REVUE D'ANALYSE NUMÉRIQUE ET DE THÉORIE DE L'APPROXIMATION
Rev. Anal. Numér. Théor. Approx., vol. 37 (2008) no. 2, pp. 169–180
ictp.acad.ro/jnaat

EXTENDING BROYDEN'S METHOD TO INTERACTION PROBLEMS

ROBBY HAELTERMAN[*], JORIS DEGROOTE[†], JAN VIERENDEELS [†]
and DIRK VAN HEULE[*]

Abstract. The solution of problems involving the interaction of different systems is a domain of ongoing research, although often a good solver already exists for each system separately. In this paper we draw our ideas from one of the best known all-round quasi-Newton methods: Broyden's rank-one update, which we extend to algorithms using 2 approximate Jacobians. A comparison is made with the iterative substructuring method and Aitken's acceleration method. It is shown that a Broyden method using only a single approximate Jacobian performs best.

MSC 2000. 90C53, 15A06, 65F10, 74F10.

Keywords. Quasi-Newton method, iterative method.

1. INTRODUCTION

Simulations of coupled systems are becoming ever more important. While powerful solvers often exist for problems in a single physical domain (e.g. structural problems, flow problems, ...), development of similar tools for multiphysics problems is still ongoing, and these can be broadly put in one of the following categories:

- *Monolithic or simultaneous solution*: the whole problem is treated as a monolithic entity and solved with a specialized solver.
- *Partitioned solution*: the physical components are treated as isolated entities that are solved separately.

The relative merits of these methods are very problem dependent: the advantage of the monolithic approach is the enhanced stability. This comes at a cost, however, as specialized software has to be written for each type of interaction problem, which can result in very large systems. The partitioned approach allows for the use of available specialized solvers for each physical component (structure, fluid,...), on condition that the coupling effects can be treated efficiently. The latter is often no problem for loosely coupled problems,

[*] Department of Mathematics, Royal Military Academy, Renaissancelaan 30, B-1000 Brussels, Belgium, e-mail: Robby.Haelterman@rma.ac.be.

[†] Department of Flow, Heat and Combustion, Ghent University, St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium, e-mail: Joris.Degroote@UGent.be.

which can be solved by sequentially calling the solvers for each component. Strongly coupled problems, on the other hand, still pose a real challenge; the simplest solution method is the *iterative substructuring method*, but this often displays very poor stability [4].

In this paper we write the coupled problem as a root-finding problem which we solve with Broyden's quasi-Newton method and an extension of this method in which two approximate Jacobians are used, one for each physical problem. This results in a partitioned approach in which each of the physical problems is solved with a specialized code that we consider to be a black box and of which the Jacobian is unknown. We assume the dominant cost lies in the use of these black boxes.

This paper is organized as follows: in section 2 we describe the type of problems we are interested in; in section 3, resp. 4, we describe two well-known methods for the solution of these problems: the iterative substructuring method, resp. Aitken's Δ^2 method; in section 5 we briefly describe Broyden's quasi-Newton method together with two proposed variations; in section 6 we test the various algorithms on two coupled problems.

2. THE GENERAL MODEL

We consider coupled problems that can be stated in the following form:

$$F(g) = p, \tag{1a}$$

$$S(p) = g, \tag{1b}$$

where $F : D_F \subset \mathbb{R}^{M\times 1} \to \mathbb{R}^{N\times 1} : g \mapsto F(g)$ and $S : D_S \subset \mathbb{R}^{N\times 1} \to \mathbb{R}^{M\times 1} : p \mapsto S(p)$. Each equation describes a physical problem that is spatially decomposed. We limit p and g to values on the interface between the two physical problems. In this way the physically decoupled nature of the problem is exploited, and the method can be regarded as a special case of *heterogeneous domain decomposition methods* [2].

We want to solve (1) with a partitioned method that keeps the available solvers for each individual equation (1a), resp. (1b), of which we lack knowledge of the Jacobian. (1) can also be written as the root-finding problem

$$F(S(p)) - p = H(p) - p = K(p) = 0. \tag{2}$$

Depending on the algorithm we will either use the formulation of (1) or (2). We assume that F, S, H and K satisfy the following properties:

(1) F, resp S, H, is twice continuously differentiable in an open set D_F, resp D_S, D_H.
(2) There is a p_o such that the set $\mathcal{C}(p_o) = \{p; \|K(p)\| \leq \|K(p_o)\|\}$ is compact.
(3) $K(p) = 0$ has a single solution p^* in $\mathcal{C}(p_o)$
(4) $(K'(p))^{-1}$ exists and is continuous in an open set containing $\mathcal{C}(p_o)$.

REMARK. The choice, in equation (2), between $F(S(p)) = 0$ or $S(F(g)) = 0$, will depend on the relative dimensions of p and g; we will choose the formulation such that the (square) Jacobian of K is of the lowest dimension. □

3. ITERATIVE SUBSTRUCTURING METHOD

The simplest strongly coupled method to solve (1) is the iterative substructuring method; it proceeds as follows:

```
ALGORITHM 3.1 (Iterative substructuring method (ISM)).
1. Startup. Take an initial value $p_o$.
   Set $s = 0$.
2. Loop until sufficiently converged
   a. $g_s = S(p_s)$
   b. $p_{s+1} = F(g_s)$
   c. $s = s + 1$
```

Depending on the strength of the coupling between the two equations in (1) this method is not always stable [4].

4. AITKEN'S Δ^2 METHOD

Another method that is often used for the solution of (2) is Aitken's Δ^2 method [3].

```
ALGORITHM 4.1 (Aitken's $\Delta^2$ method (A$\Delta^2$)).
1. Startup:
   Take a starting value $p_o$ and $\theta_o$;
   compute $K(p_o) = H(p_o) - p_o$;
   compute $p_1 = p_o + \theta_o K(p_o)$;
   set $s = 1$.
2. Loop until sufficiently converged:
   a. Compute $K(p_s) = H(p_s) - p_s$.
   b. $\theta_s = -\theta_{s-1} \frac{K(p_{s-1})^T \delta K_{s-1}}{\langle \delta K_{s-1}, \delta K_{s-1} \rangle}$
   c. $p_{s+1} = p_s + \theta_s K(p_s)$
   d. $s = s + 1$
```

($\delta K_{s-1} = K(p_s) - K(p_{s-1})$ and $\langle \cdot, \cdot \rangle$ is the classical scalar product.)

5. BROYDEN'S METHOD AND VARIATIONS

A well-known solution method for the nonlinear problem in (2) is Newton's method. However, as we suppose we don't have access to the Jacobians of either F, S, H or K, we will resort to quasi-Newton methods, which we apply either to the single equation (2) or the system (1).

5.1. Basic Broyden method. Broyden's method [1] was developed for a single equation as in (2). The resulting algorithm can be written as follows.

```
ALGORITHM 5.1 (Broyden's algorithm (Br)).
1. Startup:
   Take a starting value $p_o$ and $\hat{K}'_o$;
   compute $K(p_o) = H(p_o) - p_o$;
   compute $p_1 = (1-\omega)p_o + \omega H(p_o)$ ($\omega$ being a relaxation parameter);
   set $s = 1$.
2. Loop until sufficiently converged:
   a. Compute $K(p_s) = H(p_s) - p_s$.
   b. Compute the Broyden update $\hat{K}'_s = \hat{K}'_{s-1} + \frac{(\delta K_{s-1} - \hat{K}'_{s-1}\delta p_{s-1})\delta p^T_{s-1}}{\langle \delta p_{s-1}, \delta p_{s-1} \rangle}$
   c. $p_{s+1} = p_s - (\hat{K}'_s)^{-1} K(p_s)$
   d. $s = s + 1$
```

($\delta p_{s-1} = p_s - p_{s-1}$.)

A good initial guess for $\hat{K}'_o$ is needed. As we do not have any idea what the Jacobian of F or S is, we will set them equal to zero, such that we have $\hat{K}'_o = -I$ as a consequence.

5.2. Extension of Broyden's method. We recall that $K(p) = F(S(p)) - p$ and that as such

$$K'(p_s) \quad = \quad F'(S(p_s))S'(p_s) - I.$$

We therefore propose to build approximate Jacobians $\hat{F}'_s$ and $\hat{S}'_s$ along the lines of the Broyden update:

$$\hat{F}'_s = \hat{F}'_{s-1} + \frac{(\delta F_{s-1} - \hat{F}'_{s-1}\delta g_{s-1})\delta g^T_{s-1}}{\langle \delta g_{s-1}, \delta g_{s-1} \rangle}, \tag{3a}$$

$$\hat{S}'_s = \hat{S}'_{s-1} + \frac{(\delta S_{s-1} - \hat{S}'_{s-1}\delta p_{s-1})\delta p^T_{s-1}}{\langle \delta p_{s-1}, \delta p_{s-1} \rangle}, \tag{3b}$$

where $\delta g_{s-1} = g_s - g_{s-1}$, $\delta F_{s-1} = F(g_s) - F(g_{s-1})$, and $\delta S_{s-1} = S(p_s) - S(p_{s-1})$; we set $\hat{K}'_s = \hat{F}'_s \hat{S}'_s - I$.

To keep in line with $\hat{K}'_o$ in §5.1 we set $\hat{F}'_o = \hat{S}'_o = 0$.

It is easy to see that this construction respects the following secant equations:[1]

$$\hat{F}'_s \delta g_{s-1} \quad = \quad \delta F_{s-1}, \tag{4a}$$

$$\hat{S}'_s \delta p_{s-1} \quad = \quad \delta S_{s-1}, \tag{4b}$$

$$\hat{K}'_s \delta p_{s-1} \quad = \quad \delta K_{s-1}. \tag{4c}$$

[1] Broyden's original update also respects the secant equation.

This now allows us to write the following algorithm:

```
ALGORITHM 5.2 (Broyden's method–2 Jacobians–Variant 1 (BrVar1)).

1. Startup:
   Take a starting value $p_o$, $\hat{F}'_o$ and $\hat{S}'_o$;
   compute $g_o = S(p_o)$ and $p_1 = (1-\omega)p_o + \omega F(g_o)$($\omega$ being a relaxation parameter);
   set $s = 1$.
2. Loop until sufficiently converged
   a. Compute $g_s = S(p_s)$.
   b. Make approximate Jacobian $\hat{S}'_s$ according to (3b).
   c. Compute $H(p_s) = F(g_s)$.
   d. Make approximate Jacobian $\hat{F}'_s$ according to (3a).
   e. Quasi-Newton step: $p_{s+1} = p_s - (\hat{F}'_s \hat{S}'_s - I)^{-1}(H(p_s) - p_s)$
   f. $s = s + 1$
```

REMARK. While Broyden's original update also respects (4c), it can be shown that the approximate Jacobians differ between algorithms 5.1 and 5.2. □

In [5] another approach was proposed to couple (1) with two approximate Jacobians, which we now apply to Broyden's method. In this approach we look at each equation in (1) separately and write a Taylor expansion in which we insert the approximate Jacobians:

$$\begin{aligned} F(g_{s+1}) - F(g_s) &= \hat{F}'_s(g_{s+1} - g_s), \\ S(p_{s+1}) - S(p_s) &= \hat{S}'_s(p_{s+1} - p_s). \end{aligned}$$

We require the solution of the coupled problem to satisfy (1) at iteration $s+1$; it follows that

$$p - \hat{F}'_s g = F(g_s) - \hat{F}'_s g_s, \tag{5a}$$

$$g - \hat{S}'_s p = S(p_s) - \hat{S}'_s p_s. \tag{5b}$$

Solving (5) for g and p gives

$$p = (I - \hat{F}'_s \hat{S}'_s)^{-1} \Big(F(g_s) + \hat{F}'_s (S(p_s) - \hat{S}'_s p_s - g_s) \Big), \tag{6a}$$

$$g = (I - \hat{S}'_s \hat{F}'_s)^{-1} \Big(S(p_s) + \hat{S}'_s (F(g_s) - \hat{F}'_s g_s - p_s) \Big). \tag{6b}$$

We assign the value of p in equation (6a) to p_{s+1}, and use it as an input for S,

$$p_{s+1} = (I - \hat{F}'_s \hat{S}'_s)^{-1} \Big(F(g_s) + \hat{F}'_s (S(p_s) - \hat{S}'_s p_s - g_s) \Big).$$

We can then update the Jacobian ($\hat{S}'$) in equation (6b) and assign the value of g to g_{s+1}:

$$g_{s+1} \;=\; (I - \hat{S}'_{s+1}\hat{F}'_s)^{-1}\left(S(p_{s+1}) + \hat{S}'_{s+1}(F(g_s) - \hat{F}'_s g_s - p_{s+1})\right).$$

Combining this with Broyden's method results in the following algorithm:

```
ALGORITHM 5.3 (Broyden's method-2 Jacobians-Variant 2 (BrVar2)).

1. Startup:
   Take a starting value $p_o$, $\hat{F}'_o$ and $\hat{S}'_o$;
   compute $g_o = S(p_o)$ and $p_1 = (1-\omega)p_o + \omega F(g_o)$ ($\omega$ being a relaxation parameter);
   compute $g_1 = S(p_1)$
2. Make approximate Jacobian $\hat{S}'_1$ according to (3b);
   set $s = 1$.
3. Loop until sufficiently converged
   a. Compute $F(g_s)$.
   b. Make approximate Jacobian $\hat{F}'_s$ according to (3a).
   c. $p_{s+1} = (I - \hat{F}'_s\hat{S}'_s)^{-1}\left(F(g_s) + \hat{F}'_s(S(p_s) - \hat{S}'_s p_s - g_s)\right)$
   d. Compute $S(p_{s+1})$.
   e. Make approximate Jacobian $\hat{S}'_{s+1}$ according to (3b).
   f. $g_{s+1} = (I - \hat{S}'_{s+1}\hat{F}'_s)^{-1}\left(S(p_{s+1}) + \hat{S}'_{s+1}(F(g_s) - \hat{F}'_s g_s - p_{s+1})\right)$
   g. $s = s + 1$
```

REMARK. It is obvious from the way we construct $\hat{F}'_s$ and $\hat{S}'_s$ and equations (5a) and (5b) that they respect the secant equations (4a) and (4b). For reasons of comparison with the other algorithms, we start from $\hat{F}'_o = \hat{S}'_o = 0$. □

6. TEST-CASES

We propose two test-cases: the heat equation and a fluid-structure interaction problem. The convergence requirement will be defined based on a relative reduction of the residual, which is defined by $\frac{\|K(p_s)\|}{\|K(p_o)\|} \leq 10^{-5}$ for algorithms 4.1, 5.1 and 5.2, and on $\frac{\|F(g_s)-p_s\|}{\|F(g_o)-p_o\|} \leq 10^{-5}$ for algorithms 3.1 and 5.3, as $K(p_s)$ is not explicitly computed in these. The performance parameter is the number of times the function F (or S) is called, as we have assumed this is the dominant cost of the algorithm; we call this value $\mathcal{FC}$. We break off the algorithm if $\mathcal{FC} > 100$ if it has not converged at that point.

6.1. Heat equation.

6.1.1. *The model.* The heat equation in one dimension without heat source is given by:

$$\rho C \frac{\partial T}{\partial t} - \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) = 0, \tag{7}$$

where T is temperature, ρ is density, C is heat capacity, k is thermal conductivity. We assume we only have a linear solver to solve (7) with an imposed value of ρ, C and k that may vary in space.

If we want to take dependency of ρ, C and k on T into account, we need to do this outside the solver for (7). They can be obtained by the following interpolation polynomials (based on values between 0 and 300°C in [6]):

$$k = 7.0277 \cdot 10^{-5} T + 2.4388 \cdot 10^{-2}, \tag{8a}$$

$$C = 4.3004 \cdot 10^{-7} T^2 + 1.1850 \cdot 10^{-5} T + 1.0048, \tag{8b}$$

$$\rho = 5.3641 \cdot 10^{-6} T^2 - 3.7809 \cdot 10^{-3} T + 1.2781. \tag{8c}$$

We use a finite differencing scheme on N equidistant nodes for the solver of (7), with spacing Δx, and an implicit time discretization with time step Δt. This gives the N linear equations ($i = 1, \ldots, N$):

$$\begin{aligned} \rho_i C_i (T_i^{n+1} - T_i^n) - \frac{\Delta t}{2\Delta x^2}\Big((k_{i+1} + k_i) T_{i+1}^{n+1} - \\ -(k_{i+1} + 2k_i + k_{i-1}) T_i^{n+1} + (k_i + k_{i-1}) T_{i-1}^{n+1}\Big) = 0. \end{aligned} \tag{9}$$

We will use g to denote the ensemble of discretized constants ρ_i, C_i and k_i, which will be defined by a node-to-node relation with T_i according to (8).

We only consider one time-step and solve (9) for the temperature at the next time-step. We write this as $F(g_s) = T_{s+1}$. The parameters are computed for a given temperature, which is written as $S(T_s) = g_s$.

As an initial condition we take the values at time-level n, which is a uniform temperature of $T^o = 150K$, the right boundary condition is a temperature of $150K$, while the left boundary condition is a function of time, defined as $T_L = T^o + \frac{T^o}{2} \sin\left(\frac{\pi t}{10}\right)$.

We use an initial relaxation factor $\omega = 0.1$.

6.1.2. *Results and discussion.* Various combinations of the ratio $\frac{\Delta t}{\Delta x^2}$ were tried for $N = 100$ and $N = 1000$ (tables 1 and 2). For small values of $\frac{\Delta t}{\Delta x^2}$ the three Broyden algorithms and the ISM performed equally well; Aitken's method trailed behind. As the ratio $\frac{\Delta t}{\Delta x^2}$ grows, the problem becomes harder to solve and the number of iterations becomes larger. The performance of the ISM initially degrades faster than the Broyden methods, which perform equally well. For $N = 1000$ and $\frac{\Delta t}{\Delta x^2} = 10^9$ the ISM method catches up with the basic Broyden method, while the other three methods lag behind. For even higher values of $\frac{\Delta t}{\Delta x^2}$ convergence could not be obtained. It is thus seen that for this mildly non-linear problem there is not much difference between the methods, but the basic Broyden method is always the best. This is perhaps surprising as it only uses one Jacobian and thus less information than the Broyden variants.

Table 1. $\mathcal{FC}$ required for convergence for the heat equation. $N = 100$.

$\frac{\Delta t}{\Delta x^2}$	Br	BrVar1	BrVar2	ISM	AΔ^2
10^{-10}	3	3	3	3	4
10^{-9}	3	3	3	3	4
10^{-8}	3	3	3	3	4
10^{-7}	3	3	3	3	5
10^{-6}	3	3	3	4	5
10^{-5}	3	3	3	4	5
10^{-4}	3	3	3	4	5
10^{-3}	3	3	3	4	5
10^{-2}	4	4	4	5	6
10^{-1}	4	4	4	5	6
1	5	5	5	6	7
10	6	6	6	8	8
10^{2}	6	6	6	8	8
10^{3}	6	6	6	8	9
10^{4}	6	6	6	8	9
10^{5}	6	6	6	7	9
10^{6}	6	6	6	7	9
10^{7}	6	6	6	7	9
10^{8}	6	6	6	7	9
10^{9}	6	6	6	7	9
10^{10}	7	7	7	7	9

6.2. Fluid-structure interaction problem. In this problem both physical problems are vector-based and non-linear.

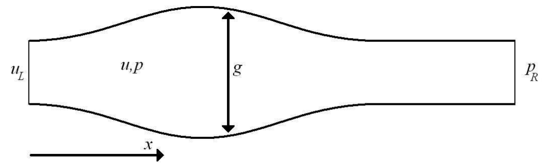

Fig. 1. One-dimensional flow in a flexible tube.

6.2.1. *The model.* We consider one-dimensional unsteady flow in an elastic tube (figure 1). (For a two-dimensional simulation, see [5]). The fluid is incompressible and inviscid and gravity is neglected. The governing equations are the conservation of mass and momentum which can be written in conservative

Table 2. $\mathcal{FC}$ required for convergence for the heat equation. $N = 1000$. "div" = divergence or non-convergence: $\mathcal{FC} > 100$.

$\frac{\Delta t}{\Delta x^2}$	Br	BrVar1	BrVar2	ISM	AΔ^2
10^{-10}	3	3	3	3	4
10^{-9}	3	3	3	3	4
10^{-8}	3	3	3	3	4
10^{-7}	3	3	3	3	5
10^{-6}	3	3	3	4	5
10^{-5}	3	3	3	4	5
10^{-4}	3	3	3	4	5
10^{-3}	3	3	3	4	5
10^{-2}	4	4	4	5	6
10^{-1}	4	4	4	5	6
1	5	5	5	6	7
10	6	6	6	8	8
10^{2}	6	6	6	8	8
10^{3}	6	6	6	8	9
10^{4}	6	6	6	8	9
10^{5}	6	6	6	8	9
10^{6}	6	6	6	8	9
10^{7}	6	6	6	7	9
10^{8}	6	6	6	7	9
10^{9}	8	9	9	8	9
10^{10}	div	div	div	div	div

form as

$$\frac{\partial g}{\partial t} + \frac{\partial (gu)}{\partial x} = 0, \tag{10a}$$

$$\frac{\partial (gu)}{\partial t} + \frac{\partial (gu^2)}{\partial x} + \frac{1}{\rho}\left(\frac{\partial (gp')}{\partial x} - p'\frac{\partial g}{\partial x}\right) = 0 \tag{10b}$$

to which suitable boundary conditions need to be added (velocity at the inlet and pressure at the outlet). g represents the cross-sectional area of the tube, u the velocity along the axis of the tube, ρ is the density of the fluid and p' the static pressure.

We introduce the kinematic pressure $p = p'/\rho$. If the elastic wall of the tube has a constitutive law of the form $g = g(p(x,t))$, with its inertia neglected, then (10) can be rewritten in the following form:

$$\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} + c^2\frac{\partial u}{\partial x} = 0, \tag{11a}$$

$$\frac{\partial gu}{\partial t} + \frac{\partial gu^2}{\partial x} + \frac{\partial gp}{\partial x} - p\frac{\partial g}{\partial x} = 0. \tag{11b}$$

The wave speed c is defined by

$$c^2 = \frac{g}{\frac{\partial g}{\partial p}}.$$

For the constitutive law of the structure we take the non-linear relation defined by:

$$g = g_o \left(\frac{p - 2c_{mk}^2}{p_o - 2c_{o,mk}^2} \right)^2 ,$$

where c_{mk} is the Moens-Korteweg wave speed $\left(c_{mk} = \sqrt{\frac{Eh}{2\rho r_o}} \right)$, E Young's modulus, h the thickness of the tube wall and r_o a transversal reference length of the tube. The length of the tube is L.

The velocity is imposed at the inlet as

$$u = u_o + \frac{u_o}{10} \sin^2 \left(\frac{\pi t}{T} \right) ,$$

where u_o is the initial velocity and T the period ($= 1$). At the outlet, a non-reflecting boundary condition is imposed:

$$\frac{\partial u}{\partial t}\Big|_{x=L} = \frac{1}{c} \frac{\partial p}{\partial t}\Big|_{x=L} .$$

We use a one-dimensional mesh with N nodes of length Δx. Fluid velocity, pressure and diameter of the tube are stored at the cell centers. Central discretization is used for all terms in (11), except for the convective term in (11b) where a first-order upwind scheme is used; pressure stabilization is added [5]. Pressure and velocity at the inlet, resp. outlet are linearly interpolated from nearby mesh-points. For the time-discretization we use backward Euler with time-step Δt.

As in §6.1, we only consider one time-step and use those values as initial values for the iteration. We solve the discretized fluid problem for the pressure at the next time-step and a fixed geometry, which we write as $F(g_s) = p_{s+1}$. For the structural problem we compute g for a given pressure, which we write as $S(p_s) = g_s$.

The initial relaxation parameter is $\omega = 10^{-2}$, except for $(\kappa, \tau) = (10, 10^{-2})$ and $(10, 10^{-3})$, where it is 10^{-4}, resp. 10^{-5}.

6.2.2. *Results and discussion.* Various combinations of parameters were tried for $N = 100$ and $N = 1000$. When the governing equations are non-dimensionalized it is seen that they are characterized by two non-dimensional parameters $\kappa = \frac{c_o}{u_o}$, being a dimensionless structural stiffness, and $\tau = \frac{u_o \Delta t}{L}$, being a dimensionless time-step; c_o, u_o, are the linearizing, i.e. reference, values of the wave speed, resp. fluid velocity. Various combinations of these were tested (tables 3 and 4). It is seen that a more flexible tube (lower value of κ) and a smaller time-step (lower value of τ) represent a more difficult problem to solve. The three Broyden methods steadily outperform the ISM and Aitken's method but share similar performance for the "easier" test-cases. When the problem becomes harder, the basic Broyden method appears the best, followed by the second variant of Broyden's method. Again this is somewhat surprising as all the information of the two systems S and F is lumped into a single approximate Jacobian $\hat{K}_s'$.

Table 3. $\mathcal{FC}$ required for convergence for the FSI problem. $N = 100$. "div" = divergence or non-convergence: $\mathcal{FC} > 100$.

κ	τ	Br	BrVar1	BrVar2	ISM	AΔ^2
1000	10^{-1}	3	3	3	5	4
1000	10^{-2}	3	3	3	7	5
1000	10^{-3}	5	5	5	14	8
100	10^{-1}	4	4	4	7	5
100	10^{-2}	6	6	6	17	10
100	10^{-3}	9	9	9	div	16
10	10^{-1}	6	6	6	28	11
10	10^{-2}	9	9	9	div	18
10	10^{-3}	37	38	37	div	div

Table 4. $\mathcal{FC}$ required for convergence for the FSI problem. $N = 1000$. "div" = divergence or non-convergence: $\mathcal{FC} > 100$.

κ	τ	Br	BrVar1	BrVar2	ISM	AΔ^2
1000	10^{-1}	3	3	3	5	4
1000	10^{-2}	3	3	3	7	5
1000	10^{-3}	6	6	6	17	10
100	10^{-1}	4	4	4	7	5
100	10^{-2}	6	6	6	28	14
100	10^{-3}	9	9	9	div	11
10	10^{-1}	6	6	6	34	11
10	10^{-2}	13	14	13	div	27
10	10^{-3}	36	46	40	div	div

7. CONCLUSION

The basic Broyden method was tested on interaction problems and compared with two proposed variants as well as the iterative substructuring method and the Aitken Δ^2 method. For weakly non-linear problems and/or problems requiring few iterations to converge there was little to chose from between the three Broyden methods, although they were better than the remaining two. For more difficult problems not only did the three Broyden-based methods outperform the other two, but it appeared that the basic Broyden method came out best. This is surprising as, compared with the two variants, it only uses a single Jacobian to capture all the information that is available of the two constituent systems, while the other two use two approximate Jacobians, one for each system.

Acknowledgement. The first author kindly acknowledges the financial support from the "T. Popoviciu" Institute of Numerical Analysis (Romanian Academy), for attending the conference.

REFERENCES

[1] Broyden, C.G., *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., **19**, pp. 577–593, 1965.

[2] Deparis, S., Discacciati, M., Fourestey, G. and Quarteroni, A., *Fluid-structure algorithms based on Steklov-Poincaré operators*, Computer Methods in Applied Mechanics and Engineering, **195/41-43**, pp. 5797–5812, 2006.

[3] Küttler, U. and Wall, W., *Fixed-point fluid-structure interaction solvers with dynamic relaxation*, Computational Mechanics 2008, DOI 10.1007/s00466-008-0255-5, to appear.

[4] Michler, C., Van Brummelen, E.H., de Borst, R., *Error-amplification analysis of subiteration-preconditioned GMRES for fluid-structure interaction*, Comput. Methods Appl. Mech. Engrg., **195**, pp. 2124–2148, 2006.

[5] Vierendeels, J., Lanoye, L., Degroote, J. and Verdonck, P., *Implicit coupling of partitioned fluid-structure interaction problems with reduced order models*, Comput. & Structures, **85**, pp. 970–976, 2007.

[6] `http://www.engineeringtoolbox.com/air-properties-d_156.html`, April 2008.

Received by the editors: May 21, 2008.