Coupling of Partitioned Physics Codes with Quasi-Newton Methods

Rob Haelterman, Alfred Bogaers, Joris Degroote, Silviu Cracana

***Abstract*—Many physics problems can only be studied by coupling various numerical codes, each modeling a subaspect of the physics problem that is addressed. Often, each of these codes needs to be considered as a black box, either because the codes were written by different programmers, are proprietary software or are legacy code that can only be modified with major effort.**
Running these black boxes one after another, until convergence is reached, is a standard solution technique. It is easy to implement but comes at the cost of slow or even conditional convergence.
A recent interpretation of this approach as a root-finding problem has opened the door to acceleration techniques based on quasi-Newton methods. These quasi-Newton methods can easily be "strapped onto" the original iterative loop without the need to modify the underlying code and with little extra computational cost.
In this paper we analyze the performance of ten acceleration techniques that can be applied to accelerate the convergence of a non-linear Gauss-Seidel iteration, on three different multi-physics problems. The methods range from the very well known Broyden method to the arcane Eirola-Nevanlinna method. A switching strategy that was mooted a number of years ago for Broyden's method, and was claimed to give promising results, but then fell by the wayside, is also considered. For the first time, this idea has been generalized to a wider class of quasi-Newton methods.

***Index Terms*—Partitioned methods, iterative methods, quasi-Newton.**

I. Introduction

ENGINEERING problems often take the form of the non-linear system of equations given by

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ & \vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{cases} \quad (1)$$

where $x_1, x_2, \ldots, x_n$ can be scalars or vectors of varying sizes.

For the problems that we are interested in, we typically have the following characteristics [8]:

1) Good solvers exist for each equation of the system (e.g. in multi-physics where each equation represents one of the physical components). For this reason each equation will be solved separately, i.e. we use a *partitioned solution method*.
2) The problem has a large dimensionality, often imposing the use of matrix-free implementations.
3) The analytic form of f_i $(i = 1, 2, \ldots, n)$ is unknown, preventing the use of Newton's method, for instance.
4) Evaluating f_i $(i = 1, 2, \ldots, n)$ is computationally costly, preventing line-search techniques and matrix free Newton-Krylov techniques. This also means that the required number of evaluations (or "calls") to reach convergence is a good proxy of the performance of an algorithm.

If F' (the Jacobian of $F = (f_1, \ldots, f_n)$) satisfies the condition

$$\forall i < j < n : [F']_{ij} = 0 \quad (2)$$

then quasi-Newton (QN) acceleration can be applied to the system (1) as follows [13].

Quasi-Newton acceleration

1. Startup:
   1.1. Take initial values x_2^1, x_3^1, $\ldots x_n^1$.
   1.2. Set $s = 1$.
2. Loop until convergence:
   2.1. Solve $f_1(x_1, x_2^s, \ldots, x_n^s) = 0$ for x_1, resulting in x_1^{s+1}.
   2.2. Solve $f_2(x_1^{s+1}, x_2, \ldots, x_n^s) = 0$ for x_2, resulting in x_2^{s+1}.
   $\ldots$
   2.n. Solve $f_n(x_1^{s+1}, x_2^{s+1}, \ldots, x_n) = 0$ for x_n. Consider the obtained value to be $H(x_n^s)$.
   2.n+1. Define K by $K(x) = H(x) - x$ and compute an approximate Jacobian $\hat{K}'_s$ of K (see below)
   2.n+2. $x_n^{s+1} = x_n^s - (\hat{K}'_s)^{-1} K(x_n^s)$
   2.n+3. Set $s = s + 1$.

II. Different choices of quasi-Newton methods

The difference between the various quasi-Newton methods that we consider here lies in the choice of $\hat{K}'_s$.

We define $\delta x_s = x_n^{s+1} - x_n^s$, $\delta K_s = K(x_n^{s+1}) - K(x_n^s)$ and $\{\imath_j; j = 1, \ldots, n\}$ as the canonical (orthonormal) basis for $\mathbb{R}^{n \times 1}$.

Manuscript received December 09, 2016; revised January 24, 2017.

R. Haelterman is associate professor at the Royal Military Academy, Dept. Mathematics, Renaissancelaan 30, B-1000 Brussels, Belgium. email: robby.haelterman@mil.be

A. Bogaers is senior researcher at the Council for Scientific and Industrial Research, Advanced Mathematical Modelling, Modelling and Digital Sciences, Meiring Naudé Road; Brummeria, Pretoria, South Africa. email: abogaers@csir.co.za

J. Degroote is associate professor at Ghent University, Dept. Flow, Heat and Combustion Mechanics, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium. email: joris.degroote@ugent.be

S. Cracana is a student at the Technical Military Academy of Bucharest, Bulevardul George Coşbuc 81-83, Bucureşti 050141, Romania. email: cracana.s@gmail.com

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1) Non-Linear Gauss-Seidel[1] (GS). This method is seldom considered to be a quasi-Newton method, but can take its form if we set $(\hat{K}'_s)^{-1} = -I$. In essence, it is the procedure resulting from solving the different equations sequentially for a single variable, i.e. $x_n^{s+1} = H(x_n^s)$.
2) Aitken's δ^2 method ($A\delta^2$) [1]. While again seldom seen as a quasi-Newton method, it can take its form if we define $(\hat{K}'_{s+1})^{-1} = -\frac{1}{\omega_s} I$ with

$$\omega_{s+1} = -\omega_s \frac{\langle K(x_n^{s-1}), K(x_n^s) - K(x_n^{s-1})\rangle}{\langle K(x_n^s) - K(x_n^{s-1}), K(x_n^s) - K(x_n^{s-1})\rangle}. \quad (3)$$

3) Broyden's first (or "good") method (BG) [2], [3], [4], [5], [9]:

$$(\hat{K}'_{s+1})^{-1} = (\hat{K}'_s)^{-1} + \frac{(\delta x_s - (\hat{K}'_s)^{-1}\delta K_s)\delta x_s^T (\hat{K}'_s)^{-1}}{\langle \delta x_s, (\hat{K}'_s)^{-1}\delta K_s\rangle}. \quad (4)$$

4) Broyden's second (or "bad") method (BB) [2], [5], [9]:

$$(\hat{K}'_{s+1})^{-1} = (\hat{K}'_s)^{-1} + \frac{(\delta x_s - (\hat{K}'_s)^{-1}\delta K_s)\delta K_s^T}{\langle \delta K_s, \delta K_s\rangle}. \quad (5)$$

5) Switched Broyden method (BS)[15] . This is a method where at each iteration a choice is made between BG and BB as follows. If

$$\frac{|\delta x_s^T \delta x_{s-1}|}{|\delta x_s^T (\hat{K}'_s)^{-1}\delta K_s|} < \frac{|\delta K_s^T \delta K_{s-1}|}{\delta K_s^T \delta K_s} \quad (6)$$

then (4) is used, otherwise (5) is used.
6) Column-Updating Method (CU) [10], [16], [18], [19]:

$$(\hat{K}'_{s+1})^{-1} = (\hat{K}'_s)^{-1} + \frac{(\delta x_s - (\hat{K}'_s)^{-1}\delta K_s)\imath_{j_{K,s}}^T (\hat{K}'_s)^{-1}}{\langle \imath_{j_{K,s}}, (\hat{K}'_s)^{-1}\delta K_s\rangle} \quad (7)$$

where $\imath_{j_{K,s}}$ is chosen such that

$$j_{K,s} = \quad \text{Argmax}\{|\langle \imath_j, \delta x_s\rangle|; j = 1, \ldots, n\}.$$

7) Inverse Column-Updating Method (ICU) [14], [17]:

$$(\hat{K}'_{s+1})^{-1} = (\hat{K}'_s)^{-1} + \frac{(\delta x_s - (\hat{K}'_s)^{-1}\delta K_s)\imath_{j_{M,s}}^T}{\langle \imath_{j_{M,s}}, \delta K_s\rangle}, \quad (8)$$

where $\imath_{j_{M,s}}$ is chosen such that

$$j_{M,s} = \quad \text{Argmax}\{|\langle \imath_j, \delta K_s\rangle|; j = 1, \ldots, n\}.$$

8) Switched (Inverse) Column-Updating Method (CUS). As far as the authors are aware the idea behind BS has not been applied to CU and ICU, even though it is straightforward. As for BS, we use the condition (6) at every iteration. When it is satisfied, then CU (equation (7)) is used, otherwise ICU (equation (8)).

[1]Also called, among others, "Iterative Substructuring Method" or "Picard iteration"; see [11] and references therein.

It is clear from the different update formulas for the approximate Jacobian of BG, BB, BS, CU, ICU and CUS that they can only be applied starting with $\hat{K}'_2$. In other words, $\hat{K}'_1$ needs to be chosen. Conventionally, this is set to be equal to $-I$. Likewise, for $A\delta^2$, ω_1 needs to be chosen and is set to 1. As a result all of the methods given above will have an identical first iteration, i.e. $x_n^2 = H(x_n^1)$.
The Eirola-Nevanlinna methods that we present now are different in this respect as they compute $\hat{K}'_1$ based on an implicit choice of $\hat{K}'_0$. Again we will set these to be equal to $-I$, which can also be interpreted as setting the initial approximation of the Jacobian of H as zero.

9) Eirola-Nevanlinna Type I method (EN1) [7] :

$$(\hat{K}'_s)^{-1} = (\hat{K}'_{s-1})^{-1} + (p_s - (\hat{K}'_{s-1})^{-1} q_s)\frac{p_s^T (\hat{K}'_{s-1})^{-1}}{\langle p_s, (\hat{K}'_{s-1})^{-1} q_s\rangle}, \quad (9)$$

where $p_s = -(\hat{K}'_{s-1})^{-1} K(x_n^s)$,
$q_s = K(x_n^s + p_s) - K(x_n^s)$
10) Eirola-Nevanlinna Type II method (EN2) [8]:

$$(\hat{K}'_s)^{-1} = (\hat{K}'_{s-1})^{-1} + (p_s - (\hat{K}'_{s-1})^{-1} q_s)\frac{q_s^T}{\langle q_s, q_s\rangle}, \quad (10)$$

where p_s and q_s are defined as in the EN1 method.
11) Switched Eirola-Nevanlinna method (ENS). As far as the authors are aware the idea behind BS has not been applied to EN1 and EN2, even though it is straightforward. As for BS, we use the condition (6) at every iteration. When it is satisfied, then EN1 (equation (9)) is used, otherwise EN2 (equation (10)).

Note that the EN algorithms require two calls of K (or H) per iteration.

III. Test-cases

A. Simplified model of plasma heating by radio frequency waves in a tokamak plasma

The model is a simplified version of the set of codes commonly used to describe the steady state of plasma heating by radio frequency waves in a tokamak plasma. It consists of a wave equation as well as a Fokker-Planck velocity space diffusion and a radial energy diffusion model. While simple, it still has all the characteristics needed to validate our ideas:

1) The model captures enough of the real physics to be representative for more elaborate models.
2) It is a legacy code which has been (partly) developed by third-party code-writers.
3) The runtime of a fixed-point iteration with this code is sufficiently high so that the extra runtime of most of the standard acceleration techniques can safely be neglected.

As one of the main objectives of using this code is to gain insights in how legacy codes can be improved by adding

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

a quasi-Newton step as a "wrapper", we will abstain from altering any part of the code previously used in [12]. This means that we also keep the input and output variables as they were defined previously.
In an abstract form the governing equations can be written as:

$$\begin{cases} f_1(y_1, y_2, y_3, y_4, y_5, y_6, y_7) &= 0 \\ f_2(y_1, y_2, y_3, y_4, y_6, y_8, y_9, y_{10}) &= 0 \\ f_3(y_1, y_2, y_3, y_5, y_6, y_7, y_8, y_9, y_{10}) &= 0, \end{cases} \quad (11)$$

where

- f_1 = simplified 1-component wave equation for the fast magneto-sonic wave launched from the antenna and damped inside the plasma,
- f_2 = simplified isotropic Fokker-Planck equation studying the impact of the absorbed wave power on the distribution functions of the minority species,
- f_3 = simplified 1D diffusion equation describing the transport of energy across magnetic surfaces, establishing a temperature profile consistent with the absorbed power,

and

- y_1, resp. y_2 and y_3 = temperature profile of the majority ions, resp. minority ions and electrons, specified at a set of grid points;
- y_4 = effective temperature (average energy) of the minority ions;
- y_5, resp. y_6 and y_7 = direct power density profile of the fast wave damping onto the majority ions, resp. minority ions and electrons;
- y_8, resp. y_9 and y_{10} = Coulomb collisionally redistributed minority power density fraction profile onto the majority ions, resp. minority ions and electrons.

As the system in (11) does not satisfy condition (2), variables need to be grouped, as explained in [13]. The minimum grouping to respect the condition (and to keep the sequence of the different solvers, which are hardwired in the legacy code) is (y_1, y_2, y_3, y_4), which will play the role of $x_n = x_3$ in the quasi-Newton algorithm. A convergence criterium of $\frac{\|x_n^{s+1} - x_n^s\|}{\|x_n^s\|} \leq 10^{-7}$ is used.

The results are shown in tables I–III for various values of launched power and diffusion coefficient κ. The QN methods clearly outperform G-S and Aitken's method. Of the three groups of QN methods (Broyden, Column Updating, Eirola-Nevanlinna), the Broyden methods most often give the best results, even though, for some parameter combinations, Column Updating has a slight edge. While not equivocally so, the switching strategy often (slightly) improves the convergence speed of the underlying methods.

B. 1D flexible tube

This test-case describes one-dimensional unsteady flow in a flexible tube of length L.
The fluid is incompressible and inviscid and gravity is neglected. The governing equations are the conservation of mass and momentum (in conservative form).

TABLE I: Simplified tokamak model. Number of function calls (of H) needed to reach convergence for various values of the diffusion coefficient κ and different iterative methods. Launched power = 2MW. "div" = divergence or no convergence after 100 iterations. The top performing method for each setting is highlighted in bold.

$\kappa(\cdot 10^{-2})$	3.5	4.0	4.5	5.0	7.5	10	25	50	75	100
G-S	div	div	95	88	67	55	30	19	14	14
$A\delta^2$	63	52	52	49	48	33	24	17	14	15
BG	div	33	30	29	28	23	18	14	**13**	**12**
BB	**30**	**30**	29	30	28	23	18	14	**13**	**12**
BS	31	div	**28**	**25**	**26**	**21**	18	14	**13**	**12**
CU	37	div	div	37	32	22	**17**	**12**	**13**	**12**
ICU	34	36	33	32	30	24	18	14	14	**12**
CUS	33	38	**28**	29	30	23	18	14	**13**	**12**
EN1	32	34	40	30	30	24	22	16	16	16
EN2	34	36	34	34	32	26	22	16	16	16
ENS	div	div	34	38	30	24	20	16	16	16

TABLE II: Simplified tokamak model. Number of function calls (of H) needed to reach convergence for various values of the diffusion coefficient κ and different iterative methods. Launched power = 5MW. "div" = divergence or no convergence after 100 iterations. The top performing method for each setting is highlighted in bold.

$\kappa(\cdot 10^{-2})$	7.5	10	25	50	75	100
G-S	67	52	28	19	16	15
$A\delta^2$	39	36	23	19	16	16
BG	26	24	**19**	**17**	**14**	13
BB	25	24	20	**17**	**14**	13
BS	**24**	**23**	**19**	**17**	15	13
CU	36	div	21	**17**	**14**	**12**
ICU	27	30	21	18	15	14
CUS	26	27	21	**17**	**14**	14
EN1	28	28	24	20	16	16
EN2	28	28	24	20	16	16
ENS	30	32	24	18	16	16

TABLE III: Simplified tokamak model. Number of function calls (of H) needed to reach convergence for various values of the diffusion coefficient κ and different iterative methods. Launched power = 8MW. The top performing method for each setting is highlighted in bold.

$\kappa(\cdot 10^{-2})$	10	25	50	75	100	125
G-S	54	29	22	17	15	16
$A\delta^2$	35	25	24	17	16	16
BG	25	20	19	**15**	**13**	**13**
BB	**24**	20	19	**15**	14	**13**
BS	**24**	21	**16**	**15**	14	15
CU	26	22	20	**15**	14	14
ICU	26	22	17	16	14	14
CUS	27	**16**	20	16	14	**13**
EN1	28	24	24	18	16	16
EN2	28	24	22	18	16	16
ENS	28	24	22	18	16	16

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The velocity at the inlet of the tube is imposed as

$$\mathrm{u}(t) \quad = \quad \mathrm{u}_o + \frac{\mathrm{u}_o}{10} \sin^2(\pi t), \tag{12}$$

where u_o is a reference velocity and t is time.
A non-reflecting boundary condition is prescribed at the outlet.
The resulting flow equations are discretized on a one-dimensional equidistant mesh with 1001 cells. The fluid velocity and pressure are stored in the mesh nodes. Central discretization of all terms in the continuity and momentum equations is used, except for the convective term in the momentum equation which is discretized with a first-order upwind scheme. The time discretization scheme is backward Euler.

The elastic wall of the tube has a Hookean constitutive law where the cross-sectional area is only a function of the local kinematic pressure and its mass/inertia is neglected with regards to that of the fluid. The geometrical discretization of the elastic problem is identical to that of the flow problem to avoid errors in the data transfer between the fluid and the structure. More details can be found in [11].

For this test-case solving $f_1(x_1, x_2) = 0$ for x_1 represents a call of the structural solver and solving $f_2(x_1, x_2) = 0$ for x_2 a call of the flow solver, with x_1 being the variable describing the geometry and x_2 the variable describing the flow variables. Here we only consider pressure, which is exchanged between the flow and the structural solver; the velocity is an internal variable to the flow problem.

At each time step a convergence criterion of $\|K(x_n^s)\| \leq 10^{-8}$ is used. At the beginning of a new time step the initial approximate Jacobian is set equal to the last approximation at the previous time step or, alternatively, re-set to $-I$ at the beginning of each new time step. Whenever the current approximation of the Jacobian is $-I$, under-relaxation is applied with a factor σ. The results are shown in tables IV–V for various values of the dimensionless structural stiffness κ and the dimensionless time step τ.

The QN methods clearly outperform G-S and Aitken's method. Of the three groups of QN methods (Broyden, Column Updating, Eirola-Nevanlinna), the Broyden methods give the best results. Within the Broyden class, Broyden's "good" method has the fastest convergence, with little to be gained from the switching strategy.
This example also clearly shows the benefits of re-using the Jacobian of the previous time step, which for the harder parameter combinations can result in a reduction of the number of iterations that are required for convergence up to 75%. A notable exception is the ENS algorithm, for reasons that are not yet fully understood.

TABLE IV: 1D flexible tube. Number of function calls (of H) needed to reach convergence (averaged over ten time steps) for various values of the dimensionless structural stiffness κ and the dimensionless time step τ when the Jacobian at the end of the iterations within a time step is carried over to the beginning of the new time step. "div" = divergence or no convergence after 100 iterations. The top performing method for each setting is highlighted in bold.

κ	10^3	10^3	10^3	10^3	10^2	10^2
τ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}
σ	10^{-2}	10^{-2}	10^{-2}	10^{-3}	10^{-2}	10^{-2}
G-S	5.0	7.0	div	div	8.0	div
$A\delta^2$	5.8	6.0	8.2	23.9	6.8	9.0
BG	**2.8**	**3.0**	**3.5**	**4.4**	**3.3**	**3.7**
BB	**2.8**	**3.0**	3.6	5.6	**3.3**	3.9
BS	**2.8**	**3.0**	3.6	5.3	**3.3**	**3.7**
CUS	3.0	3.1	4.3	6.5	3.4	4.3
ICU	3.0	3.1	5.1	10.4	3.4	5.1
CUS	3.0	3.1	4.3	6.5	3.4	4.4
EN1	4.0	4.0	6.0	5.0	4.8	5.8
EN2	4.0	4.0	6.0	5.0	4.8	5.8
ENS	4.0	4.0	6.0	9.2	4.8	5.8

κ	10^2	10^2	10	10	10	10
τ	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
σ	10^{-2}	10^{-3}	10^{-2}	10^{-4}	10^{-5}	10^{-6}
G-S	div	div	div	div	div	div
$A\delta^2$	18.6	div	56.5	81.0	div	div
BG	**4.6**	**9.2**	4.8	**5.1**	**9.5**	div
BB	5.9	div	5.2	7.3	div	div
BS	5.3	12.0	**4.6**	6.4	12.9	div
CUS	6.7	16.4	5.2	7.2	17.2	div
ICU	12.4	div	5.4	14.1	10.3	div
CUS	6.5	21.3	4.9	7.3	17.5	div
EN1	5.4	9.4	6.2	6.6	13.2	div
EN2	5.4	9.4	6.2	6.6	13.2	div
ENS	9.0	9.4	6.2	11.8	div	div

C. 2D flexible beam

The selected test case is a fluid-structure interaction problem consisting of flow around a fixed cylinder with an attached flexible beam. The beam undergoes large deformations induced by oscillating vortices formed by flow around the circular bluff body. The problem was first proposed by Turek *et al.* [20], and has received substantial numerical verification. The problem layout and material properties are provided in Figure 1(a). The problem is solved here by coupling OpenFOAM [21], an open-source, finite-volume based fluid flow solver, and Caculix [6], an open-source finite-element based solver for the structural domain deformations.

The FSI problem consists of a 0.02m thick, 0.35m long flexible beam, attached to a fixed cylinder with diameter of 0.1m. The cylinder center is by design constructed to be non-symmetric to remove dependence on numerical errors to induce the onset of deformations. A parabolic inlet boundary condition, with mean flow velocity of $\bar{U} = 1$m/s is slowly ramped up for $t < 0.5$s via $(1 - \cos(\pi t/2))/2$. The top, bottom and fixed cylinder walls are defined as non-slip boundaries. The problem is solved here using 3800 finite-volume fluid cells, and 72 full integration, bi-quadratic finite elements, resulting in an interface Jacobian matrix size of $n = 1119$.

For this test-case f_1 represents the flow solver and f_2 the structural solver. At the beginning of a new time step the initial approximate Jacobian is set equal to the last

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE V: 1D flexible tube. Number of function calls (of H) needed to reach convergence (averaged over ten time steps) for various values of the dimensionless structural stiffness κ and the dimensionless time step τ when the Jacobian at the beginning of each new time step is re-set. "div" = divergence or no convergence after 100 iterations. The top performing method for each setting is highlighted in bold.

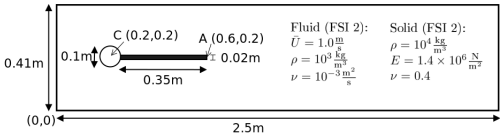
κ	10^3	10^3	10^3	10^3	10^2	10^2
τ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}
σ	10^{-2}	10^{-2}	10^{-2}	10^{-3}	10^{-2}	10^{-2}
G-S	5.0	7.0	div	div	8.0	div
$A\delta^2$	5.8	6.0	8.2	23.9	6.8	9.0
BG	**3.0**	**3.0**	**4.9**	**8.7**	**4.0**	**5.0**
BB	**3.0**	**3.0**	**4.9**	**8.7**	**4.0**	**5.0**
BS	**3.0**	**3.0**	**4.9**	**8.7**	**4.0**	**5.0**
CUS	**3.0**	4.0	5.9	10.5	4.2	6.0
ICU	**3.0**	4.0	5.9	12.1	4.2	6.0
CUS	**3.0**	4.0	5.9	10.4	4.2	6.0
EN1	4.0	4.0	6.0	9.8	6.0	6.0
EN2	4.0	4.0	6.0	9.8	6.0	6.0
ENS	4.0	4.0	6.0	9.8	6.0	6.0

κ	10^2	10^2	10	10	10	10
τ	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
σ	10^{-2}	10^{-3}	10^{-2}	10^{-4}	10^{-5}	10^{-6}
G-S	div	div	div	div	div	div
$A\delta^2$	18.6	div	56.5	81.0	div	div
BG	**9.0**	35.5	**5.5**	**9.7**	37.3	div
BB	**9.0**	div	**5.5**	11.1	div	div
BS	**9.0**	**35.0**	**5.5**	**9.7**	**37.1**	div
CUS	11.0	39.2	6.3	11.5	51.3	div
ICU	13.2	div	6.2	14.7	div	div
CUS	11.1	39.9	6.3	11.5	57.5	div
EN1	10.2	39.0	7.4	10.6	div	div
EN2	10.2	39.0	7.4	10.6	div	div
ENS	10.4	39.0	7.4	11.0	div	div

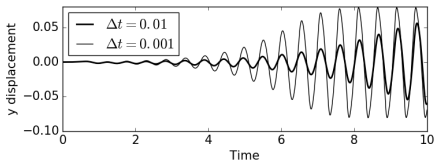
approximation at the previous time step or, alternatively, re-set to $-I$ at the beginning of each new time step.

We investigate three different settings, namely for a comparatively large time step size of $\Delta t = 0.01$s for two different convergence criteria of $\epsilon = \frac{\|K(x_n^s)\|}{\sqrt{n}} = 10^{-5}$ and $\epsilon = 10^{-8}$ in order to gain some insight into the convergence behavior of the various QN methods as well as for a small time step size of $\Delta t = 0.001$s for $\epsilon = 10^{-8}$.

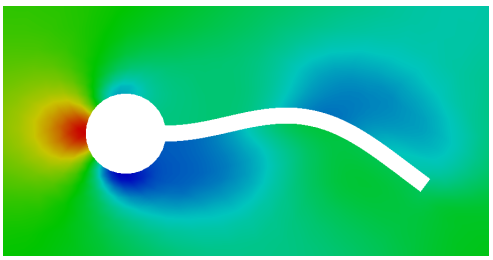
The beam tip displacement for both time step sizes over the full simulation window is shown in Figure 1(b) with a snapshot of the beam displacement at 8.7s shown in Figure 1(c). The convergence behavior for the various QN methods is summarized in Table VI and VII. Overall, when the Jacobian is carried over to the new time-step, switched Broyden outperforms all the QN methods across all three settings, with the switched strategies providing improved performance for both the conventional EN and CU methods. When the Jacobian is re-set at each time-step, the appeal of the switched strategies diminishes for each class of QN methods. Nevertheless, the group of Broyden methods remains the best performing.



(a)



(b)



(c)

Fig. 1: (a) Flexible beam problem description, (b) beam tip displacement over the 10s simulation window, and (c) beam displacement and pressure contours at 8.7 seconds.

TABLE VI: 2D flexible beam. Number of iterations to reach convergence, averaged over the number of time steps, for the flexible tail benchmark problem for the various QN methods. The Jacobian at the end of the iterations within a time step is carried over to the beginning of the new time step. Failure to converge is indicated by "div" where the time step at which failure occurred is indicated in brackets, and the top performing method for each setting is highlighted in bold.

	$\Delta t = 0.01$, $\epsilon = 10^{-5}$	$\Delta t = 0.01$, $\epsilon = 10^{-8}$	$\Delta t = 0.001$, $\epsilon = 10^{-8}$
GS	div(1)	div(1)	div(1)
$A\delta^2$	6.263	14.995	-(10)
BG	2.938	4.017	4.407
BB	3.039	4.250	4.480
BS	**2.906**	**3.983**	**4.024**
CU	3.342	5.264	-(1581)
ICU	3.108	5.391	8.388
CUS	2.975	4.836	6.025
EN1	3.192	4.614	5.443
EN2	3.147	4.543	5.854
ENS	3.137	4.535	4.920

IV. Conclusion

We have tested a wide variety of acceleration techniques on three different multi-physics problems that are written as a fixed-point problem. While the choice of the best method remains problem dependent, it is clear that the best choice is the class of quasi-Newton methods, of which, more often than not, the tried and trusted Broyden method comes out on top.

Re-using the Jacobian of all the QN methods at the beginning of the iterations of the next time step results in important reductions in the required number of iterations. With a few exceptions, a switching strategy, that hasn't drawn much attention in the past, is shown to offer a slight boost of perfor-

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE VII: 2D flexible beam. Number of iterations to reach convergence, averaged over the number of time steps, for the flexible tail benchmark problem for the various QN methods. The Jacobian at the beginning of each new time step is re-set to $-I$. Failure to converge is indicated by "div" where the time step at which failure occurred is indicated in brackets, and the top performing method for each setting is highlighted in bold.

	$\Delta t = 0.01$, $\epsilon = 10^{-5}$	$\Delta t = 0.01$, $\epsilon = 10^{-8}$	$\Delta t = 0.001$, $\epsilon = 10^{-8}$
GS	div(8)	div(1)	div(1)
$A\delta^2$	4.504	11.459	-(5)
BG	4.170	6.617	6.872
BB	**3.879**	**6.518**	7.369
BS	4.196	6.611	**6.920**
CU	4.499	7.165	8.295
ICU	4.323	7.042	9.158
CUS	4.500	7.168	8.275
EN1	4.879	7.605	8.793
EN2	7.632	7.583	8.448
ENS	4.879	7.605	8.795

mance in exchange for a negligeable penalty in complexity. The class of Eirola-Nevanlinna methods, which are among the lesser known QN methods, have not shown their worth, and in the authors' opinion do not seem to warrant the complexity that they entail.

REFERENCES

[1] A.C. Aitken, On Bernouilli's numerical solution of algebraic equations. Proc. Roy. Soc. Edinb. **46**, pp. 289–305 (1926)
[2] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations. *Math. Comp.* **19**, pp. 577–593 (1965)
[3] C.G. Broyden, Quasi-Newton methods and their applications to function minimization. *Math. Comp.* **21**, pp. 368–381 (1967)
[4] J.E. Dennis, J.J. Moré, Quasi-Newton methods: motivation and theory. *SIAM Rev.* **19**, pp. 46–89 (1977)
[5] J.E. Dennis, R.B. Schnabel, Least Change Secant Updates for quasi-Newton methods. *SIAM Rev.* **21**, pp. 443–459 (1979)
[6] G. Dhondt, CalculiX CrunchiX USER'S MANUAL Version 2.5 (2007)
[7] T. Eirola, O. Nevanlinna, Accelerating with rank-one updates. *Linear Algebra Appl.* **121**, pp. 511–520 (1989)
[8] H.-R. Fang, Y. Saad, Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, **16**/3, pp. 197–221 (2009).
[9] A. Friedlander, M.A. Gomes-Ruggiero, D.N. Kozakevich, J.M. Martinez, S.A. dos Santos, Solving nonlinear systems of equations by means of quasi-Newton methods with a nonmonotone strategy. *Optim. Methods Softw.* **8**, pp. 25–51 (1997)
[10] M.A. Gomez-Ruggiero, J.M. Martinez, The Column-Updating Method for solving nonlinear equations in Hilbert space. *RAIRO Mathematical Modelling and Numerical Analysis* **26**, pp. 309–330 (1992)
[11] R. Haelterman, Analytical Study of the Least Squares Quasi-Newton Method for Interaction Problems, Ph.D. thesis, Ghent University, Ghent, Belgium, 2009.
[12] R. Haelterman, D. Van Eester, D. Verleyen, Accelerating the solution of a physics model inside a Tokamak using the (Inverse) Column Updating Method, *Journal of Computational and Applied Mathematics* **279**, pp. 133–144 (2015)
[13] R. Haelterman, D. Van Eester, S. Cracana, Does Anderson Always Accelerate Picard ?, *14th Copper Mountain Conference on Iterative Methods*, Copper Mountain, USA (2016).
[14] V.L.R. Lopes, J.M. Martinez, Convergence properties of the Inverse Column-Updating Method. *Optim. Methods Softw.* **6**, pp. 127–144 (1995)
[15] J.M. Martinez, L.S. Ochi, Sobre Dois Metodos de Broyden. *Mat. Apl. Comput.* **1**/2, pp. 135–143 (1982)
[16] J.M. Martinez, A quasi-Newton method with modification of one column per iteration. *Computing* **33**, pp. 353–362 (1984)
[17] J.M. Martinez, M.C. Zambaldi, An Inverse Column-Updating Method for solving large-scale nonlinear systems of equations. *Optim. Methods Softw.* **1**, pp. 129–140 (1992)
[18] J.M. Martinez, On the convergence of the column-updating method. *Comp. Appl. Math.* **12**/2, pp. 83–94 (1993)
[19] J.M. Martinez, Practical quasi-Newton method for solving nonlinear systems. *J. Comput. Appl. Math.* **124**, pp. 97–122 (2000)
[20] S. Turek, J. Hron, Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow, In: *Fluid-Structure Interaction*, Ed. H.-J. Bungartz and M. Schäfer, Michael, Series "Modelling, Simulation, Optimisation" Vol. **53**, Springer Berlin Heidelberg, ISSN 1439-7358, pp. 371–385 (2006)
[21] H. Weller, OpenFOAM: The Open Source CFD Toolbox User Guide, Version 2.1.0, (2010), http://openfoam.com/

ISBN: 978-988-14047-7-0
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)