INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **58**:463–480 (DOI: 10.1002/nme.751)

Improved multi-level Newton solvers for fully coupled multi-physics problems

J. Y. Kim[1], N. R. Aluru[2] and D. A. Tortorelli[1,3,*,†]

[1]*Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.*
[2]*Beckman Institute and Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.*
[3]*Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.*

SUMMARY

An algorithm is suggested to improve the efficiency of the multi-level Newton method that is used to solve multi-physics problems. It accounts for full coupling between the subsystems by using the direct differentiation method rather than error prone finite difference calculations and retains the advantage of greater flexibility over the tightly coupled approaches. Performance of the algorithm is demonstrated by solving a fluid–structure interaction problem. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: multi-physics analysis; fluid–structure interaction; multi-level Newton method

1. INTRODUCTION

Numerical techniques employed for coupled multi-physical domain simulations that are encountered in, for example, electromechanical systems, thermomechanical systems and fluid–structure systems have been based on relaxation [1, 2], the strongly coupled Newton method [3–5] and the relatively recently introduced multi-level Newton method [6, 7].

The relaxation approach neglects inter-domain coupling in the solution procedure so its implementation is straight-forward as it just uses the existing system-specific analysis tools without modification. However it fails to converge when the coupling between disciplines[‡] is significant,[§] therefore its use is restricted [6, 8].

*Correspondence to: D. A. Tortorelli, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.
†E-mail: dtort@acm6.me.uiuc.edu

‡In this work each subsystem corresponds to a specific engineering model that is not further partitioned for parallel analysis. That is, each subsystem corresponds to one domain that is modelled by one physical discipline. As such the terms subsystem and discipline are used interchangeably.

§It is noted that Haftka *et al.* [8] classify a system as strongly coupled if the largest eigenvalue of $(D\mathscr{R} - \mathbf{I})$ is of order 1 or greater.

Received 14 December 2001
Revised 5 August 2002
Accepted 18 November 2002

The strongly coupled Newton approach, on the other hand, incorporates all of the coupling between all relevant subsystems in the solution procedure and hence converges well. However this approach cannot be easily extended to include other subsystems as its implementation requires a new program to be developed for each combination of physical systems.

The multi-level Newton method uses existing discipline-specific solvers like the relaxation method. However, like the strongly coupled Newton method it accounts for full coupling in the solution procedure. Consequently, this approach offers the advantages of the previous two, i.e. it is easily implemented and converges well.

We consider a simple example to illustrate the multi-level Newton analysis of the coupled system of non-linear implicit equations:

$$f(x, y) = 0$$

$$g(y, x) = 0$$

We first guess the solution (x, y). For the given guess (x, y) we evaluate the *response* $\hat{x}(y)$ by fixing y and solving the subsystem problem $f(\hat{x}(y); y) = 0$ and similarly we evaluate $\hat{y}(x)$ by fixing x and solving the subsystem problem $g(\hat{y}(x); x) = 0$. Here the semi-colon is used to delimit the input from the output in the equations. As mentioned earlier the functions f and g are non-linear and implicit; therefore the evaluations of $\hat{x}(y)$ and $\hat{y}(x)$ use Newton's method. If the computed response $(\hat{x}(y), \hat{y}(x))$ equals the solution guesses (x, y), i.e. if

$$\mathscr{R}(x, y) = \begin{Bmatrix} x - \hat{x}(y) \\ \\ y - \hat{y}(x) \end{Bmatrix} = \mathbf{0} \tag{1}$$

then we have obtained a self-consistent solution and the analysis is complete. Otherwise the solution guess (x, y) is updated to $(x + \Delta x, y + \Delta y)$ via Newton's method, i.e. we solve

$$\mathrm{D}\mathscr{R}(x, y)\mathbf{\Delta} = -\mathscr{R}(x, y) \tag{2}$$

for $\mathbf{\Delta} = [\Delta x \;\; \Delta y]^{\mathrm{T}}$ where

$$\mathrm{D}\mathscr{R}(x, y) = \begin{bmatrix} 1 & -\hat{x}'(y) \\ \\ -\hat{y}'(x) & 1 \end{bmatrix} \tag{3}$$

We repeat this process of (i) evaluating responses $\hat{x}(y)$ and $\hat{y}(x)$, (ii) evaluating the residual in Equation (1) and (iii) if necessary updating the solution via Equation (2) until convergence is achieved, i.e. until Equation (1) is satisfied. The multi-level terminology is attributed to the uses of Newton's method to resolve the subsystem equations, e.g. $f(\hat{x}(y), y) = 0$ and the consistency equation (1).

The multi-level Newton method has been demonstrated to be an efficient and effective means for solving coupled multi-disciplinary analyses. Aluru and White use it for the analysis of an electromechanical system and demonstrate its superior performance over the relaxation method [6]. Bächtold *et al.* [7] employ it at the domain interface level to model the interaction between

the electrical and elastic domains—their method is referred to as the *surface Newton method*. These works use the computationally inefficient and possibly error prone finite difference approximation for evaluating the coupling, e.g. $\hat{x}'(y)$ in Equation (3) as described later in Section 2.

The algorithm suggested in the next section aims to further improve the performance of the multi-level Newton method by computing the inter-system coupling with analytical sensitivity evaluation techniques. Section 2 develops this improved multi-level Newton method while Section 3 applies it to solve a fluid–structure interaction problem.

2. MULTI-LEVEL NEWTON METHOD

In the multi-level Newton method we determine the responses of N coupled subsystems. The solution guess or the *state* of subsystem i is denoted by $\mathbf{u}_i$, e.g. x in our previous example. Each subsystem i has a residual equation that is used to evaluate its response function $\hat{\mathbf{u}}_i$, e.g. $\hat{x}$. The input to these equations are the states of the remaining $N-1$ subsystems, i.e. $\mathbf{u}^{\langle i\rangle} \stackrel{\text{def}}{=} [\mathbf{u}_1^{\mathrm{T}}, \mathbf{u}_2^{\mathrm{T}}, \ldots, \mathbf{u}_{i-1}^{\mathrm{T}}, \mathbf{u}_{i+1}^{\mathrm{T}}, \ldots, \mathbf{u}_N^{\mathrm{T}}]^{\mathrm{T}}$, e.g. y. We express these residual subsystem equations as $\mathbf{R}_i(\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle}) = \mathbf{0}$, where we use the hat to denote the implicitly defined response to be evaluated, e.g. $f(\hat{x}(y); y) = 0$.

These N subsystem residual equations comprise the subsystem analyses. The hatted response functions are evaluated via Newton's method and if we find after the N subsystem analyses that $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}) = \mathbf{u}_i$ for $i = 1, 2, \ldots, N$ then we have a self-consistent solution. To quantify this condition, we define the consistency, i.e. global residual equation

$$\mathscr{R}(\mathbf{u}) = \begin{Bmatrix} \mathbf{u}_1 - \hat{\mathbf{u}}_1(\mathbf{u}^{\langle 1\rangle}) \\ \vdots \\ \mathbf{u}_i - \hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}) \\ \vdots \\ \mathbf{u}_N - \hat{\mathbf{u}}_N(\mathbf{u}^{\langle N\rangle}) \end{Bmatrix} = \mathbf{0} \tag{4}$$

where $\mathbf{u} = [\mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_N^{\mathrm{T}}]^{\mathrm{T}}$, cf. Equation (1).

If the global residual equation is satisfied, then we have a self-consistent solution and our problem is solved. Otherwise, we do not have a self-consistent solution and update our states to $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}$ via a Newton iteration on the global level,¶ i.e. on the outer loop. The Newton update $\Delta\mathbf{u} = [\Delta\mathbf{u}_1^{\mathrm{T}}, \ldots, \Delta\mathbf{u}_N^{\mathrm{T}}]^{\mathrm{T}}$ about the iteration k solution $\mathbf{u}^k$ is obtained from

$$\mathrm{D}\mathscr{R}(\mathbf{u}^k)\Delta\mathbf{u} = -\mathscr{R}(\mathbf{u}^k) \tag{5}$$

¶Superscripts refer to the iteration number.

where

$$\mathrm{D}\mathscr{R} = \begin{bmatrix} \mathbf{I} & -\dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_2} & -\dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_3} & \cdots & -\dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_N} \\ -\dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_1} & \mathbf{I} & -\dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_3} & \cdots & -\dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_N} \\ \vdots & \ddots & \mathbf{I} & \ddots & \vdots \\ -\dfrac{\partial \hat{\mathbf{u}}_{N-1}}{\partial \mathbf{u}_1} & -\dfrac{\partial \hat{\mathbf{u}}_{N-1}}{\partial \mathbf{u}_2} & \cdots & \mathbf{I} & -\dfrac{\partial \hat{\mathbf{u}}_{N-1}}{\partial \mathbf{u}_N} \\ -\dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_1} & -\dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_2} & \cdots & -\dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_{N-1}} & \mathbf{I} \end{bmatrix} \tag{6}$$

and $\mathbf{I}$ is the identity matrix, cf. Equations (2) and (3).

Summarizing, the N hatted response functions $\hat{\mathbf{u}}_i$ are evaluated by N subsystem analyses via Newton's method; their values are not updated in the global Newton iteration. On the other hand, the N unhatted states $\mathbf{u}_i$ are updated by the global Newton iteration to resolve the global residual equation (5); they are not updated in the subsystem Newton iterations. The subsystem analyses for the evaluations of $\hat{\mathbf{u}}_i$, followed by the global update of $\mathbf{u}_i$ are repeated until convergence of Equation (4) is obtained.

As seen above the Jacobian $\mathrm{D}\mathscr{R}$ in Equation (6) is of no particular structure. And as seen momentarily the partitions of $\mathrm{D}\mathscr{R}$, e.g. $\partial \hat{\mathbf{u}}_1/\partial \mathbf{u}_2$, require the potentially costly computations of the inverse Jacobians of the subsystem analyses, e.g. $[\partial \mathbf{R}_1/\partial \hat{\mathbf{u}}_1]^{-1}$. For these reasons, i.e. to reduce the memory requirements and computations, we adopt an iterative matrix free solver to evaluate the solution $\Delta \mathbf{u}$ of the update Equation (5). In such solvers the product of the Jacobian matrix with the iterate vector $\mathbf{s} = [\mathbf{s}_1^{\mathrm{T}}, \ldots, \mathbf{s}_N^{\mathrm{T}}]^{\mathrm{T}}$, i.e.

$$\mathrm{D}\mathscr{R}\,\mathbf{s} = \begin{Bmatrix} \mathbf{s}_1 - \dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_2}\mathbf{s}_2 - \dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_3}\mathbf{s}_3 - \cdots - \dfrac{\partial \hat{\mathbf{u}}_1}{\partial \mathbf{u}_N}\mathbf{s}_N \\ \mathbf{s}_2 - \dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_1}\mathbf{s}_1 - \dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_3}\mathbf{s}_3 - \cdots - \dfrac{\partial \hat{\mathbf{u}}_2}{\partial \mathbf{u}_N}\mathbf{s}_N \\ \vdots \\ \mathbf{s}_N - \dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_1}\mathbf{s}_1 - \dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_2}\mathbf{s}_2 - \dfrac{\partial \hat{\mathbf{u}}_N}{\partial \mathbf{u}_{N-1}}\mathbf{s}_{N-1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{s}_1 - \dfrac{D\hat{\mathbf{u}}_1}{D\mathbf{u}^{\langle 1\rangle}}\mathbf{s}^{\langle 1\rangle} \\ \mathbf{s}_2 - \dfrac{D\hat{\mathbf{u}}_2}{D\mathbf{u}^{\langle 2\rangle}}\mathbf{s}^{\langle 2\rangle} \\ \vdots \\ \mathbf{s}_N - \dfrac{D\hat{\mathbf{u}}_N}{D\mathbf{u}^{\langle N\rangle}}\mathbf{s}^{\langle N\rangle} \end{Bmatrix} \tag{7}$$

is evaluated in a matrix-free manner where $\mathbf{s}^{\langle i\rangle} = [\mathbf{s}_1^{\mathrm{T}}, \mathbf{s}_2^{\mathrm{T}}, \ldots \mathbf{s}_{i-1}^{\mathrm{T}}, \mathbf{s}_{i+1}^{\mathrm{T}}, \ldots, \mathbf{s}_N^{\mathrm{T}}]^{\mathrm{T}}$. Moreover the product is evaluated efficiently, i.e. it does not require any inverse computations. Multiple

evaluations of Equation (7) are required until Equation (5) is satisfied to the desired degree of accuracy whereupon $\Delta\mathbf{u} = \mathbf{s}$. It is imperative to efficiently calculate the matrix–vector product of Equation (7).

The inter-system coupling terms are shown as the products $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle}$, $i = 1, \ldots, N$. Traditionally these directional derivatives are computed using the finite difference approximation, i.e.

$$-\frac{D\hat{\mathbf{u}}_i}{D\mathbf{u}^{\langle i\rangle}}\,\mathbf{s}^{\langle i\rangle} \approx \frac{1}{\theta}[\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}) - \hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle})] \tag{8}$$

where θ is the perturbation and $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle})$ is the solution to the subsystem i analysis in which the input $\mathbf{u}^{\langle i\rangle}$ is perturbed to $\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}$, i.e. the solution to $\mathbf{R}_i(\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}) = \mathbf{0}$. This method is popular because it is easily implemented allowing the discipline-specific solvers to be treated as black-boxes. Unfortunately this approach is plagued by three drawbacks. (i) The value of θ must be in an interval within which the numerically computed derivatives yield accurate results. This interval can change with $\mathbf{u}^{\langle i\rangle}$ and with the subsystem analyses. Indeed Aluru and White [6] report that the multi-level Newton algorithm is highly sensitive to the value of θ. In Equation (8) it should be noted that one subsystem analysis $\mathbf{R}_i(\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}) = \mathbf{0}$ is needed for the evaluation of $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle}$. However, if a single acceptable value for θ does not exist for $i = 1, \ldots, i-1, i+1, \ldots, N$, then this finite difference computation must be broken up as $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle} = \sum_{j=1, j\neq i}^{N} \frac{1}{\theta_j}[\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}) - \hat{\mathbf{u}}_i(\mathbf{u}_1, \ldots, \mathbf{u}_j + \theta_j\mathbf{s}_j, \ldots, \mathbf{u}_{i-1}, \mathbf{u}_{i+1}, \ldots, \mathbf{u}_N)]$ which would require multiple subsystem analyses. (ii) The finite difference approximation in Equation (8) requires at least N additional non-linear subsystem analyses $\mathbf{R}_i(\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}) = \mathbf{0}$ for a total of $N \times M$ subsystem analyses per global Newton iteration of the global equation (4), where M is the number of iterations required to solve Equation (5). This is a computationally expensive proposal. (iii) By using finite difference sensitivities to approximate the product $\mathrm{D}\mathscr{R}\mathbf{s}$ we attain at best super linear convergence rather than terminal quadratic convergence of the global analysis in Equation (4) [9]. Hence more global iterations will be required. Haftka *et al.* discuss the cascading numerical errors associated with the finite difference approximation [8].

The computational disadvantages associated with the finite difference computation of Equation (8) can be circumvented by analytically evaluating the inter-system coupling terms in Equation (7). To this end, let us first consider the subsystem i residual equation:

$$\mathbf{R}_i(\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle}) = \mathbf{0} \tag{9}$$

which we solve by Newton's method for $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle})$ given $\mathbf{u}^{\langle i\rangle}$. We use the functional relation for $\hat{\mathbf{u}}_i$ since the value $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle})$ clearly depends on the input value $\mathbf{u}^{\langle i\rangle}$. Differentiating Equation (9) with respect to $\mathbf{u}_j$, multiplying by $\mathbf{s}_j$ where $j \neq i$ and rearranging the result gives the following, so called, pseudo problem:

$$\frac{\partial\mathbf{R}_i}{\partial\hat{\mathbf{u}}_i}\left[\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_j}\,\mathbf{s}_j\right] = -\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}_j}\,\mathbf{s}_j \tag{10}$$

which we can solve for $(\partial\hat{\mathbf{u}}_i/\partial\mathbf{u}_j)\,\mathbf{s}_j$. Since the problem is linear, we can generalize the above to obtain

$$\frac{\partial\mathbf{R}_i}{\partial\hat{\mathbf{u}}_i}\left\{\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_1}\,\mathbf{s}_1+\cdots+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_{i-1}}\,\mathbf{s}_{i-1}+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_{i+1}}\,\mathbf{s}_{i+1}+\cdots+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_N}\,\mathbf{s}_N\right\}$$

$$=-\left\{\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}_1}\,\mathbf{s}_1+\cdots+\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}_{i-1}}\,\mathbf{s}_{i-1}+\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}_{i+1}}\,\mathbf{s}_{i+1}+\cdots+\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}_N}\,\mathbf{s}_N\right\} \tag{11}$$

or

$$\frac{\partial\mathbf{R}_i}{\partial\hat{\mathbf{u}}_i}\,\frac{D\hat{\mathbf{u}}_i}{D\mathbf{u}^{\langle i\rangle}}\,\mathbf{s}^{\langle i\rangle}=-\frac{\partial\mathbf{R}_i}{\partial\mathbf{u}^{\langle i\rangle}}\,\mathbf{s}^{\langle i\rangle} \tag{12}$$

The above pseudo-problem is solved once for each of the N disciplines for the given $\mathbf{s}=\{\mathbf{s}_1^{\mathrm{T}},\ldots,\mathbf{s}_N^{\mathrm{T}}\}^{\mathrm{T}}$. We note that the pseudo analysis for $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\,\mathbf{s}^{\langle i\rangle}$ comes from the direct differentiation sensitivity analysis (see Scheuing and Tortorelli [10] for details).

Using Newton's method to solve Equation (9) we recognize that $\partial\mathbf{R}_i/\partial\hat{\mathbf{u}}_i$ is the subsystem i Jacobian which we use to evaluate the $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i\rangle})$. So if a direct solver is used to compute the solution update $\Delta\hat{\mathbf{u}}_i$ for Equation (9) then we can evaluate

$$\frac{D\hat{\mathbf{u}}_i}{D\mathbf{u}^{\langle i\rangle}}\,\mathbf{s}^{\langle i\rangle}=\left\{\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_1}\,\mathbf{s}_1+\cdots+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_{i-1}}\,\mathbf{s}_{i-1}+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_{i+1}}\,\mathbf{s}_{i+1}+\cdots+\frac{\partial\hat{\mathbf{u}}_i}{\partial\mathbf{u}_N}\,\mathbf{s}_N\right\}$$

in Equation (11) by forming the so called pseudo vectors $(\partial\mathbf{R}_i/\partial\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle}$ followed by a back substitution with the factored Jacobian. If iterative solvers are used to compute the solution update $\Delta\hat{\mathbf{u}}_i$, then we presumably already have an efficient means to solve Equation (11) as it is of the same form as the subsystem update equation, i.e. $(\partial\mathbf{R}_i/\partial\hat{\mathbf{u}}_i)\Delta\hat{\mathbf{u}}_i=-\mathbf{R}_i$. We emphasize here that the pseudo analysis for $(D\mathbf{R}_i/D\mathbf{u}^{\langle i\rangle})\,\mathbf{s}^{\langle i\rangle}$ is linear as opposed to the subsystem i analysis which may be non-linear.

Summarizing, at each iteration k of the global analysis we solve Equation (5) iteratively for $\Delta\mathbf{u}$. At each of these iterations we are supplied with a solution iterate vector $\mathbf{s}$ from which we evaluate the vectors $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\,\mathbf{s}^{\langle i\rangle}$ for $i=1,\ldots,N$, cf. Equation (7). By using Newton's method and a direct linear solver for the subsystem analyses, we can compute the vectors $(D\hat{\mathbf{u}}_i/D\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle}$ by back substitutions with the $\partial\mathbf{R}_i/\partial\hat{\mathbf{u}}_i$ Jacobians. This requires a total of $N\times M$ vector computations and back substitutions per Newton update iteration of Equation (5). Using an iterative solver for the $P\leqslant N$ of the subsystem analyses requires a total of $P\times M$ iterative solutions to the linear equation (11) in place of the $P\times M$ vector computations and back substitutions. In either case, the computational savings over the finite difference method which requires at least $N\times M$ non-linear subsystem analyses are substantial. Moreover, our computations are not subject to truncation errors and hence we obtain terminal quadratic convergence of Equation (4).

Analytical evaluation of the pseudo load $(D\mathbf{R}_i/D\mathbf{u}^{\langle i\rangle})\mathbf{s}^{\langle i\rangle}$ requires the differentiation of residual equations and thereby code modifications. An alternative way is to use a semi-analytical

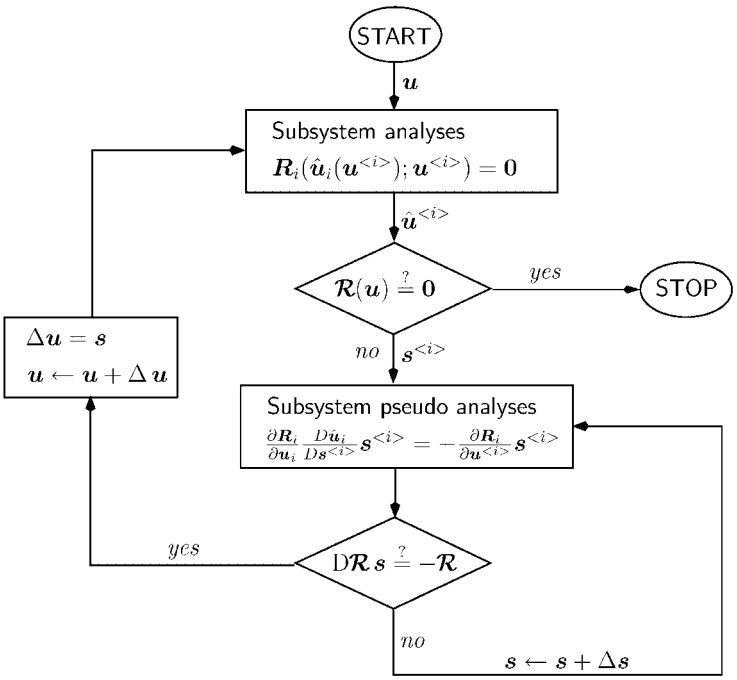


Figure 1. Multi-level Newton procedure.

technique, i.e. compute the pseudo load by the finite difference approximation

$$-\frac{D\mathbf{R}_i}{D\mathbf{u}^{\langle i\rangle}}\,\mathbf{s}^{\langle i\rangle} \approx \frac{1}{\theta}[\mathbf{R}_i(\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle}) - \mathbf{R}_i(\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle})]$$

$$\approx -\frac{1}{\theta}\,\mathbf{R}_i(\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle}) \tag{13}$$

where we use the fact that the subsystem analysis for $\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle})$ has been completed so $\mathbf{R}_i(\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle}) \approx \mathbf{0}$. Note that we do not solve the above for $\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle})$. Rather we only perform the single residual evaluation $\mathbf{R}_i(\hat{\mathbf{u}}(\mathbf{u}^{\langle i\rangle}); \mathbf{u}^{\langle i\rangle} + \theta\mathbf{s}^{\langle i\rangle})$. Implementation of this method requires no code modification to the subsystem solvers which is desirable. Unfortunately, the convergence rate of the semi-analytical method is often slowed for the same reasons as that of the finite difference based multi-level Newton method. Olhoff and Rasmussen [11] suggest ways to improve the accuracy by using the central finite difference scheme in place of the forward difference scheme of Equation (8) in a general sensitivity analysis setting.

As alluded to above, the Jacobian partitions, e.g. $\partial\hat{\mathbf{u}}_1/\partial\mathbf{u}_2$ in $D\mathscr{R}$ (cf. Equation (6)), require the computation of the subsystem Jacobian inverses, e.q. $[\partial\mathbf{R}_1/\partial\hat{\mathbf{u}}_1]^{-1}$. This is seen from Equation (10), e.g. $\partial\hat{\mathbf{u}}_1/\partial\mathbf{u}_2 = -[\partial\mathbf{R}_1/\partial\hat{\mathbf{u}}_1]^{-1}\,\partial\mathbf{R}_1/\partial\mathbf{u}_2$. As such the computation of the Jacobian $D\mathscr{R}$ and the subsequent use of direct or non-matrix-free iterative methods to resolve Equation (5) requires both more computation and storage than the proposed method.

Summarizing the subsystem analyses are performed to obtain the response function values $\hat{\mathbf{u}}_i(\mathbf{u}^{\langle i \rangle})$ for the solution state $\mathbf{u}^{\langle i \rangle}$. If convergence of Equation (4) is not reached we update the state vectors $\mathbf{u}_i$ by Newton's method, cf. Equation (5). An iterative linear solver is used to evaluate the update in Equation (5). Each iteration for the analysis of Equation (5) requires the directional derivative computations (cf. Equation (7)) that are obtained by solving the pseudo problems (cf. Equation (12)). The solution procedure is illustrated in Figure 1.

3. FLUID–STRUCTURE INTERACTION EXAMPLE

Here we model the non-linear interaction of an incompressible laminar Newtonian fluid in steady-state with a solid structure undergoing finite deformation. Coupling arises from the viscous flow that exerts traction to deform the solid structure, and from the deformed structure that defines the fluid domain and hence the traction.

Computationally, the fluid–structure interaction problem consists of three coupled analyses, i.e. $N = 3$; namely the computations of the incident flow field, structural displacement and fluid rezone displacement [12]. Because the fluid analysis uses an Eulerian kinematic description and because we do not know the fluid deformed configuration *a priori* the fluid domain is deformed from a known unloaded (undeformed) configuration via the fluid rezone displacement field. A laplace-type equation governs the fluid rezone displacement field [13–15].

Fluid–structure interaction problems have been solved using relaxation-based iteration algorithms [1, 2] and strongly-coupled algorithms [3–5, 16]. Although the relaxation-based approaches are easy to implement, they are inappropriate for solving problems with *strong* coupling as they are slow to converge if they converge at all [5]. Ghattas and Li [4] and Lund *et al.* [5] developed a strongly coupled finite element model for steady-state non-linear fluid–structure interaction problems in which the solutions are obtained by satisfying the interface conditions across the fluid–structure boundary in addition to satisfying the usual governing field equations in the fluid and structural domains. However due to the difficulties originating from (i) an ill-conditioned Jacobian and (ii) the storage requirements resulting from the dense matrices associated with the Jacobians of the interface conditions, they neglect some coupling terms. Neglecting these terms leads to decoupled computations. Consequently the solution converges linearly at best and may not converge if the neglected coupling is strong. In later work Ghattas and Li [16] incorporate all coupling and employ an iterative linear solver to circumvent their previous storage limitations. They use the approximate Jacobian of their previous approach [4] as the preconditioner. However the matrix–vector products are computed by forward finite difference approximations which lead to slower convergence rates and substantially increased computational costs, as explained in Section 2.

3.1. MNM formulation for fluid–structure interaction

Figure 2 illustrates the relevant subsystem domains for the fluid–structure interaction problem: Ω_q is the fluid domain which is defined by the deformed structure (see Figure 2(b)) since the fluid analysis uses an Eulerian kinematic description; Ω_w is the fluid rezone domain (it is deformed into Ω_q) and Ω_u is the structural domain; these latter two domains are defined by the undeformed configuration because their corresponding analyses use Lagrangian kinematic descriptions. The boundaries for the respective domains are denoted by Γ_q, Γ_w and Γ_u where Γ_q

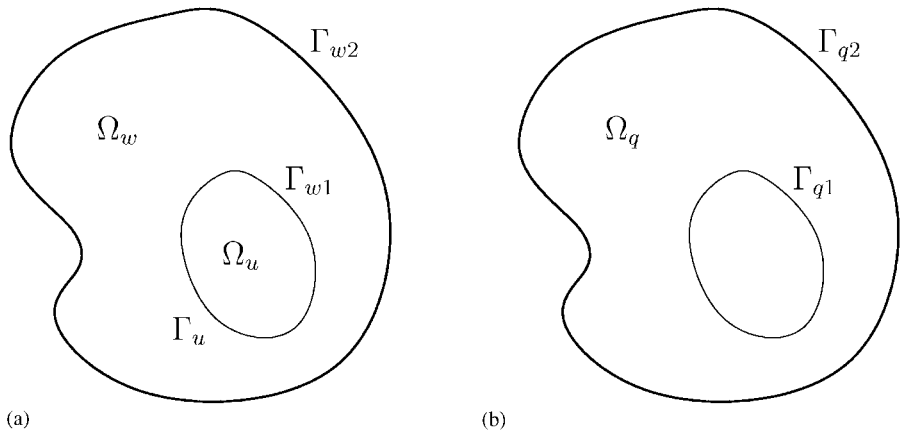


Figure 2. Domains of the fluid–structure interaction problem.

and Γ_w may consist of disjoint surfaces as seen in Figure 2, e.g. $\Gamma_q = \Gamma_{q1} \cup \Gamma_{q2}$ and $\Gamma_{q1} \cap \Gamma_{q2} = \emptyset$. Note again that the fluid rezone displacement deforms the Ω_w configuration into the Ω_q configuration. The non-homogeneous essential boundary conditions for this fluid rezone analysis are dictated by the structural displacement of the fluid–structure boundary $\Gamma_B = \Gamma_u \cap \Gamma_w$.

Using these domains a self-consistent residual is formulated as

$$\mathscr{R}(\mathbf{q}, \mathbf{w}, \mathbf{u}) = \begin{Bmatrix} \mathbf{q} - \hat{\mathbf{q}}(\mathbf{w}, \mathbf{u}) \\ \mathbf{w} - \hat{\mathbf{w}}(\mathbf{u}) \\ \mathbf{u} - \hat{\mathbf{u}}(\mathbf{w}, \mathbf{u}) \end{Bmatrix} = \mathbf{0} \tag{14}$$

where $\mathbf{q}$, $\mathbf{w}$ and $\mathbf{u}$ denote parameters describing the states for the flow field, fluid rezone displacement and structural displacement, respectively, and $\hat{\mathbf{q}}(\mathbf{w}, \mathbf{u})$, $\hat{\mathbf{w}}(\mathbf{u})$ and $\hat{\mathbf{u}}(\mathbf{w}, \mathbf{u})$, respectively, are their response functions calculated from the individual subsystem analyses. The interaction mechanism within the system is described by the dependencies above, e.g. $\hat{\mathbf{w}}$ on $\mathbf{u}$, however our choice is not unique. Over the boundary Γ_B the fluid rezone displacement is equated to the structural displacement and hence the flow response $\hat{\mathbf{q}}$ has dependencies on the states $\mathbf{w}$ and $\mathbf{u}$ which deform Ω_w into Ω_q. As just discussed the fluid rezone displacement response $\hat{\mathbf{w}}$ has a dependency on only the state of $\mathbf{u}$.[‖] The structural deformation response $\hat{\mathbf{u}}$ has a dependency on the fluid-induced traction which is obtained from the current response $\hat{\mathbf{q}}$ and hence it depends on the states $\mathbf{w}$ and $\mathbf{u}$. This dependency reduces storage requirements as explained shortly, however it requires the completion of the fluid subsystem analysis prior to the structural subsystem analysis in each Newton iteration of the global problem whereas the fluid rezone subsystem analysis can be performed irrespective of the order.

The update for the multi-level Newton analysis at iteration k, i.e. at $\mathbf{q}^k$, $\mathbf{w}^k$ and $\mathbf{u}^k$, is

$$\mathrm{D}\mathscr{R}(\mathbf{q}^k, \mathbf{w}^k, \mathbf{u}^k)\,\boldsymbol{\Delta} = -\mathscr{R}(\mathbf{q}^k, \mathbf{w}^k, \mathbf{u}^k) \tag{15}$$

[‖] We assume that the remaining boundaries of Ω_w and hence Ω_q are fixed, i.e. we do not consider moving boundaries.

where $\mathbf{\Delta} = [\mathbf{\Delta}_q^{\mathrm{T}} \ \mathbf{\Delta}_w^{\mathrm{T}} \ \mathbf{\Delta}_u^{\mathrm{T}}]^{\mathrm{T}}$ and

$$\mathrm{D}\mathscr{R}(\mathbf{q}^k, \mathbf{w}^k, \mathbf{u}^k) = \begin{bmatrix} \mathbf{I} & -\dfrac{\partial \hat{\mathbf{q}}}{\partial \mathbf{w}}(\mathbf{w}^k, \mathbf{u}^k) & -\dfrac{\partial \hat{\mathbf{q}}}{\partial \mathbf{u}}(\mathbf{w}^k, \mathbf{u}^k) \\ \mathbf{0} & \mathbf{I} & -\dfrac{\partial \hat{\mathbf{w}}}{\partial \mathbf{u}}(\mathbf{u}^k) \\ \mathbf{0} & -\dfrac{\partial \hat{\mathbf{u}}}{\partial \mathbf{w}}(\mathbf{w}^k, \mathbf{u}^k) & \mathbf{I} - \dfrac{\partial \hat{\mathbf{u}}}{\partial \mathbf{u}}(\mathbf{w}^k, \mathbf{u}^k) \end{bmatrix} \tag{16}$$

Once the solution to Equation (15) is found we update the state vectors by

$$\begin{aligned} \mathbf{q}^{k+1} &= \mathbf{q}^k + \mathbf{\Delta}_q \\ \mathbf{w}^{k+1} &= \mathbf{w}^k + \mathbf{\Delta}_w \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \mathbf{\Delta}_u \end{aligned} \tag{17}$$

The iterative solution to Equation (15) requires the matrix–vector products

$$\mathrm{D}\mathscr{R}(\mathbf{q}^k, \mathbf{w}^k, \mathbf{u}^k)\mathbf{s} = \begin{Bmatrix} \mathbf{s}_q - \dfrac{\partial \hat{\mathbf{q}}(\mathbf{w}^k, \mathbf{u}^k)}{\partial \mathbf{w}} \mathbf{s}_w - \dfrac{\partial \hat{\mathbf{q}}(\mathbf{w}^k, \mathbf{u}^k)}{\partial \mathbf{u}} \mathbf{s}_u \\ \mathbf{s}_w - \dfrac{\partial \hat{\mathbf{w}}(\mathbf{u}^k)}{\partial \mathbf{u}} \mathbf{s}_u \\ \mathbf{s}_u - \dfrac{\partial \hat{\mathbf{u}}(\mathbf{w}^k, \mathbf{u}^k)}{\partial \mathbf{w}} \mathbf{s}_w - \dfrac{\partial \hat{\mathbf{u}}(\mathbf{w}^k, \mathbf{u}^k)}{\partial \mathbf{u}} \mathbf{s}_u \end{Bmatrix} \tag{18}$$

where $\mathbf{s} = [\mathbf{s}_q^{\mathrm{T}} \ \mathbf{s}_w^{\mathrm{T}} \ \mathbf{s}_u^{\mathrm{T}}]^{\mathrm{T}}$. As discussed earlier, the directional derivatives in Equation (18) are evaluated analytically by the direct differentiation method.

3.2. Subsystem computations

The fluid analysis uses a standard Galerkin velocity/pressure mixed formulation to model the steady laminar flow of an incompressible Newtonian fluid [17]. The response $\hat{\mathbf{q}}$ and its directional derivative with respect to $\mathbf{s}_w$ and $\mathbf{s}_u$ are evaluated from

$$\mathbf{R}_q(\hat{\mathbf{q}}(\mathbf{w}, \mathbf{u}); \ \mathbf{w}, \mathbf{u}) = \mathbf{0} \tag{19}$$

$$\frac{\partial \mathbf{R}_q}{\partial \hat{\mathbf{q}}} \left\{ \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{u}} \mathbf{s}_u \right\} = - \left\{ \frac{\partial \mathbf{R}_q}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \mathbf{R}_q}{\partial \mathbf{u}} \mathbf{s}_u \right\} \tag{20}$$

The right-hand side of Equation (20), i.e. the pseudo load, requires the differentiation of the fluid subsystem equation with respect to node co-ordinates (see Appendix A for details). Derivatives are computed at the element level and then assembled into a global vector in

much the same way that the residual vector is computed in the fluid analysis. This pseudo load computation requires some enhancements to the fluid subsystem analysis. However the cost associated with this computation is only slightly more expensive than that of one fluid subsystem residual vector evaluation.

The reaction forces on the fluid–solid interface in the fluid analysis serve as externally applied loads in the structural analysis. To compute the reaction forces we define $\hat{\mathbf{q}}^p$ i.e. the vector with prescribed degrees of freedom. The reaction forces $\mathbf{P}_q^p$ are then computed from

$$\mathbf{P}_q^p(\mathbf{w},\mathbf{u}) = -\mathbf{S}_q^p(\hat{\mathbf{q}}^p, \hat{\mathbf{q}}(\mathbf{w},\mathbf{u})) \tag{21}$$

where $\mathbf{S}_q^p$ is the internal force vector associated with the nodes that have the prescribed degrees of freedom in the fluid subsystem analysis.** This vector is assembled analogously to the internal force vector that is used to compute $\mathbf{R}_q$ (see Reference [18] for details). Since we solve $\mathbf{R}_q(\hat{\mathbf{q}}(\mathbf{w},\mathbf{u});\mathbf{w},\mathbf{u}) = \mathbf{0}$ for $\hat{\mathbf{q}}(\mathbf{w},\mathbf{u})$ we can directly evaluate the $\mathbf{P}_q^p(\mathbf{w},\mathbf{u}) = -\mathbf{S}_q^p(\hat{\mathbf{q}}^p, \hat{\mathbf{q}}(\mathbf{w},\mathbf{u}))$ as $\hat{\mathbf{q}}^p$ is known. The reaction force directional derivatives are also evaluated directly by differentiating Equation (21), i.e.

$$\left\{ \frac{\partial \mathbf{P}_q^p}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \mathbf{P}_q^p}{\partial \mathbf{u}} \mathbf{s}_u \right\} = -\frac{\partial \mathbf{S}_q^p}{\partial \hat{\mathbf{q}}} \left\{ \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \hat{\mathbf{q}}}{\partial \mathbf{u}} \mathbf{s}_u \right\} - \left\{ \frac{\partial \mathbf{S}_q^p}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \mathbf{S}_q^p}{\partial \mathbf{u}} \mathbf{s}_u \right\} \tag{22}$$

where $\partial \mathbf{S}_q^p / \partial \hat{\mathbf{q}}$ corresponds to the $\mathbf{K}^{pf}$ matrix in Reference [18]. The elements of $\mathbf{P}_q^p$ and the left-hand side vector of Equation (22) that map to the nodes on Γ_B are used to define the externally applied loads and their derivatives in the structural analysis.

As mentioned earlier the way to describe the interactions between the fluid and structural domains is not unique. We could introduce the traction as an additional response field which would increase the number of degrees of freedom or use the reaction from the previous global iteration $(k-1)$ i.e. $(\mathbf{P}_q^p)^{k-1}$ which would increase storage requirements and, based on our numerical experiments, possibly erode the initial convergence rates. All approaches require similar amounts of code modifications.

The fluid rezone analysis adopted in this work solves the modified elasticity equation proposed by Masud and Hughes [15]. Various other approaches are discussed in Reference [13]. The response $\hat{\mathbf{w}}$ and its directional sensitivity with respect to the fluid–structure boundary location, i.e. with respect to the structural displacement $\mathbf{u}$ are evaluated from

$$\mathbf{R}_w(\hat{\mathbf{w}}(\mathbf{u}); \mathbf{w}_B(\mathbf{u})) = \mathbf{0} \tag{23}$$

$$\frac{\partial \mathbf{R}_w}{\partial \hat{\mathbf{w}}} \frac{\partial \hat{\mathbf{w}}}{\partial \mathbf{u}} \mathbf{s}_u = -\frac{\partial \mathbf{R}_w}{\partial \mathbf{w}_B} \mathbf{W}_B \mathbf{s}_u \tag{24}$$

In the above $\mathbf{w}_B(\mathbf{u}) = \mathbf{W}_B\mathbf{u}$ defines the essential boundary conditions corresponding to the nodes which lie on Γ_B where $\mathbf{W}_B$ a Boolean matrix that projects the necessary elements of $\mathbf{u}$ to the necessary elements of $\mathbf{w}_B$.†† The matrix $\partial \mathbf{R}_w / \partial \mathbf{w}_B$ corresponds to the $\mathbf{K}^{fp}$ matrix of

**These nodes include those on the fluid–structure interface where the no-slip boundary condition is applied.

††Recall that the nodal vectors in $\mathbf{w}$ and $\mathbf{u}$ are equal at the corresponding nodes over Γ_B.

the fluid rezone analysis and is similar to the $\partial \mathbf{S}_q^p / \partial \hat{\mathbf{q}}$ matrix discussed in the fluid sensitivity analysis.

Minimal code modifications are required of the fluid rezone analysis software to compute the derivatives. Basically we need access to the decomposed Jacobian $\partial \mathbf{R}_u / \partial \hat{\mathbf{u}}$ and the $\partial \mathbf{R}_w / \partial \mathbf{w}_\mathrm{B}$ matrix.

The structural analysis assumes geometrically non-linear elastic behaviour. The response $\hat{\mathbf{u}}$ and its directional sensitivity with respect to $\mathbf{s}_w$ and $\mathbf{s}_u$ are computed from

$$\mathbf{R}_u(\hat{\mathbf{u}}(\mathbf{w},\mathbf{u}); \mathbf{F}_\mathrm{B}(\mathbf{w},\mathbf{u})) = \mathbf{0} \tag{25}$$

$$\frac{\partial \mathbf{R}_u}{\partial \hat{\mathbf{u}}} \left\{ \frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \hat{\mathbf{u}}}{\partial \mathbf{u}} \mathbf{s}_u \right\} = -\frac{\partial \mathbf{R}_u}{\partial \mathbf{F}_\mathrm{B}} \left\{ \frac{\partial \mathbf{F}_\mathrm{B}}{\partial \mathbf{w}} \mathbf{s}_w + \frac{\partial \mathbf{F}_\mathrm{B}}{\partial \mathbf{u}} \mathbf{s}_u \right\} \tag{26}$$

where $\mathbf{F}_\mathrm{B} = \mathbf{P}_q^p$ (cf. Equation (21)) contains the externally applied nodal forces acting on the nodes which lie on Γ_B in fluid analysis and $\partial \mathbf{R}_u / \partial \mathbf{F}_\mathrm{B}$ is a Boolean matrix that projects the appropriate elements of the fluid reaction force vector to the appropriate elements of the structural applied force vector. The vector $\{\partial \mathbf{F}_\mathrm{B}/\partial \mathbf{w}\, \mathbf{s}_w + \partial \mathbf{F}_\mathrm{B}/\partial \mathbf{u}\, \mathbf{s}_u\} = \{\partial \mathbf{P}_q^p/\partial \mathbf{w}\, \mathbf{s}_w + \partial \mathbf{P}_q^p/\partial \mathbf{u}\, \mathbf{s}_u\}$ is obtained from the pseudo problem of the fluid analysis (cf. Equation (22)).

As in the fluid rezone analysis, few code enhancements are required of the structural analysis software; we only need access to the decomposed Jacobian $\partial \mathbf{R}_u / \partial \hat{\mathbf{u}}$.

3.3. Numerical implementation

Our scheme is implemented numerically using the following procedure. First the state vectors $\mathbf{q}$, $\mathbf{w}$ and $\mathbf{u}$ are initialized. Then for each Newton iteration the fluid mesh is defined in accordance to the state vectors $\mathbf{w}$ and $\mathbf{u}$ and Equation (A3) and then the fluid analysis of Equations (19) and (21) is performed. Upon completion of the fluid analysis the response $\hat{\mathbf{q}}(\mathbf{w},\mathbf{u})$, reaction force vector $\mathbf{P}_q^p$ and the tangent stiffness matrix in decomposed form are saved for later use. The structural analyses of Equations (23) and (25) follow using the recently computed force vector $\mathbf{P}_q^p$ in the structural analysis. Like the fluid analysis, the responses $\hat{\mathbf{w}}(\mathbf{u})$ and $\hat{\mathbf{u}}(\mathbf{w},\mathbf{u})$ and their tangent stiffness matrices in decomposed form are saved. The fluid rezone analysis can be performed irrespective of the order.

Having computed the responses $\hat{\mathbf{q}}(\mathbf{w},\mathbf{u})$, $\hat{\mathbf{w}}(\mathbf{u})$ and $\hat{\mathbf{u}}(\mathbf{w},\mathbf{u})$, the global residual in Equation (14) is evaluated and the update vector $\boldsymbol{\Delta}$ is computed iteratively from Equation (15). Each iteration of Equation (15) requires the evaluation of Equation (18) which is done by solving the pseudo analyses of Equations (20), (22), (24) and (26). Upon convergence of Equation (15) we set $\boldsymbol{\Delta} = \mathbf{s}$ and update the state vectors $\mathbf{q}$, $\mathbf{w}$ and $\mathbf{u}$ via Equation (17). Iterations continue until convergence of the global residual Equation (14).

All subsystem analyses employ a skyline direct solver [19] whereas the iterative solver GMRES is used to solve Equation (15). GMRES [20] exhibits good convergence rates without using a preconditioner for the examples studied in this work. The pressure computed in the fluid analysis is scaled by the maximum pressure to improve the conditioning of the Jacobian $\mathrm{D}\mathscr{R}$ (*cf.* Equation (20)), i.e. we directly compute the pressure in the fluid analysis of Equation (19) and compute the scaled pressure in the global analysis of Equation (14).

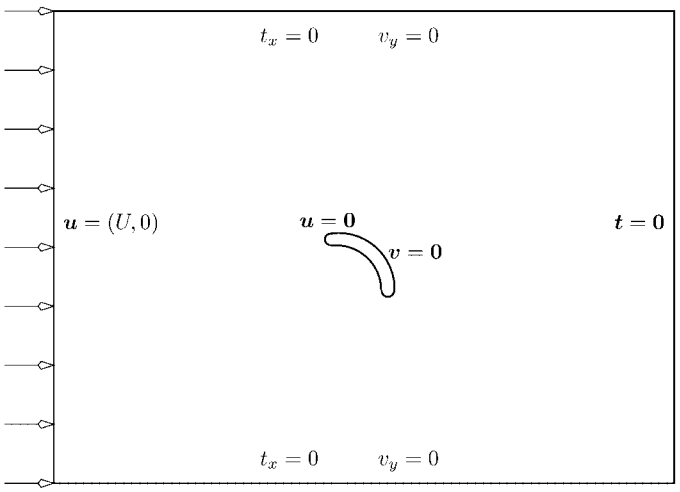


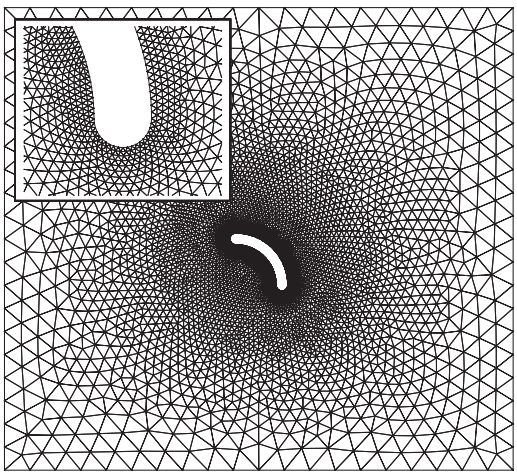
Figure 3. Flow is left to right; flap is fixed at the upstream end. A no-slip boundary condition on the flap and traction-free boundary conditions on the fixed fluid boundary as indicated by $t_x = 0$ or $t_y = 0$ are applied in the fluid analysis.

3.4. Results and discussions

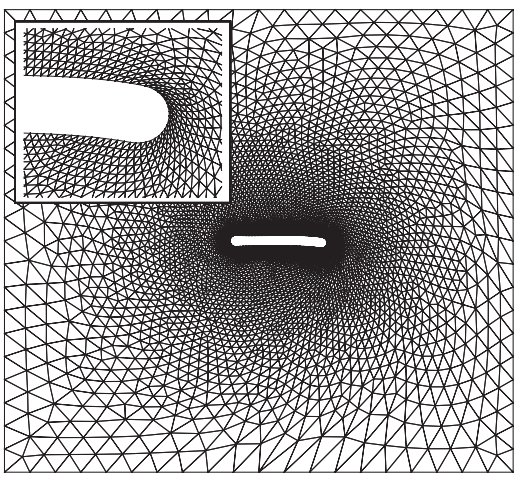
Here we consider a slender elastic arch that is placed in a fluid flowing from left to right (cf. Figure 3). A low Reynolds number ($Re = 10$) is used to ensure a laminar flow, however, the structural material is sufficiently soft rendering a strong coupling between the fluid flow and structure responses. Conforming elements are used across the fluid–structure boundary [21] (cf. Figure 4(a)). The mesh in the fluid rezone domain (which defines the fluid domain) has a mixture of small and large elements; the large elements exhibit most of the distortion due to the fluid rezone displacement. This is achieved by assigning a large stiffness to small elements and a low stiffness to large elements, as discussed in Masud and Hughes [15].

The upstream end of the flap is fixed hence the flap undergoes a large displacement to align itself in the horizontal position as shown in Figure 4(b). Figure 5 shows the streamlines of the flow in the vicinity of the flap and the contour plot of the structural displacement magnitude.

The multi-level Newton solver converges in 5 iterations to the desired tolerance 10^{-8} and exhibits terminal quadratic convergence (see Figure 6(a)). Each global Newton iteration typically requires an additional 15 GMRES iterations to solve the update Equation (15). This is a significant computational cost savings when compared with other conventional methods; see Figure 6 for comparisons with the successive substitution method and the MNM with semi-analytical derivative computations. The computational times for the fluid, fluid rezone and structural subsystem analyses 15, 1.5 and 0.2 min, respectively, on a Ultra Sparc 400 MHz processor with 1 GB main memory; therefore the total time for one global residual evaluation amounts to 16.7 min which is common for all three methods. Each evaluation of Equation (18) takes 45 s with our method and the semi-analytical method in contrast to 16.7 min for the finite difference method. The successive substitution method does not require the evaluation of Equation (15), however it takes many more global iterations to reach the same level of convergence. The total times (global iterations) taken by our method, the semi-analytical



(a)



(b)

Figure 4. Fluid mesh consisting of 4541 nodes and 8842 elements: (a) initial configuration; and (b) deformed configuration.

MNM, the finite difference based MNM and the successive substitution method (for various relaxation factors) are 161(6), 215(8), 2134(8) and 350–584(21–35) minutes (iterations), respectively.[‡‡] The truncation error due to the semi-analytical method is more dominant as the solution converges (see Figure 6). Convergence behaviours for the increased Reynolds number (see Figure 6(b)) are similar.

[‡‡]Computation time for the finite difference based method is based on the number of iterations of the semi-analytical method which is shown in Figure 6(a) and 15 GMRES iterations per global Newton iteration.

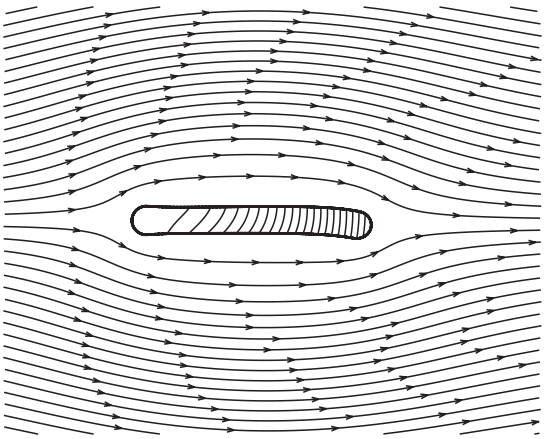


Figure 5. Flow and structural fields near the flap.

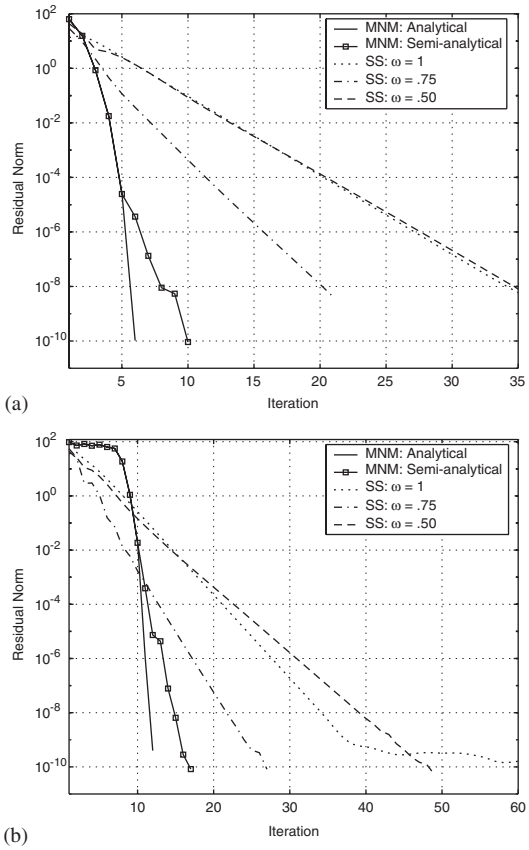


Figure 6. Convergence history using Multi-level Newton method (MNM) with analytical sensitivities, MNM with semi-analytical sensitivities and successive substitution (SS) with varying relaxation factors ω where the solution update is governed by $\mathbf{u}^{k+1} = (1-\omega)\mathbf{u}^{k-1} + \omega\mathbf{u}^{k}$: (a) $Re = 10$; and (b) $Re = 50$.

The computational savings by our method over the finite difference based multi-level Newton method (using the assumptions of footnote ‡‡) is at least the ratio between times taken to evaluate Equation (18) by the finite difference method and the analytical sensitivity method, which for this example is ~ 22. Moreover the numerical error associated with the finite difference based method increases the number of iterations required to obtain global convergence of Equation (14) and hence the number of times the linear system of Equation (15) must be solved. As seen in Figure 5, the initial performance of our method can be improved by employing the successive substitution method for a few iterations and then switching to the MNM. Other such adaptations, such as relaxation, may also hasten convergence.

4. CONCLUSIONS

Improvements to the multi-level Newton method are obtained by computing the inter-system coupling by analytical sensitivity techniques. The method demonstrates superior computational performance over the others. Additionally the framework of the multi-level Newton method is retained, hence new subsystems are easily integrated. Unlike the black-box approach of the finite difference based multi-level Newton method, our pseudo analyses may require code enhancements. However less development effort is required for these enhancements versus the effort to implement strongly coupled methods. Our method is amenable to multidisciplinary design optimization problems as accurate gradient information can be easily provided, this is in contrast to the relaxation-based methods.

APPENDIX A. SHAPE SENSITIVITY EVALUATION FOR FLUID ANALYSIS

Here we present the pseudo load formulation of the fluid subsystem analysis recalling that its pseudo load computation is performed on the element level whereupon it is assembled in the usual way. The governing equations for the element contributions to the conservation of mass and the conservation of the linear momentum are shown in functional form in Equations (A1) and (A2), respectively, for a standard Galerkin formulation. The integrals are expressed over the reference element domain Ω_r in this isoparametric analysis so that we can readily differentiate with respect to the node co-ordinates.

$$\Pi_A = \int_{\Omega_r} \omega(\mathbf{I} \cdot \nabla\tilde{\mathbf{v}}\mathbf{J}^{-1}) \det(\mathbf{J})\, \mathrm{d}\Omega \tag{A1}$$

$$\Pi_B = \int_{\Omega_r} \left\{\boldsymbol{\varphi} \cdot \rho\mathbf{b} - \nabla\boldsymbol{\varphi}\mathbf{J}^{-1} \cdot \boldsymbol{\sigma} - \boldsymbol{\varphi} \cdot \rho\nabla\tilde{\mathbf{v}}\mathbf{J}^{-1}\tilde{\mathbf{v}}\right\} \det(\mathbf{J})\, \mathrm{d}\Omega + \int_{\Gamma_r^h} \boldsymbol{\varphi} \cdot \mathbf{h} \det(\mathbf{J}) \|\mathbf{J}^{-\mathrm{T}}\mathbf{n}_r\|\, \mathrm{d}\Gamma \tag{A2}$$

In the above equations $\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\mathscr{S}(\nabla\tilde{\mathbf{v}}\mathbf{J}^{-1})$ states the constitutive relation for Newtonian fluid; ρ is the fluid density; μ is the fluid viscosity; $\mathbf{J}$ is the Jacobian for the isoparametric finite element mapping; Γ_r^h is the natural element boundary over which tractions are prescribed; $\tilde{\mathbf{v}}$ is the referential velocity vector; p is the pressure; $\mathbf{h}$ is the prescribed traction vector; $\mathbf{n}_r$ is the outward normal vector on Γ_r and ω and $\boldsymbol{\varphi}$ are the test functions for integrals Π_A

and Π_B. Finally $\mathscr{S}$ is a fourth-order tensor that operates on a second-order tensor returning a symmetric second-order tensor: $\mathscr{S}\mathbf{A} \stackrel{\text{def}}{=} \frac{1}{2}(\mathbf{A} + \mathbf{A}^{\mathrm{T}})$.

The node l co-ordinate vector $\mathbf{x}^l$ in the fluid domain Ω_q, i.e. Eulerian mesh, adapts due to the moving fluid–structure boundary. Using the nodal displacement $\mathbf{u}^l$ on Γ_{B} and the fluid rezone nodal displacement $\mathbf{w}^l$ the mesh adapts to

$$\mathbf{x}^l = \mathbf{X}^l + \boldsymbol{\xi}^l \quad \text{where } \boldsymbol{\xi}^l = \begin{cases} \mathbf{u}^l & \text{if } \mathbf{X}^l \in \Gamma_{\mathrm{B}} \\ \mathbf{w}^l & \text{otherwise} \end{cases} \tag{A3}$$

where $\mathbf{X}^l$ denotes the node l co-ordinate vector in the undeformed mesh, i.e. corresponding to Ω_w.

Since $\mathbf{X}^l$ is constant, we see that the derivatives of the node co-ordinates $\mathbf{x}^l$ in the $\mathbf{s}^l$ direction equal the derivatives of $\boldsymbol{\xi}^l$ in the same direction. Consequently we have for the function $\mathbf{f}(\boldsymbol{\xi}^l) = \mathbf{f}(\mathbf{x}^l - \mathbf{X}^l) = \bar{\mathbf{f}}(\mathbf{x}^l)|_{\mathbf{x}^l = \mathbf{X}^l + \boldsymbol{\xi}^l}$

$$\nabla \mathbf{f}(\boldsymbol{\xi}^l) = \nabla \bar{\mathbf{f}}(\mathbf{x}^l)|_{\mathbf{x}^l = \mathbf{X}^l + \boldsymbol{\xi}^l} \tag{A4}$$

Using this, we can evaluate the Jacobian, Jacobian directional derivative and the directional derivatives of its inverse and determinant as

$$\mathbf{J} = \mathbf{X}^e \nabla^{\mathrm{T}} \boldsymbol{\psi} \tag{A5}$$

$$\boldsymbol{\delta}_s \mathbf{J} = \mathbf{S}^e \nabla^{\mathrm{T}} \boldsymbol{\psi} \tag{A6}$$

$$\boldsymbol{\delta}_s \mathbf{J}^{-1} = -\mathbf{J}^{-1}(\boldsymbol{\delta}_s \mathbf{J})\mathbf{J}^{-1} \tag{A7}$$

$$\boldsymbol{\delta}_s \det(\mathbf{J}) = \det(\mathbf{J})(\mathbf{J}^{-\mathrm{T}} \cdot \boldsymbol{\delta}_s \mathbf{J}) \tag{A8}$$

where $\boldsymbol{\psi}$ is the interpolation vector (i.e. shape function vector) for the isoparametric mapping, $\mathbf{X}^e$ is the matrix whose columns contain the element node co-ordinate vectors $\mathbf{X}^l$ and $\mathbf{S}^e$ is the matrix whose columns contain the elements of $\mathbf{s}$ that correspond to the nodal fluid rezone displacement vectors. Note that $\mathbf{S}^e$ may contain elements of both $\mathbf{s}_w$ and $\mathbf{s}_u$.

The derivative of the integral Π_A in Equation (A9) is given

$$\boldsymbol{\delta}_s \Pi_A = \int_{\Omega_r} \omega\{(\mathbf{I} \cdot \nabla \tilde{\mathbf{v}} \boldsymbol{\delta}_s \mathbf{J}^{-1}) \det(\mathbf{J}) + (\mathbf{I} \cdot \nabla \tilde{\mathbf{v}} \mathbf{J}^{-1}) \boldsymbol{\delta}_s \det(\mathbf{J})\} \, \mathrm{d}\Omega \tag{A9}$$

And similarly for Π_B,

$$\begin{aligned} \boldsymbol{\delta}_s \Pi_B = &- \int_{\Omega_r} (\nabla \boldsymbol{\varphi} \boldsymbol{\delta}_s \mathbf{J}^{-1}) \cdot \boldsymbol{\sigma} \det(\mathbf{J}) \, \mathrm{d}\Omega \\ &- \int_{\Omega_r} \nabla \boldsymbol{\varphi} \mathbf{J}^{-1} \cdot 2\mu \mathscr{S}(\nabla \tilde{\mathbf{v}} \boldsymbol{\delta}_s \mathbf{J}^{-1}) \det(\mathbf{J}) \, \mathrm{d}\Omega \\ &- \int_{\Omega_r} \boldsymbol{\varphi} \cdot (\rho \tilde{\mathbf{v}} \boldsymbol{\delta}_s \mathbf{J}^{-1}) \tilde{\mathbf{v}} \det(\mathbf{J}) \, \mathrm{d}\Omega \end{aligned}$$

$$+\int_{\Omega_r}[\rho\mathbf{b}-\nabla\boldsymbol{\varphi}\mathbf{J}^{-1}\cdot\boldsymbol{\sigma}-\boldsymbol{\varphi}\cdot\rho\nabla\tilde{\mathbf{v}}\mathbf{J}^{-1}\tilde{\mathbf{v}}]\boldsymbol{\delta}_s\det(\mathbf{J})\,\mathrm{d}\Omega$$

$$+\int_{\Gamma_r^h}\boldsymbol{\varphi}\cdot\mathbf{h}\{\det(\mathbf{J})\boldsymbol{\delta}_s\|\mathbf{J}^{-\mathrm{T}}\mathbf{n}_r\|+\boldsymbol{\delta}_s\det(\mathbf{J})\|\mathbf{J}^{-\mathrm{T}}\mathbf{n}_r\|\}\,\mathrm{d}\Gamma \tag{A10}$$

REFERENCES

1. Guruswamy GP. Coupled finite-difference/finite element approach for wing-body aeroelasticity. In *Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization* 1992; 1–12.
2. Wood B, Loth E, Geubelle P, McIlwain S. A numerical methodology for an aeroelastic SBLI flow. In 38*th Aerospace Sciences Meeting & Exhibit*, *January 2000*, *Reno*, *Nevada. AIAA Paper 2000-0552*.
3. Felker FF. Direct solution of the two-dimensional Navier-Stokes equations for static aeroelasticity problems. *AIAA Journal* 1993; **31**(1):148–153.
4. Ghattas O, Li X. A variational finite element method for stationary non-linear fluid-solid interaction. *Journal of Computational Physics* 1995; **121**:347–356.
5. Lund E, Møller H, Jakobsen LA. Shape design optimization of steady fluid–structure interaction problems with large displacements. *AIAA Paper 2001-1624*.
6. Aluru NR, White J. A multilevel Newton method for mixed-energy domain simulation of MEMS. *Journal of Microelectromechanical Systems* 1999; **8**(3):299–308.
7. Bächtold M, Korvink JG, Funk J, Baltes H. New convergence scheme for self-consistent electromechanical analysis of iMEMS. In *Proc. Int. Electron Devices Meeting* 1995; 605–608.
8. Haftka R, Sobieszczanski-Sobieski J, Padula S. On options for interdisciplinary analysis and design optimization. *Structural Optimization* 1992; **4**:65–74.
9. Luenberger DG. *Linear and Nonlinear Programming*. Addison-Wesley: Reading, Massachusetts, 1984.
10. Scheuing JE, Tortorelli DA. Inverse heat conduction problem solution via second-order design sensitivities and Newton's method. *Inverse Problems in Engineering* 1996; **2**:227–262.
11. Olhoff N, Rasmussen J. Study of inaccuracy in semi-analytical sensitivity analysis—a model problem. *Structural Optimization* 1991; **3**(4):203–213.
12. Belytschko T, Flanagan DP, Kennedy JM. Finite element methods with user-controlled meshes for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**:669–688.
13. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: Chichester, 2000.
14. Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics* 1982; **46**:342–368.
15. Masud A, Hughes TJR. A space-time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems. *Computer Methods in Applied Mechanics and Engineering* 1997; **146**:91–126.
16. Ghattas O, Li X. Domain decomposition methods for sensitivity analysis of a non-linear aeroelasticity problem. *International Journal of Computational Fluid Dynamics* 1998; **11**:113–130.
17. Bathe KJ. *Finite Element Procedures*. Prentice Hall: Englewood Cliffs, NJ, 1996.
18. Smith DE, Tortorelli DA, Tucker CL. Optimal design for polymer extrusion. Part I: Sensitivity analysis for non-linear steady-state systems. *Computer Methods in Applied Mechanics and Engineering* 1998; **167**: 283–302.
19. Kikuchi N. *Finite Element Methods in Mechanics*. Cambridge University Press: New York, 1986.
20. Vandevender WH, Haskell KH. The SLATEC mathematical subroutine library. *Signum Newsletter* 1982; **2**(3): 16–21.
21. *Gambit 1.3 User manual*. Fluent Inc., 2000.
22. Barthelemy B, Chon CT, Haftka RT. Accuracy problems associated with semi-analytical derivatives of static response. *Finite Elements in Analysis and Design* 1988; **4**:249–265.
23. Heath MT. *Scientific Computing: An Introductory Survey*. McGraw Hill: New York, 1997.
24. Jansen KE, Shakib F, Hughes TJR. Fast projection algorithm for unstructured meshes. In *Computational Nonlinear Mechanics in Aerospace Engineering*, Atluri SN (ed.). AIAA: Washington, D.C., 1992; 175–204.
25. Tortorelli DA, Michalaris P. Design sensitivity analysis: overview and review. *Inverse Problems in Engineering* 1994; **1**:71–103.

