D problem is related to the number of subdomains, $s+1$, by the expression $n_{it} = C(s+1)^q$, where $C > 0$, $q \geq 0$ and the interfaces are evenly distributed, we have

1. For $0 \leq q < 1$, if $s+1 > C^{1/1-q}$ then $\tau < 1$.
2. For $q = 1$, $\tau < 1$ if $C < 1$.
3. For $q > 1$, if $s+1 \not> C^{1/1-q}$ then $\tau < 1$.

Proof : For the first case, we require $C(M-2-s)/(s+1)^{1-q}(M-2) < 1$, i.e. $(C-(s+1)^{1-q})(M-2) - Cs < 0$. If $s+1 > C^{1/1-q}$ then $C-(s+1)^{1-q} < 0$, and hence the result follows. For the second case, we require $C(M-2-s)/(M-2) < 1$, i.e. $(C-1)(M-2)-Cs < 0$ which is true if $C < 1$. For the third case, we require $C(M-2-s)(s+1)^{q-1}/(M-2) < 1$, i.e. $(C(s+1)^{q-1}-1)(M-2) - Cs(s+1)^{q-1} < 0$. If $s+1 \not> C^{1/1-q}$ then $C(s+1)^{q-1} < 1$, and hence the result follows.

5. Numerical Examples

We consider the following convection dominant flow problem,

$$\frac{d^2\phi}{dx^2} - \gamma\frac{d\phi}{dx} = 0, \qquad \phi(0) = 0, \phi(1) = 1 \tag{20}$$

where $\gamma \gg 1$. The domain is subdivided into $s+1$ subdomains with interfaces, Γ_k, $k = 1, 2, \ldots, s$, distributed evenly across the domain. We use exact solutions in each of the subdomains and are interested to compare the number of iterations, n_{it}, required to update the function values along the interfaces. The stopping criterion is $\|\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^*\| < 0.5 \times 10^{-5}$. The results are presented in Tables 1 and 2 for $\gamma = 40$ and 50 respectively. Note that an "x" represents divergence of the test. We also compute and present in Table 3 the values of C and q for the various algorithms applied to the present problem.

Obviously, Algorithm 3.1 is very attractive because $q = 0$. From Proposition 4.1, we can choose $s+1 > 2$ in order to provide an efficient parallel method. Note that the construction of $J(\boldsymbol{\lambda}^{(1)})$ requires $2s$ subproblem solvers. Whereas the subproblem solvers involved in the construction of the Jacobian matrix can be performed in two parallel parts, the same situation does not repeat in 2-D case. Therefore Algorithm 3.1 is only recommended

Algorithm	$s+1$				
	4	8	16	32	64
3.1	2	2	2	2	2
3.2	6	14	29	50	112
3.3	9	14	21	21	39
3.4	11	15	18	24	46
3.5	10	16	19	26	38
3.6	13	19	35	59	143

Table 1: n_{it} for $\gamma = 40$.

Algorithm	$s+1$				
	4	8	16	32	64
3.1	2	2	2	2	2
3.2	6	14	25	48	88
3.3	7	13	19	21	36
3.4	7	13	18	24	x
3.5	9	14	16	22	36
3.6	10	19	31	51	102

Table 2: n_{it} for $\gamma = 50$.

as a parallel shooting method for 1-D problems and should not be employed in the 2-D case [5].

For Algorithm 3.2, we have $C = 1.57$ and $q = 1.03$ which gives $C^{1/1-q} = 3 \times 10^{-7}$ and it is impossible to have the number of subdomains less than this value. Hence Algorithm 3.2 is not suitable for the present applications.

For Algorithms 3.3, 3.4, and 3.5, the minimum numbers of subdomains are estimated as 21, 33, and 24 respectively. However for Algorithm 3.6, the minimum number of subdomains is 5×10^9 which is unrealistic and hence is also not suitable for parallel implementation.

Algorithm	3.1	3.2	3.3	3.4	3.5	3.6
C	2	1.57	4.02	3.14	4.38	2.45
q	0	1.03	0.54	0.67	0.53	0.96

Table 3: Values of C and q.

6. Conclusion

We examine a number of quasi-Newton methods for the solutions of interface problems. We establish a criterion for the minimum number of subdomains to be employed in order to assess whether an algorithm can be efficiently implemented on coarse-grained parallel computers of either shared or distributed memory. Extension to 2-D problems and theoretical estimations of C and q are currently being studied.

Acknowledgements

The author wishes to thank Dr. H.J.J. te Riele for his suggestion on this work and the members of staff from NW3 for their support during the stay of the author at CWI.

References

[1] Dennis, Jr. J.E. and Moré, J.J., *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math Comp, 28, 549 - 560, 1974

[2] Dennis, Jr. J.E. and Schnabel, R.B., *Least change secant updates for quasi-newton methods*, SIAM Review, 21, 443 - 459, 1979

[3] *Domain Decomposition Methods IV*, R. Glowinski, et. al., eds., SIAM Philadelphia, 1991

[4] Kim, S. and Tewarson, R.P., *A quasi-Newton method for solving nonlinear algebraic equations*, Computers Math Applic, 24, 93 - 97, 1992

[5] Lai, C.-H., *An iteration scheme for non-symmetric interface operator*, ERCIM Research Report, ERCIM-92-R003, 1992

[6] Lam, B., *On the convergence of a quasi-Newton method for sparse nonlinear systems*, Math Comp, 32, 447 - 451, 1978

[7] Marwil, E., *Convergence results for Schubert's method for solving sparse nonlinear equations*, SIAM J Num Anal, 16, 588 - 604, 1979

[8] Moré, J.J. and Trangenstein, J.A., *On the global convergence of Broyden's method*, Math Comp, 30, 523 - 540, 1976